Linking Neural and Behavioral Data: Discriminative and Generative Models

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Auguste Henriette Schulz
aus München

Tübingen
2025

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	04.02.2026
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter/-in:	Prof. Dr. Jakob Macke
2. Berichterstatter/-in:	Prof. Dr. Anna Levina

To the architect who would seize my yearly math riddle tests after school to solve them for fun, showing me that the enjoyment of STEM is not limited to one gender.

Contents

Acronyms

GLM Generalized Linear Model.

VAE Variational Autoencoder.

GAN Generative Adversarial Network.

DDPM Denoising Diffusion Probabilistic Models.

LDNS Latent Diffusion for Neural Spiking Data.

ELBO Evidence Lower Bound.

Abstract

A central goal in neuroscience is to understand how neural activity gives rise to behavior. Encoding and decoding models formalize this relationship by estimating conditional distributions of neural activity given behavior or other covariates, and vice versa. Advances in neural and behavioral recording techniques now enable recordings of thousands of neurons during continuous behaviors beyond rigid trial structures. These developments pose a challenge to classical encoding and decoding methods, which often struggle with high dimensionality, variability, and non-linear relationships. Thus, the overarching aim of this thesis was to develop statistical approaches linking high-dimensional neural activity with continuous behavior.

The first contribution addresses whether the midbrain, in particular superior colliculus (SC), differentiates visual motion resulting from self-movement or object-movement. We tested this in virtual reality, where mice experienced identical visual loom for these two contexts. Because neural activity drives, but is also shaped by behavior, we had to account for behavioral differences when decoding context from neural activity. Using a multivariate discriminative decoding framework, we found that SC activity, particularly in intermediate layers, differs between contexts, even after controlling for behavior.

The second contribution directly accounts for this neural-behavioral bidirectionality, hidden in distinct encoding and decoding models. We developed a probabilistic latent variable model based on masked variational autoencoders (VAEs) to jointly model conditional distributions of neural activity and behavior. This framework allowed us to model and sample from the distribution over continuous behaviors given neural activity and to generate neural activity conditioned on unseen behavior. Masked VAEs provided calibrated uncertainty estimates, indicating higher uncertainty when predictions were likely wrong—an advance increasingly important for highly variable data and only achievable with probabilistic approaches.

In the third contribution, we extended these approaches to diffusion-based probabilistic models that enhance sampling fidelity and conditioning flexibility. To preserve low-dimensional neural representations and account for the discrete nature of neural spikes, we introduced Latent Diffusion for Neural Spiking Data (LDNS). LDNS enabled the extraction of behaviorally meaningful neural latents, and training a diffusion model directly on these latents generated realistic spiking data for various tasks. Flexible conditioning on scalars or entire time-series renders LDNS a highly powerful encoding model enabling scalable hypothesis generation.

Collectively, these works develop complementary statistical approaches for continuous neural-behavioral datasets. By integrating classical encoding and decoding with probabilistic deep generative models, this work scales classical analyses to large-scale datasets and highlights the importance of modeling variability and uncertainty.

Zusammenfassung

Ein zentrales Ziel in der Neurowissenschaft ist es, zu verstehen, wie neuronale Aktivität Verhalten hervorbringt. Sogenannte Encoding- und Decoding-Modelle formalisieren diese Beziehung, indem sie bedingte Wahrscheinlichkeitsverteilungen neuronaler Aktivität gegeben eines bestimmten Verhaltens und umgekehrt beschreiben. Fortschritte sowohl in der neuronalen als auch in der Verhaltensmesstechnik ermöglichen es heutzutage, gleichzeitig Tausende Neuronen und komplexes, kontinuierliches Verhalten jenseits starrer Versuchsstrukturen aufzunehmen. Diese Entwicklungen, insbesondere die daraus resultierende hohe Dimensionalität sowie die Variabilität und nichtlineare Beziehungen, stellen klassische Encoding- und Decoding-Ansätze vor Herausforderungen. Ziel dieser Arbeit war daher die Entwicklung statistischer Methoden, die hochdimensionale neuronale Aktivität mit kontinuierlichem Verhalten verknüpfen.

In der ersten Studie haben wir untersucht, ob das Mittelhirn, insbesondere der Colliculus Superior (SC), optische Bewegungen auseinanderhält, die aus Eigen- oder Objektbewegung entstehen. Dies haben wir in Virtual Reality getestet. Dabei waren Mäuse in beiden Kontexten identischen optischen Bewegungen ausgesetzt. Da neuronale Aktivität Verhalten steuert, aber auch selbst von diesem beeinflusst wird, haben wir bei der Decodierung des Kontexts aus neuronaler Aktivität Verhaltensunterschiede berücksichtigt. Mithilfe eines multivariaten diskriminativen Decoding-Ansatzes konnten wir zeigen, dass sich die Aktivität im SC, insbesondere in den mittleren Schichten, zwischen den Kontexten unterscheidet.

In der zweiten Studie haben wir diese wechselseitige Beziehung zwischen neuronaler Aktivität und Verhalten explizit gemacht, etwas, das in separaten Encoding- und Decoding-Modellen oft verborgen bleibt. Unser probabilistisches latentes Variablenmodell auf Basis sogenannter maskierter Variational Autoencoder (VAEs) ermöglicht es, bedingte Wahrscheinlichkeitsverteilungen von neuronaler Aktivität und Verhalten gemeinsam zu modellieren. So konnten wir mit diesem Framework die Verteilung kontinuierlicher Armbewegungen gegeben neuronaler Aktivität modellieren und umgekehrt passende neuronale Aktivität für gegebene Bewegungen generieren. Maskierte VAEs lieferten sogenannte kalibrierte Unsicherheitsabschätzungen, die höhere Unsicherheit signalisieren, wenn Vorhersagen wahrscheinlich fehlerhaft sind—eine, insbesondere bei hochvariablen Daten, zunehmend wichtige Modelleigenschaft, die probabilistische Ansätze voraussetzt.

Im dritten Teil erweiterten wir diese Methoden auf diffusionsbasierte probabilistische Modelle, die präzisere Samples und mehr Flexibilität bei der Konditionierung ermöglichen. Um niedrigdimensionale neuronale Repräsentationen zu bewahren und die diskrete Natur neuronaler Aktionspotentiale zu berücksichtigen, entwickelten wir Latent Diffusion for Neural Spiking Data (LDNS). LDNS ermöglichte

verhaltensrelevante neuronale latente Variablen zu extrahieren. Das Training eines Diffusionsmodells direkt auf diesen latenten Variablen führte zu realistischen Spiking-Samples für verschiedene Datensätze. Die flexible Konditionierung auf Skalare oder gar ganze Zeitreihen macht LDNS zu einem leistungsfähigen Encoding-Modell für skalierbare Hypothesengenerierung.
In dieser Arbeit wurden komplementäre statistische Ansätze für kontinuierliche neuronale und Verhaltensdaten entwickelt. Durch die Integration klassischer Encoding- und Decoding-Methoden mit probabilistischen generativen Modellen ermöglicht diese Arbeit, traditionelle Analysen auf große Datensätze zu skalieren. Sie unterstreicht zudem die Bedeutung der Modellierung von Variabilität und Unsicherheit.

Acknowledgments

Science builds on countless invisible contributions I would like to acknowledge—from open-source communities and anonymous reviewers, to the public institutions that fund and sustain research. Writing these acknowledgments makes me realize how much of my work would not have happened without the many people we never meet but who make scientific progress possible. It is a privilege to be allowed to explore ideas freely and work toward understanding, rather than specific outcomes.

It is a privilege worth protecting, making it even more important to communicate science openly to a broader public and to explain why the scientific process matters. I am deeply grateful to have worked with an advisor who truly recognized the importance of that and encouraged my interest in not only doing research but also science communication.

Thank you, Jakob, for your trust, guidance, scientific ideas, and support throughout my PhD. You took a leap of faith on someone without prior experience in machine learning, remaining confident, even in phases when I was not, that I would succeed in developing deep probabilistic models for neuroscience. From the beginning, you allowed me to explore freely, both scientifically and personally. Your advice to focus on what I find genuinely exciting, rather than trying to meet the expectations of others (including your own), has shaped how I approach research and other aspects of my life, and I am deeply grateful for that. Following you to Tübingen turned out to be the best decision I could have made.

Pedro, I cannot thank you enough for being the kindest, most passionate, and most brilliant scientist one could hope to have as a co-advisor. Since we started our weekly meetings, my enjoyment of science and excitement for and confidence in my work grew so much. Your sharp questions, thoughtful ideas, and patience allowed me to develop as a researcher and strongly shaped how the projects evolved. I am so grateful for your mentorship and for setting an example of the kind of scientist we all should aspire to be.

For many, thesis advisory committee meetings are a formality. For me, the one hour each year with Zeynep Akata, Peter Dayan, and Jakob was extremely helpful and motivating, and provided enough directions and ideas to work on for months. Thank you, Peter, for deeply engaging with my work. Your clarity, also with respect to which directions *not* to pursue or how to refine the title of a PhD thesis within two seconds, is truly exceptional. I would also like to thank Anna Levina, not only for evaluating this thesis but also for her influence from the very beginning. Your enthusiasm for science (incl. my previous research area) and our joint computational

neuroscience journal club made it so much easier and enjoyable to arrive in the Tübingen ML community.

I would also like to thank my fantastic collaborators Stefano Zucca, Sam Solomon, Aman Saleem, Daniel Morales, Michael Berry, Kanishk Jain, Tomaso Muzzu, Gordon Berman, Victor Lobato-Rios, and Pavan Ramdya. Working with you taught me so much about the biological side of our research and the challenges of behavioral and experimental neuroscience—and also humbled me in how much easier we have it on the purely computational side. I am so grateful for the opportunity to experience so many fantastic cross-country and cross-disciplinary collaborations, and to have collaborated with such kind and smart people, something that matters most when some projects never see the light of day. I am also incredibly grateful to Julijana Gjorgjieva and Christoph Miehl for helping me with the final push to write up my master's thesis project after starting my PhD, and for remaining two of my favorite people to look forward to seeing at every conference. Julijana, thank you for continuing to be my mentor. I learned so much from working with you, and for someone like me who does not really believe in role models, you come extremely close.

It is easy to underestimate how much the environment one works in shapes a PhD. The spirit and mission of the Machine Learning for Science Excellence Cluster and the Tübingen AI Center are truly unique in the AI and ML world. I am especially grateful to everyone committed to improving our research environment, let it be the TWiML team for striving to make machine learning more enjoyable and welcoming for everyone or the dedicated people I worked with as part of the AI Center Council initiatives. In general, thanks to the entire A-team who runs and organizes both the cluster and the AI Center. A special thanks to Theresa Authaler and Patrick Klügel for encouraging my science outreach activities from the very beginning.

This brings me to a large group of people to whom I owe so much: the wonderful KI macht Schule crowd—from the early online years, to the Tübingen local group, and the nationwide team these days. I am so grateful to have been immersed in a community that deeply cares about the societal impact of AI and science education, and also one that ensured we would go to the best parties in Tübingen.

This PhD would not have been possible, nor nearly as fun, if not for the wonderful people in the lab! I am so grateful for all the support, the scientific exchange, and daily fun that came with being part of this group. Coming to the office and knowing that there would always be smart, kind, and quirky people around was by far the best part of my PhD. Our *mackelab doesn't judge* playlist perfectly captures how all of you made me feel throughout my years in the lab: like I could truly be myself through all the highs and lows of a PhD. Thank you so much, Poornima, Julius, Zina, Richard, Jan, Janne, Michael, Franzi, Pedro, Artur, Jan-Matthis, Jai, Felix, Guy, Matthijs, Manuel, Linda, Lisa, Sebastian, Cornelius, Daniel, Byoungsoo, Isaac, Lucas, Stefan, Nicolas, Steffi, Alana, Franzi G., Tamara, Rachel, Jonas, Max, Annalena, Meghal, Raees, Muthu, Eszter, Álvaro, and David. Jakob, thank you for bringing together such a wonderful team in every generation. I learned a great deal from

every single one of you. This also includes the exceptional master's students—Cansu, Paul, Jonas, Philipp, Lu, Devank, Jin— I had the privilege to advise and who each have taught me a lot in the process.

A special thanks, of course, goes to our MVP—Franzi. Thank you for not only making our lives easier, but also for making the mackelab such a fun place to work. You managed to turn handling admin tasks into one of the perks of a workday, simply because it meant dropping by your office.

I feel so lucky to have been part of the first Tübingen 2.0 generation and to have found in all of you not only colleagues but really close friends—Poornima, Janne, Michael, Jan, and Richard. Since recency bias often influences acknowledgments, I would like to thank you, Poornima, explicitly. You were the first to patiently explain things to me, from basic concepts like log-likelihoods to complex models. Our breakfasts in the office, long walks, discussions, and fun evenings with the others were definitely the best parts of the challenging time of arriving here during the pandemic. And I am even luckier that the kindness and friendship of the people I shared my office with continued unchanged—thank you, Julius and Zina! Jai, our joint project and working with you was by far the most fun I had working on a project, and I am so proud of what we achieved together. Thank you also to everyone who supported me during the writing process—from Ben Miller for providing his template, and Janne and Julius for holding me accountable during the writing phase, to Jai, Linda, Sebastian, Zina, Richard, and Julius for your feedback.

To my wonderful friends from home, from university, and those I met along the way in science, let it be in our lab, our book club, at Cajal, at conferences, or in our graduate school singing in forests, buses, and beyond—thank you for being absolutely fantastic, for being there when I needed support, and for joining me whenever there was something to celebrate. I would also like to thank my entire extended family, who have been as supportive as any family could have possibly been throughout a PhD and life in general. Thank you, my wonderful parents, for never adding pressure and just supporting me along the way. I know what a privilege that is. Having my PhD overlap with yours, Alix, made everything so much easier, knowing someone knew exactly how I felt and would be empathetic and supportive no matter what. And how lucky I was that my collaborations with UCL allowed me to relive the great year I spent in London during my studies. Thank you, Helene, Stephi, Lotte, and Nora, for making me feel right at home again during these research stays.

And finally, Richard—it is impossible to put into words in how many ways I felt supported by you and how much fun and lightheartedness you brought both into my scientific and overall life. The most important lesson I probably learned during my PhD is from you—to simply embrace stupidity in order to become a better scientist (and human). Embracing stupidity also makes it way easier not to take oneself too seriously, to ask questions freely, and to take every opportunity to "suck a little less" at a task one wants to learn.

Auguste
October 2025

List of Publications

Primary publications

1. Stefano Zucca*, Auguste Schulz*, Pedro J Gonçalves, Jakob H Macke, Aman B Saleem, Samuel G Solomon (2025) Visual loom caused by self-movement or object-movement elicits distinct responses in mouse superior colliculus. *Current Biology.*

2. Auguste Schulz, Julius Vetter, Richard Gao, Daniel Morales, Victor Lobato-Rios, Pavan Ramdya, Pedro J Gonçalves, Jakob H Macke (2025) Modeling conditional distributions of neural and behavioral data with masked variational autoencoders. *Cell Reports.*

3. Jaivardhan Kapoor*, Auguste Schulz*, Julius Vetter, Felix Pei, Richard Gao, Jakob H. Macke (2024) Latent Diffusion for Neural Spiking Data. *The Thirty-eighth Annual Conference on Neural Information Processing Systems.*

Supporting Publications

4. Auguste Schulz*, Christoph Miehl*, Michael J Berry II, Julijana Gjorgjieva. (2021) The generation of cortical novelty responses through inhibitory plasticity. *eLife.*

5. Sebastian Bischoff, Alana Darcher, Michael Deistler, Richard Gao, Franziska Gerken, Manuel Gloeckler, Lisa Haxel, Jaivardhan Kapoor, Janne K Lappalainen, Jakob H. Macke, Guy Moss, Matthijs Pals, Felix C Pei, Rachel Rapp, A Erdem Sağtekin, Cornelius Schröder, Auguste Schulz, Zinovia Stefanidi, Shoji Toyota, Linda Ulmer, Julius Vetter (2024) A Practical Guide to Sample-based Statistical Distances for Evaluating Generative Models in Science. *Transactions on Machine Learning Research.* (Authors listed in alphabetical order.)

6. Jan Boelts*, Michael Deistler*, Manuel Gloeckler, Álvaro Tejero-Cantero, Jan-Matthis Lueckmann, Guy Moss, Peter Steinbach, Thomas Moreau, Fabio Muratore, Julia Linhart, Conor Durkan, Julius Vetter, Benjamin Kurt Miller, Maternus Herold, Abolfazl Ziaeemehr, Matthijs Pals, Theo Gruner, Sebastian Bischoff, Nastya Krouglova, Richard Gao, Janne K Lappalainen, Bálint Mucsányi, Felix Pei, Auguste Schulz, Zinovia Stefanidi, Pedro Rodrigues, Cornelius Schröder, Faried Abu Zaid, Jonas Beck, Jaivardhan Kapoor, David S Greenberg, Pedro J Gonçalves, Jakob H Macke (2025) sbi reloaded: a toolkit for simulation-based inference workflows. *Journal of Open Source Software.*

7. Richard Gao, Michael Deistler, Auguste Schulz, Pedro J Gonçalves, Jakob H Macke (2024) Deep inverse modeling reveals dynamic-dependent invariances in neural circuit mechanisms. *bioRxiv.*

Included in this thesis. * denotes first author equal contribution.

1

Introduction

Interactions of neurons in the brain give rise to a complex repertoire of behaviors across the animal kingdom. Yet, the precise mechanisms by which neural activity drives and is influenced by behavior are still not fully understood, even in simpler organisms. Understanding these mechanisms requires a multidisciplinary approach, combining experimental neuroscience—where hypotheses are formulated and experimentally tested—with theoretical approaches aimed at developing computational theories, models, and data analysis tools.

Computational models in neuroscience span a broad range: from detailed mechanistic models that seek to explain *how* computations are implemented in neurons or neural circuits, to descriptive statistical models that characterize *what* is computed or how modalities are related, to interpretative normative theories that propose *why* neural codes take specific forms [1, 2]. In this thesis, I focus on statistical models that can uncover potential causal relationships, support experimental hypothesis generation, and guide experimental manipulations.

Statistically linking neural activity and behavior is commonly addressed by constructing *encoding* and *decoding* models [3–7]. *Encoding* models allow researchers to investigate which external factors drive neural responses by predicting probable neural activity patterns given behavior, stimuli, or task conditions. In contrast, *decoding* models predict probable behavior, stimuli, or even abstract concepts given neural activity.

Encoding and decoding models take many different forms: from domain-agnostic models such as linear regression or classification to generalized linear models (GLMs) [8], specifically adopted to model neural spike trains [3, 4, 9] and deep neural networks [10–12]. Such statistical approaches have greatly improved our understanding of neural representations for sensory processing [4, 13–17], decision making [18] or movement execution and planning [19, 20]. Although such studies have revealed fundamental principles of neural coding, their restriction to relatively small neural populations and simple tasks provides only a partial view of the brain's distributed

and dynamic representations, which are only revealed in large-scale recordings and analyses.

Yet, recent technological developments now enable recordings from thousands of neurons simultaneously in awake, behaving animals [21–24]. Combined with new data sharing practices, these advances have produced large-scale datasets that often span several brain areas across multiple animals [25–29]. Parallel advances in real-time behavioral tracking now make it possible to record animal behavior in unconstrained lab settings or complex task conditions [30–32]. Together, these advances hold the promise of linking brain activity to ethologically relevant behaviors and moving the field toward the more fundamental goal of understanding the neural basis of naturalistic behavior [33].

Current systems neuroscience is therefore no longer constrained by a lack of rich and large-scale data. The primary challenge now concerns the limited statistical and computational tools available to interpret and make sense of such data. To fully leverage such recordings, we require new computational approaches that scale classical methods for encoding, decoding, and exploratory data analysis to this new regime and embrace the high variability of unconstrained recordings [5, 12, 34–38].

This challenge served as the driving motivation for the work presented in this thesis. I explored three approaches to joint statistical modeling for linking neural and behavioral data at different scales (**Figure 1.1**). To appreciate the motivation behind these newly developed methods, I first revisit the current state of modeling high-dimensional neural and behavioral data, which highlights the need for new approaches and particularly *joint* models of both modalities.

A common first step in processing complex high-dimensional data is dimensionality reduction [39]. For unconstrained behavior, recent methods demonstrated that seemingly complex behaviors can often be captured by a few stereotyped motifs [40–43]. Analogously, high-dimensional neural data can often be reduced to a small number of dimensions that capture a substantial fraction of neural variability [34, 39]. Principal Component Analysis (PCA, [44]) and neuroscience-specific extensions such as demixed PCA [45] are widely used for neural dimensionality reduction but struggle when some features are highly variable or only partially recorded. These limitations motivated the development of various probabilistic latent variable models (LVMs) which have provided insights into the intrinsic, stimulus- or behavioral-state-dependent dimensionality of highly variable neural populations [46–60].

While the progress on statistical models for neural and behavioral data, respectively, has been remarkable, probabilistic LVMs that *jointly* model high-dimensional neural activity and simultaneous, possibly unconstrained behavior are still rare. In recent years, there have been some developments toward this goal [12, 59, 61–65]. Yet, there is little evidence, nor was it the focus of the respective works, of fully capturing the possible bidirectional nature of the statistical relationships when linking neural and behavioral data.

Fitting joint models and capturing bidirectionality—how neural activity is influenced

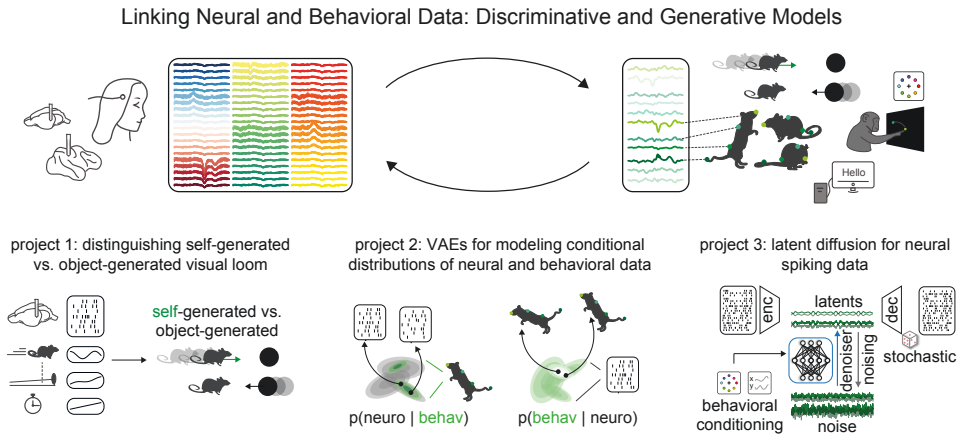


Figure 1.1: Thesis overview. Linking Neural and Behavioral Data: Discriminative and Generative Models. Panels adapted from Zucca, Schulz et al. (2025), Schulz et al. (2025), and Kapoor, Schulz et al. (2024).

by behavior while driving behavior—becomes increasingly important with multi-modal, unconstrained datasets. First, because joint models can explicitly account for confounds introduced by spontaneous, uninstructed behaviors that modulate neural activity across the brain, which become more prominent in such datasets [66–68]. Second, if we were to link separately obtained low-dimensional neural and behavioral representations post-hoc, relevant shared information would likely be lost and possibly hidden in neural-only and behavioral-only representations. Thus, joint models hold the promise of yielding more informative shared representations [12, 59, 61]. Third, when combined with generative capabilities, joint models enable probing learned statistical relationships and temporal neural-behavioral dependencies. Ideally, such models would allow generating likely behaviors given a neural activity trace, or likely neural activity patterns given some behavior. Visually inspecting synthetic data generated given certain conditions, and after directed manipulations of those conditions, will likely provide new insights when mechanistic or experimental manipulations are not possible, expensive, or challenging to perform.

Existing encoding, decoding, and LVM approaches only partially fulfill these promises. Thus, we require models that not only capture statistical dependencies but also allow sampling synthetic data from the learned distributions. Deep generative models provide such a framework. Methods such as Variational Autoencoders (VAEs; [69, 70]), Generative Adversarial Networks (GANs; [71]), and more recently Denoising Diffusion Probabilistic Models (DDPMs; [72, 73]) have been successful across domains in generating realistic samples of diverse data. Leveraging such methods for joint neural and behavioral data analysis is the main contribution of this thesis work.

The overarching goal of the work presented in this thesis was to assess how we can use and further develop machine learning methods to link potentially high-dimensional

neural population activity with complex external modalities, such as behavior or experimental context. The first project addresses the question of how to jointly model neural activity, the behavior of mice moving freely in a virtual reality corridor, and other covariates to account for confounding factors in a discriminative decoding task. Setting up a multivariate model that incorporates both neural and behavioral features, we demonstrated that activity in the superior colliculus differentially represents visually identical looming stimuli caused by either self-movement or object-movement (**Figure 1.1**, left). For the second project, I developed a unified latent variable model based on variational autoencoders that enables us to capture and generate samples from both the encoding *and* decoding distributions within a single model, thereby bridging the gap between classical encoding and decoding and powerful deep generative models (**Figure 1.1**, center). For the third project, we asked if we could leverage denoising diffusion probabilistic models to generate even more realistic neural spiking data conditioned on behavior, while preserving access to an interpretable latent space. To this end, we developed Latent Diffusion for Neural Spiking Data (LDNS), a flexible and powerful latent variable model that can generate both realistic unconditional samples and samples from the encoding distribution (**Figure 1.1**, right). Together, these three projects illustrate complementary strategies for building joint discriminative and generative models of neural and behavioral data.

Outline of this thesis First, I outline the background required to understand and contextualize the developed methods in Chapter 2. Next, in Chapter 3, I introduce the three publications that build the basis of this thesis. For each publication, I provide a more detailed motivation, summarize the methods and results, and declare my respective contributions. Finally, I discuss the overarching themes and limitations of the three studies in Chapter 4.

2

Background

In presenting the background of this thesis, I will assume that the reader is familiar with the fundamentals of neuroscience, machine learning (ML), and deep learning (DL), as covered in standard references such as Bishop [77], Kandel [78], and Goodfellow, Bengio, and Courville [79].

Thus, I do not review detailed foundations of supervised and unsupervised learning, which are the two primary branches of machine learning underlying the methods presented in this thesis. Briefly, supervised learning relies on datasets with associated labels or target modalities that provide a supervisory signal for model training. In contrast, unsupervised learning aims to uncover patterns and structures in data without explicit labels. I also assume familiarity with training deep neural networks, e.g., gradient-based optimization techniques [77, 79]. In terms of neuroscience background, I expect the reader to be familiar with action potentials (often referred to as "spikes"), and the general structure of neuroscientific experiments [78].

Throughout the background section, data are denoted with the variable x. I will refer to neural data as x^n and behavioral data as x^b. Latent, unobserved variables are denoted with z. The entire dataset is denoted with capital letters. In all examples considered here, neural activity x^n refers to electrophysiological recordings and usually takes the shape of spike counts per time bin. Most notations hold across different recording modalities, yet with different observation models (e.g., Gaussian observations rather than Poisson).

2.1 Statistical Approaches for Linking Neural and Behavioral Data

Since the late 19th century (e.g., Fritsch and Hitzig [80]), clinicians, neuroscientists, and statisticians have attempted to link recorded neural activity to corresponding behavior. While *decoding* models (**Figure 2.1**A) allow inferring if information about a particular behavior or stimulus is present in and decodable from neural data,

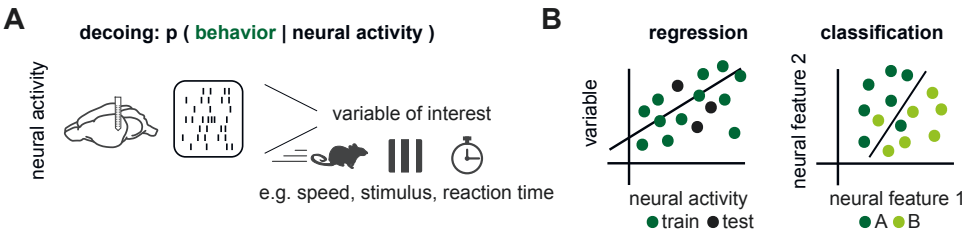


Figure 2.1: Decoding models: can we decode information about a particular behavior or stimulus from neural data? A. Schematic of decoding models that map neural activity to a target variable, such as behavior or stimulus identity. Schematics adapted from Zucca, Schulz et al. (2025). **B.** Common linear decoding approaches include linear regression, which predicts continuous variables from neural activity (left) and linear classifiers, which separate discrete classes (here class A and B) based on neural features (right).

encoding models (**Figure 2.2**A) allow studying if certain covariates are predictive of neural activity [4–7].

In this section, I focus on providing the basis for linear encoding and decoding models and corresponding techniques for optimizing their model parameters.

2.1.1 Decoding Models

Linear Regression based Decoding

One of the simplest and most commonly used methods to decode behavior from neural activity is linear regression (**Figure 2.1**B, left) or regularized versions thereof, in particular ridge-regression [81, 82]. In the LDNS project, we employ such a ridge-regression approach to map smoothed neural firing rates to hand velocities during reaching.

The underlying assumption of linear regression is that neural activity and behavior are linked through an affine function (linear mapping plus bias) from neural activity to behavior, with Gaussian observation noise:

$$p(x^b|x^n) = \mathcal{N}(x^b;\ Wx^n + b, \sigma^2 I), \tag{2.1}$$

where W is the weight matrix, mapping neural activity to behavior, b is the bias term, and σ^2, the variance of the Gaussian noise model. Given N data pairs of neural activity and behavior, we can fit the parameters of the model W, b, and σ^2 using Maximum Likelihood Estimation (MLE). In this case, optimizing is easier after first taking the logarithm (log-likelihood):

$$\log p(X^b|X^n, W, b) = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}[x_i^b - (Wx_i^n + b)]^2. \tag{2.2}$$

Note that the Gaussian noise term σ^2 is challenging to optimize and often not learned explicitly [83]. In the linear Gaussian model, the solution for finding the optimal W

is equal to minimizing the mean squared error between the predicted and observed behavior.

To obtain more robust results and models that are less prone to overfitting, regularization techniques such as ridge-regression (L2) can be used.

Classifier Based Decoding

There is a variety of linear models for classification, ranging from simple discriminant functions to Bayesian approaches. All of these approaches have the goal of assigning a class variable to an input vector x (**Figure 2.1**B, right; [77]). Here, we focus on a commonly used probabilistic discriminative model for classification—logistic regression, which we used for the context discrimination project, to discriminate between two contexts.

For binary outcomes y, such as the two contexts, denoted with 0 or 1, we describe the probability of context 1 as

$$P(y = 1|x^n) = \sigma\left(\beta_0 + \beta^n \cdot x^n\right), \tag{2.3}$$

where σ is the sigmoid function $\sigma(y) = \frac{1}{1+e^{-y}}$, β_0 the intercept, and β^n the coefficient vector for the neural data. Here, we introduced the case where only neural data is used for classification, but the extension to additional predictors, such as behavior in the logistic regression framework, is straightforward and only requires the addition of additional terms within the sigmoid, such as $... + \beta^b x^b$. The log-likelihood for logistic regression is

$$\mathcal{L}(\beta) = \sum_{i=1}^{N} \left[y_i \log P(y_i|x_i) + (1 - y_i) \log(1 - P(y_i|x_i))\right]. \tag{2.4}$$

Due to the sigmoid function, there is no closed-form solution for optimizing the parameters of the model, and more complex iterative optimizers are required. Once trained, a decision boundary classifies a test datum into either class.

Logistic regression is an example of a broad class of models known as generalized linear models (GLM, Nelder and Wedderburn [8]), which also play an important role for neuroscientific *encoding*.

2.1.2 Encoding Models

In the same vein as for decoding models, I first focus on simple encoding models prior to moving toward deep generative models, which target the encoding distribution.

Generalized Linear Models

As already demonstrated in the logistic regression case, GLMs are not to be mistaken with linear models such as linear regression. The general building blocks are a linear transformation of the input $\vec{x}$ with input-specific weights $\vec{\theta}$ (often referred to as filters), a link function f that can be non-linear, followed by a noise process [8].

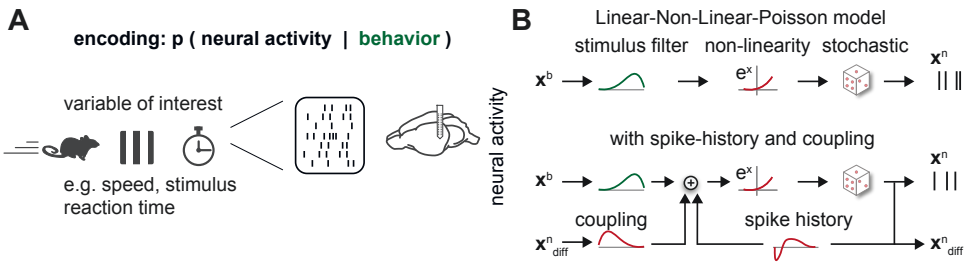


Figure 2.2: Encoding models: is information about a stimulus or behavior encoded in neural activity? **A.** Schematic of simple encoding models that predict neural activity from external stimuli, behavior, or other experimental variables. Schematics adapted from Zucca, Schulz et al. (2025). **B.** Top: Linear-Non-Linear-Poisson Models compute the firing rate of a Poisson neuron by convolving the input stimuli or behavior, x^b, with a linear filter and passing it through a non-linearity, in this case, the exponential function. Bottom: More complex version of such a GLM with coupling filters accounting for input from other neurons and the neuron's spike history, adapted from Pillow et al. [4].

Thus, GLMs are well suited for neural encoding models and probabilistic mappings from stimuli or behavior to neural activity (**Figure 2.2**A). In neuroscience, the noise process is often some adjusted version of a Poisson process to account for the discrete nature of neural spiking activity [3, 4, 8, 9]. In its simplest form, such a Linear-Non-Linear-Poisson Neuron model consists of a stimulus filter (**Figure 2.2**B, top), followed by an exponential non-linearity and Poisson noise, i.e., transferred to our neural and behavioral example $x^n \sim \text{Poisson}(f(k^\top \cdot x^b))$, where $f(x) = e^x$.

Such a simple neuron model, however, will not be able to reflect many characteristics of the spiking activity of biological neurons, which are embedded within interacting neural circuits [4]. Not only network effects but also neural dynamics that are "private" to individual neurons are not reflected, for example, refractoriness or burstiness [4, 9, 48]. GLMs are successful precisely because they can account for such effects while maintaining tractability and interpretability through adaptations of the simple model [4, 9, 84]. Each additional input, e.g., the activity of other neurons within a population, or the neuron's own history, gets transformed with respective filters and can influence the predicted rate (**Figure 2.2**B, bottom).

The conditional firing rate of a neuron given this generalized model accounting for behavior, spike-history, and coupling to other neurons is modeled as:

$$\lambda(t) = f\left(k^\top x^{\mathrm{b}}(t) + h^\top x^{\mathrm{n}}(t) + c^\top x^{\mathrm{n}}_{\mathrm{diff}}(t) + b\right), \tag{2.5}$$

where k, h, and c are the linear filters for the stimulus, the spike history, and coupling to other neurons.

For the LDNS project, we took inspiration from GLMs and extended our deep generative model using spike-history filters, which increased the realism of generated spiking activity. For many tasks, modern deep-learning-based approaches have been found to be more accurate than classical GLMs since they are more expressive and

can approximate arbitrary non-linear relations [7, 82, 85]. Yet, classical methods have several advantages over deep neural network models: 1) They are easier to use and interpret. 2) Optimization of these linear or generalized linear models reaches a unique global minimum due to the convex nature of the loss function. 3) Particularly for smaller or noisy datasets, they are less prone to overfitting.

2.1.3 Challenges with Increasing Dataset Complexity

As the dimensionality of data increases and as datasets become more diverse, modeling them becomes increasingly challenging, and linear relationships may no longer be sufficient to capture the statistical dependencies. If the data is high-dimensional (e.g., many neurons recorded simultaneously) but limited in sample size, even simple models can become prone to overfitting [79]. Furthermore, optimization procedures, such as those used in linear regression, require the inversion of a matrix, which can be computationally unstable when the number of features exceeds the number of samples. Thus, in order to deal with such datasets, alternative methods are required that aim to first distill simpler representations of and uncover structure in high-dimensional data.

2.2 Uncovering Structure in High-dimensional Neural and Behavioral Data

Not just in neuroscience, but in various scientific domains, ML methods have become necessary to uncover structure in high-dimensional data. This is often addressed with visualization tools, for example, non-linear embedding approaches such as UMAP [86] or t-SNE [87]. Such methods have successfully unveiled structure in unconstrained behavioral, [e.g., 40] and neural data [e.g., 88]. However, it often remains challenging to judge what true clustering in the neuronal or behavioral embedding space is or if a presumed structure is merely an artifact of the algorithm—a problem that has driven the development of neuroscience-specific visualization techniques [89]. Here, we focus on different forms of dimensionality reduction, less targeted at visualization, but at capturing lower-dimensional structures underlying high-dimensional neural and behavioral data.

2.2.1 Dimensionality Reduction

The goal of dimensionality reduction is to reduce the data dimensions while maintaining essential features. This makes the data easier to handle in terms of compute and storage, easier to visualize, and, in many cases, more accessible for downstream analyses by distilling useful representations. In the last three decades, with the increase in the number of simultaneously recorded neurons, dimensionality reduction has been established as a standard of neural data analysis. There has been ample evidence that a large fraction of neural variability can be described by a few underlying latent dimensions and that some patterns hidden in single-neuron activity become apparent when analyzing neural activity at the population level [19, 39, 47, 90–92].

Compression Using Autoencoders

A popular deterministic method to reduce the dimensionality of neural data is through deep learning-based autoencoders. An autoencoder learns to map inputs to reconstructions that closely reproduce the original data, while compressing the representation by requiring the information to pass through a so-called bottleneck layer [79]. The encoder maps the data to the lower-dimensional representation $z = \texttt{encoder}(x)$, and the decoder maps it back up to data space $x = \texttt{decoder}(z)$. Both encoder and decoder are commonly parameterized through deep neural networks. To prevent overfitting and or enforce smoothness in the latent space, one can additionally regularize the autoencoder during training, e.g., with L_2 regularization, or by applying smoothness priors over neighboring data points.

For neural spiking activity, it has been shown that additional regularization techniques are required to prevent autoencoders from incorrectly predicting highly localized sharp firing rates, reflecting individual spike times [93]. This can be mitigated by coordinated dropout as proposed by Keshtkaran and Pandarinath [93]: random masking of input spikes and computing the loss on the predicted rates at the masked locations. For the LDNS project, we use such a regularized autoencoder approach to map high-dimensional neural activity to low-dimensional smooth latents.

Principal Component Analysis

Arguably, the most widely used dimensionality reduction method in systems neuroscience is Principal Component Analysis (PCA) [39, 94, 95], originally introduced by Pearson [44]. We used PCA in all three projects presented in the thesis for some stage of the analysis, yet with very different purposes. PCA is a linear method that identifies orthogonal directions in the data that capture the largest variance—called principal components.

Given centered data $X \in \mathbb{R}^{N \times D}$, where N is the number of samples, e.g., different trials, and D the number of features, e.g., the number of recorded neurons, PCA identifies the principal components $\{\mathbf{u}_1, \ldots, \mathbf{u}_D\}$ which minimize the projection error, or equivalently such that the projected variance is maximized. For the first principal component, we solve

$$\mathbf{u}_1 = \arg\min_{\|\mathbf{u}\|=1} \sum_{i=1}^{N} \left\|\mathbf{x}_i - \mathbf{u}\mathbf{u}^\top \mathbf{x}_i\right\|^2 \tag{2.6}$$

$$= \arg\max_{\|\mathbf{u}\|=1} \mathbf{u}^\top \left[\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^\top\right] \mathbf{u} \tag{2.7}$$

$$= \arg\max_{\|\mathbf{u}\|=1} \mathrm{Var}(X\mathbf{u}). \tag{2.8}$$

The data projected onto the top d components is given by $X_{\mathrm{PCA}} = XW_d$, where $W_d \in \mathbb{R}^{D \times d}$ is a matrix whose columns are the first d PC directions.

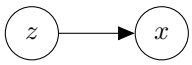
PCA applied to neural and behavioral time series has revealed underlying low-dimensional manifolds and has enabled informative visualizations [39, 91, 96]. How-

ever, one of the major drawbacks of PCA, especially relevant in the neuroscientific context, is that noisy variables with high variance can receive disproportionately high weight. This can lead to misleading components that reflect noise rather than the true underlying structure. This limitation is overcome by probabilistic latent variable models such as factor analysis, which explicitly separate shared underlying signals from independent noise [77].

2.2.2 Latent Variable Models

Modern latent variable models (LVMs) in neuroscience differ in their assumptions about latent dynamics (static, linear, non-linear, or switching), the complexity of the mapping function, and the observation noise model, and have gradually become more capable of modeling complex neural and behavioral data [41–43, 82, 97]. Here, we start with the general Bayesian inference framework and the principles of probabilistic inference of unobserved (latent) variables.

Consider the following graphical model, where latent factors z give rise to observed data x:



In the Bayesian framework, both latent variables z and observed data x are modeled as random variables. The prior over latent variables $p(z)$ incorporates our belief about how the latents are distributed before observing data x. The likelihood $p(x|z)$ specifies the probability of observing the data given some specific z. The goal is to compute the posterior distribution $p(z|x)$ given data. Following Bayes' theorem, the posterior is given by

$$p(z|x) = \frac{p(z)p(x|z)}{p(x)}, \tag{2.9}$$

where the marginal likelihood $p(x)$, also referred to as evidence, is computed by marginalizing over z, $p(x) = \int p(z,x)dz = \int p(z)p(x|z)dz$. Integrating over all possible latent states, however, is often not possible, in particular for complex models, making this integral intractable. Thus, for complex relationships, we rely on variational inference to approximate the true posterior with simpler distributions. For some models, such as the linear Gaussian Latent Variable Model (GLVM), exact inference is possible, which is the model assumption underlying Factor Analysis [77].

Factor Analysis

Concretely, the assumption when fitting a Factor Analysis model to data is that the generative model maps linearly from the latents z, which are drawn from a Gaussian distribution, to Gaussian distributed data x:

$$\begin{aligned} p(z) &= \mathcal{N}(z; \mu_z, \Psi) \\ p(x \mid z) &= \mathcal{N}(x; Cz + d, \Lambda), \end{aligned} \tag{2.10}$$

where Λ and Ψ are diagonal covariance matrices for data and latents. These assumptions allow calculating the posterior distribution, which is also Gaussian, in closed form if the model's parameters are known [77].

In most cases, however, we neither have access to ground truth latents nor the parameters. Thus, the Expectation Maximization (EM) algorithm [98] is often used to fit FA models. EM iteratively optimizes the parameters of the model by maximizing the expected data log-likelihood and inferring the latents from data given the current best guess of the parameters [77].

Extending the factor analysis model to the time domain, we arrive at state-space models, which model how unobserved latents evolve over time.

State Space Models

State-space models (SSMs) are particularly relevant in neuroscience, as they provide a framework for uncovering the temporal evolution of latent dynamics underlying neural and behavioral time series [90]. The Gaussian linear SSM, in which state transitions and observations are linear, can be considered as many FA models chained together. For discrete-time SSMs, we get the following state equations

$$z_{t+1} = Az_t + w_t, \quad w_t \sim \mathcal{N}(0, \Psi) \tag{2.11}$$

$$x_t = Cz_t + v_t, \quad v_t \sim \mathcal{N}(0, \Lambda), \tag{2.12}$$

where A is the state transition matrix, C is the observation matrix and v_t and w_t are Gaussian noise samples with zero mean and covariances Ψ and Λ. The Markov property is assumed for tractable inference: the current state z_t depends only on the previous state z_{t-1}, and not on earlier time points. Inference in linear Gaussian SSMs can be performed efficiently using the Kalman filter (KF), a recursive algorithm that computes the posterior distribution over latent states given noisy observations [99]. The KF has been widely used for inferring latent states from neural recordings, and in combination with EM, for fitting state-space models to such data.

Linear Gaussian SSMs, however, are inherently limited in their expressiveness, especially in capturing non-linear and long-range dependencies common in neural systems. To address this, more expressive models like recurrent neural networks (RNNs) have become popular alternatives to model neural dynamics [55, 56].

Recurrent Neural Networks For Non-Linear Dynamics

Recurrent Neural Networks (RNNs) extend classical sequence models by allowing the hidden state h to evolve non-linearly over time. The standard (or "vanilla") RNN update takes the form:

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b), \tag{2.13}$$

where h_t is the hidden state at time t, x_t is the input, and σ is a non-linearity such as $tanh$ or $ReLU$. These models can, in principle, approximate arbitrary dynamical systems given sufficient data and model capacity [79, 100–102].

However, vanilla RNNs often suffer from training instabilities due to the problem of exploding or vanishing gradients when modeling long sequences [103, 104]. To mitigate this, Gated Recurrent Units (GRUs; [105]) and Long Short-Term Memory (LSTM) networks [104] introduce gating mechanisms that regulate the propagation of information through time. These gates help preserve stable gradient propagation and have become the standard in many practical applications of RNNs.

Despite various advances, RNNs remain limited in their computational efficiency, as their inherently sequential structure prevents parallelization.

Structured State-Space Models for Efficient Long-Range Modeling

Structured State-Space models (S4; [106]) offer an alternative approach that retains the interpretability of linear dynamical systems while achieving the scalability and expressiveness of deep neural networks. In short, S4 layers are based on state-space transition equations that are parameterized to enable efficient computation of convolutional kernels over arbitrarily long sequences. This structure enables parallelization, overcoming some of the bottlenecks faced by traditional RNNs. Thus, we included S4 layers as the base sequence model in the LDNS project.

For an input sequence of length T, $u = [u_1, \ldots, u_T] \in \mathbb{R}^{T \times d_{\text{input}}}$ and output $x = [x_1, \ldots, x_T] \in \mathbb{R}^{T \times d_{\text{output}}}$, an S4 layer is based on the following discrete-time state-space equations

$$z_t = \overline{A} z_{t-1} + \overline{B} u_t \tag{2.14}$$

$$x_t = C z_t, \tag{2.15}$$

where $z_t \in \mathbb{R}^n$ is the latent state, and $\overline{A} \in \mathbb{R}^{n \times n}$, $\overline{B} \in \mathbb{R}^{n \times d_{\text{input}}}$, and $C \in \mathbb{R}^{d_{\text{output}} \times n}$ are learned discrete-time state transition, input, and output matrices. $\overline{A}$ and $\overline{B}$ are obtained by discretizing the continuous-time analogues using bilinear discretization [73, 107]. In practice, S4 applies such state-space dynamics independently to each feature dimension, i.e., for each input channel d, a separate transition matrix $\overline{A}^{(d)} \in \mathbb{R}^{n \times n}$ is defined which acts on a per-dimension latent state. This structure corresponds to a diagonal or block-diagonal state-space model, enabling the efficient parallel computation of S4 for multi-channel signals.

Because the core S4 block is linear in both input and state, the recurrent dynamics can be expressed as a 1D convolution over the input sequence, which enables efficient training and inference on long sequences. Unlike classical linear state space models, S4 is deployed as a layer within deep neural networks with non-linear transformations, making deep S4 architectures highly expressive and capable of modeling complex, non-linear temporal dependencies [106]. Both of these properties make them suitable components in generative probabilistic models for neural and behavioral time series.

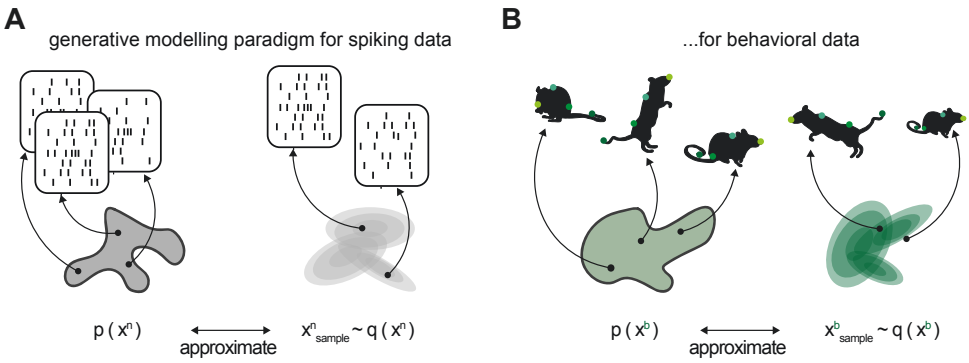


Figure 2.3: Generative models aim to approximate the data distribution and allow sampling of realistic synthetic data A. Schematic of the generative modeling paradigm for neural population spiking data. Sampled spiking data resemble, but are not exact copies of, the true spiking data distribution. **B.** Same as A. but for behavioral pose sequences of freely moving animals. Schematics adapted from Schulz et al. (2025).

2.3 Generative Modeling of Neural and Behavioral Data

Generative models have long been at the heart of scientific inquiry across various disciplines [see 108, for examples across different domains]. Mechanistic models in neuroscience build upon known properties of, for example, neural dynamics, circuits, or bodily constraints. Statistical generative models of neural and behavioral data describe the statistical dependencies, often without any mechanistic assumption about the data generation process [109]. Here, we focus on such generative models, particularly flexible deep learning-based approaches.

2.3.1 Generative Modeling Paradigm

The generative modeling paradigm describes the process of approximating a true data distribution using a statistical model. We start from some data samples x that, in the neuroscientific context, could be time segments of population spike trains, images, behavioral time series, etc. from the data distribution $p(x)$. The goal of a generative model is to approximate this distribution with $q(x)$, i.e., $q(x) \approx p(x)$, such that when sampling from q, the samples look as if they came from the original data distribution (see two neuroscientific and behavioral depictions of the generative modeling paradigm in **Figure 2.3**).

One such model class is the variational autoencoder, which forms the basis of our work on unifying encoding, decoding, and neural and behavioral dimensionality reduction.

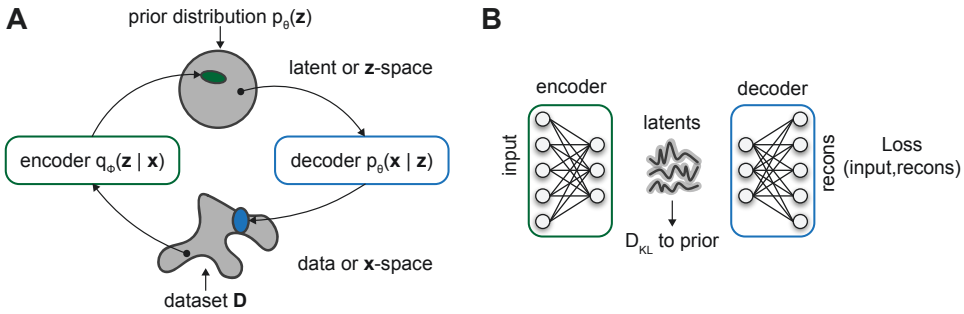


Figure 2.4: Variational Autoencoders B. VAEs map from one distribution to another, e.g., the transformation from data to latent space and vice versa, by approximating the generative and inference models. Adapted from Kingma and Welling [110]. **A.** Schematic of a VAE encoder-decoder neural network-based architecture with the reconstruction loss and regularization term for the latent distribution: the Kullback-Leibler divergence between the approximate posterior $q_\theta(z \mid x)$ and the prior $p(z)$. Schematics adapted from Schulz et al. (2025).

2.3.2 Variational Autoencoders

Overview Variational autoencoders (VAEs) are probabilistic generative models that can capture complex multi-modal data distributions [69, 70]. They are a class of probabilistic latent variable models that assume that most variability of high-dimensional data can be explained by a smaller set of latent variables. VAEs learn stochastic mappings between data and latent space (**Figure 2.4** A). Both mappings, data to latent, and vice versa, are typically parameterized through flexible neural networks (**Figure 2.4** B). Once trained, one can generate new synthetic data by sampling from the specified prior distribution and passing this sample through the probabilistic decoder (**Figure 2.4**).

Model details The generative model is described by the joint distribution of data x and latent variables z, which factorizes into

$$p_\theta(x, z) = p_\theta(z)p_\theta(x|z), \tag{2.16}$$

with parameters θ, prior over latents $p_\theta(z)$ and the probabilistic decoder, $p_\theta(x|z)$ (**Figure 2.4**, blue). The encoder $q_\phi(z|x)$ with parameters ϕ (**Figure 2.4**, green), maps from data to latent distributions and approximates the intractable posterior [69, 70, 110]. VAEs are trained by maximizing the lower bound of the data log-likelihood, the so-called Evidence Lower Bound (ELBO):

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \mathrm{D}_{\mathrm{KL}}\left[q_\phi(z|x)||p_\theta(z)\right]. \tag{2.17}$$

All parameters, θ and ϕ, can be jointly optimized via stochastic gradient descent. In the standard formulation, the prior and approximate posterior are Gaussian, though alternative parameterizations are often required to model more complex data distributions [111, 112].

One of the key contributions of both Rezende, Mohamed, and Wierstra [70] and Kingma and Welling [69] was the introduction of the reparameterization trick. The

stochastic nature of the latent variable z would, in its standard form, prevent gradients from being propagated during training. The reparameterization trick addresses this by expressing the latent variable z as a deterministic transformation of the data-dependent parameters ($\mu_\phi(x)$ and $\sigma_\phi(x)$ for the Gaussian case) and an external source of randomness, allowing gradients to propagate through the network.

Model extensions Since their inception more than 10 years ago, VAEs have been successfully extended to diverse time series data [113–115] and have strongly influenced latent variable modeling in neuroscience [43, 55, 56, 64]. By adjusting the observation model and corresponding likelihood function, VAEs can handle various data types and even heterogeneous combinations thereof [116], making them well-suited for joint models of neural and behavioral data [64, 116].

For Gaussian observations, the corresponding Gaussian-negative-log-likelihood for datum x given the model's prediction μ and standard deviation σ is given by

$$\mathcal{L}_{\text{GNLL}}(x;\mu,\sigma) = -\log P(X=x;\mu,\sigma) = \frac{1}{2}\log(2\pi\sigma^2) + \frac{(x-\mu)^2}{2\sigma^2}. \tag{2.18}$$

For discrete count data, e.g., applicable for neural spiking data, the Poisson-negative-log-likelihood for observed count x and predicted rate λ is given by

$$\mathcal{L}_{\text{Poisson}}(x;\lambda) = -\log P(X=x;\lambda) = -x\log(\lambda) + \lambda + \log(x!). \tag{2.19}$$

Our VAE work combines both of these losses for neural spiking and behavioral data, respectively. As discussed above in section 2.1.2, Poisson observation models of VAEs applied to neural time series will not account for neural dynamics such as refractoriness, oscillations, or bursting behavior that are not mediated through the latent state. To this end, some LVM approaches for neural data have incorporated more expressive observation models, which often require elaborate optimization techniques [52, 57].

In recent years, Denoising Diffusion Probabilistic Models (DDPMs) have emerged as powerful generative models across many domains [73] and form the foundation of our LDNS project. Although their formulation differs significantly from that of VAEs, DDPMs can be understood as hierarchical LVMs, and in particular as extensions of VAEs with a sequence of latent variables [117].

2.3.3 Denoising Diffusion Probabilistic Models

Overview In contrast to VAEs, DDPMs in their standard formulation do not incorporate dimensionality reduction. The primary focus is data generation that can be flexibly conditioned on external variables or a subset of the data.

Diffusion probabilistic models were first proposed by Sohl-Dickstein et al. [72] and inherit their name from diffusion processes in physics as they gradually destroy the structure in a data distribution by systematically adding noise (**Figure 2.5**A). Learning to reverse this process by denoising the corrupted data lets diffusion

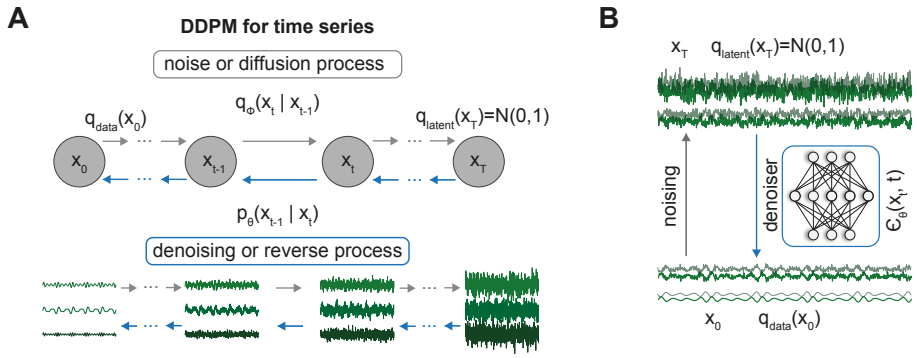


Figure 2.5: Denoising Diffusion Probabilistic Models A. DDPMs mimic a diffusion process by gradually adding noise, corrupting the original data, and then learning to reverse this process by training a denoiser network that aims to predict the noise for removal. Adapted from Ho, Jain, and Abbeel [73] **B.** The denoiser is trained to predict the noise at each time step. This allows for the generation of new data by gradually denoising an initial Gaussian noise sample during generation time.

models act as generative models. Once optimized, one can gradually remove noise from a purely random noise sample, progressively transforming it to look like data drawn from the original data distribution. Most notably, Ho, Jain, and Abbeel [73] improved upon Sohl-Dickstein et al. [72] by generalizing the approach to deep learning architectures and demonstrating high-quality sampling capabilities. Similar to the original formulation, the noise schedule (i.e., the amount of noise added at each step) is fixed and known. The denoising process is learned by training a neural network to predict the noise at each time step (**Figure 2.5**B), which enables the progressive removal of noise during generation.

Model details More concretely, for some data x_0 with subscript 0 indicating the 0th diffusion timestep, we first produce a noised version of the datum by adding Gaussian noise (**Figure 2.5**A, grey arrows). We assume a Markov process, and since all noising steps add Gaussian noise, we can describe the distribution of noised versions given the original data at any diffusion timestep t as

$$q(x_t|x_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I\right), \tag{2.20}$$

where $\bar{\alpha}_t = \prod_{k=1}^{t} \alpha_k$ is the product of all noise scaling factors α_1 to α_t and I is the identity matrix.

In order to denoise the data in the DDPM framework (**Figure 2.5**A, blue arrows), a deep neural network is trained to approximate the reverse process $p_\theta(x_{t-1}|x_t)$ for each diffusion timestep. This is similar to trying to infer the latent distribution from data $p_\theta(z|x)$ in the VAE framework at every timestep. Similarly, the true reverse or denoising process $q(x_{t-1}|x_t)$ is intractable, and training diffusion probabilistic models also relies on variational inference and approximating these distributions with

Gaussians. Training DDPMs can thus be understood as maximizing a variational lower bound (ELBO) on the data log-likelihood, analogous to VAEs.

We train the neural network $\mu_\theta(x_t, t)$, which takes as input both the noisy sample x_t and an embedding of the diffusion timestep t, to approximate the mean of the conditional reverse transition distribution $q(x_{t-1}|x_t, x_0)$ by optimizing the loss $\mathbb{E}_{x_0\sim\mathcal{D}_z,\epsilon_0,t}\|\epsilon_\theta(x_t, t) - \epsilon_0\|^2$, where ϵ_0 is the noise used to generate x_t from x_0, and $\epsilon_\theta(x_t, t)$ is the network's prediction of that noise. Training the model to recover ϵ_0 is mathematically equivalent to predicting the mean of the reverse Gaussian distribution $\mu_\theta(x_t, t)$, but leads to a simpler and more stable optimization target [73]. The embedding of t provides the network with information about the current noise level, which is essential for accurately predicting the denoising direction at different diffusion steps. At test time, we sequentially sample x_{t-1} given x_t using the learned transition $p_\theta(x_{t-1}|x_t)$, starting from standard Gaussian noise.

Model extensions Since DDPMs were introduced in 2020, they have not only influenced computer vision but have been successfully extended to other domains, e.g., natural language processing [118, 119], or time series, e.g., [120–122].

In the LDNS project, we developed a latent variable approach for neural spiking data based on time series DDPMs. For a sequence of length T, the model begins with a Gaussian noise vector of the same length and progressively transforms it into data-resembling time series (**Figure 2.5**). To handle temporal data effectively, the network architecture must incorporate suitable inductive biases [122, 123]. Common choices include convolutional networks, those incorporating RNNs, transformers, or structured state space models, as introduced above.

Conditioning Just as text prompts can condition image generation [124, 125], DDPMs for neuroscientific time series can be conditioned on auxiliary variables to guide generation, e.g., stimuli, behavioral sequences, task structure, or categorical labels. In general, conditioning modifies the reverse process $p_\theta(x_{t-1}|x_t, c)$, where c is the conditioning variable. To this end, various architectures and algorithms have been proposed, including concatenation, cross-attention, or guidance-based mechanisms [73, 122, 126, 127].

For LDNS, we focused on practical and simple conditioning approaches for the neuroscientific context: First, scalar conditioning: Scalar variables such as reach angle or trial type can be embedded and combined with the time embedding (e.g., via concatenation or addition) at each step. Second, time series conditioning: When conditioning on another temporal signal (e.g., behavioral velocity traces), the conditioning sequence can be concatenated with the input or latent sequence as additional channels. During generation, missing channels can be imputed by conditioning on the observed subset [122], enabling the generation of neural activity conditioned on behavioral time series.

While DDPMs are powerful, they are computationally expensive when applied directly to high-dimensional data. Latent Diffusion Models (LDMs) address this

by performing the diffusion process in a lower-dimensional latent space [128]. In neuroscience, where LVMs are widely used, LDMs are particularly promising, as they can improve efficiency while maintaining high generative quality.

2.3.4 Latent Diffusion Models

LDMs use two-stage training frameworks that combine dimensionality reduction and generation, building on DDPMs in latent space and have become widely used across domains [128–130]. First, the data is mapped from the high-dimensional data to a lower-dimensional latent space, preserving the shared variability in the data, commonly making use of regularized autoencoders, as introduced above [131]. In the second stage, DDPMs are trained on the lower-dimensional latent distribution. The main objective of the diffusion model is thus the faithful generation of inferred autoencoder latents, possibly conditioned on external variables.

3

Publications

In this chapter, I summarize the three publications that form the basis of this thesis, as well as my contributions to each project. In the respective result sections, the emphasis lies on the experiments and results in which I was directly involved. The papers in their original format are in the supplement.

1. Stefano Zucca*, Auguste Schulz*, Pedro J Gonçalves, Jakob H Macke, Aman B Saleem, Samuel G Solomon (2025) Visual loom caused by self-movement or object-movement elicits distinct responses in mouse superior colliculus. *Current Biology.*

2. Auguste Schulz, Julius Vetter, Richard Gao, Daniel Morales, Victor Lobato-Rios, Pavan Ramdya, Pedro J Gonçalves, Jakob H Macke (2025) Modeling conditional distributions of neural and behavioral data with masked variational autoencoders. *Cell Reports.*

3. Jaivardhan Kapoor*, Auguste Schulz*, Julius Vetter, Felix Pei, Richard Gao, Jakob H Macke (2024) Latent Diffusion for Neural Spiking Data. *The Thirty-eighth Annual Conference on Neural Information Processing Systems.*

* denotes first author equal contribution.

3.1 Visual loom caused by self-movement or object-movement elicits distinct responses in mouse superior colliculus

Motivation How animals react to certain visual stimuli in their environment can be a matter of life or death. What makes correctly assessing the situation particularly challenging is that the same visual sequence can occur in vastly different contexts.

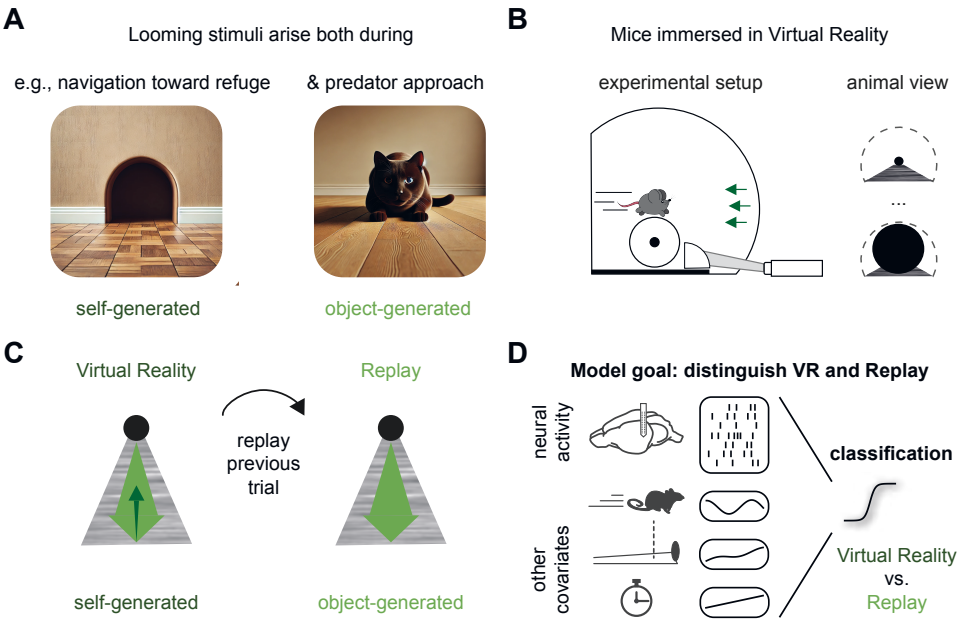


Figure 3.1: Does Superior Colliculus distinguish between self-generated and object-generated visual loom? A. Schematic of two different contexts generating similar visual stimuli looming directly at them from the perspective of a mouse. (Generated with Dall·e.) **B.** Animals were head-restrained above a treadmill in an immersive virtual reality (VR) environment in which they could move along a corridor. **C.** The coupled VR condition, where the visual scene follows the movement on the treadmill, allows us to test visual loom caused by self-movement. Replaying frames of a previous trial (Replay) mimics visual loom caused by object-movement, yet with identical stimulus statistics. **D.** The modeling goal of this study is to classify recordings into their respective contexts of VR (self-generated) vs. Replay (object-generated). Panels and legends adapted from Zucca, Schulz et al. (2025).

For mice, similar looming stimuli[1] can occur both when they navigate toward a dark refuge or when predators approach (**Figure 3.1**A). The main difference is that in one context, the dark, looming stimulus is self-generated, whereas in the other context, it is caused by the predators' movement (object-generated).

The animal's own behavior seems to be the distinguishing factor. But where along the visual pathway is behavioral information integrated with the visual information? The superior colliculus (SC) is an ancient subcortical structure that is known to play a role in visual and movement integration and is linked to innate responses such as approach and avoidance [132, 133], which are often triggered by, or the cause of, looming stimuli in the wild. In this project, we thus asked if superior colliculus differentially represents visual loom caused by self-movement or object-movement [74].

[1] Note that in this project, we deviate from the notion of an object looming from above and instead investigate visual objects looming from the front.

Methods To test this question in a controlled environment, our collaborators placed mice in a Virtual Reality (VR) environment, which allowed the mice to move along a corridor at their own pace (**Figure 3.1**B, left). A 'trial' consisted of reaching the end of said corridor, where a dark object was placed, which got larger upon approach in VR (**Figure 3.1**B, right). The free movement and lack of temporal trial alignment render this a challenging unconstrained dataset with vastly different trial lengths.

To study the differential effect of looming stimuli across contexts, our collaborators recorded the animal's behavior and neural activity from superior colliculus in two different contexts: 1) when the object movement was coupled to the animal's movement through the VR setup (self-generated); and 2) when the movement of the object was uncoupled (object-generated). To ensure the consistent stimulus statistics of the looming object, the video sequences of their previous self-generated VR trials were replayed to each animal (**Figure 3.1**C).

We set up a logistic regression-based classifier to assess if neural population activity in superficial and intermediate layers of SC reflects the two contexts despite identical visual inputs. To account for potential confounding factors, such as running speed or the exact recording times of these different contexts, we used a multivariate logistic regression approach. This allowed us to ask if neural activity improved prediction performance beyond what was already accounted for by other covariates (**Figure 3.1**D).

Results Our collaborators demonstrated that running speed has a strong influence on neural activity in the superior colliculus, particularly in the intermediate layers (SCim), compared to the more visually modulated superficial layers (SCs; paper Figs. 1 and 2). Thus, we first characterized the running behavior across the two contexts: Some animals stopped running completely as soon as the visual input was decoupled from their own movement during Replay (**Figure 3.2**A, top), while others maintained similar movement profiles across contexts (bottom). Overall, running profiles differed between VR and Replay conditions in many sessions (**Figure 3.2**B): During Replay, animals were less active (left) and had lower running speeds when active (right).

Including sessions with highly disparate behavioral profiles across contexts would inflate classification accuracy by merely picking up on the behavioral differences. We thus excluded 21 of the 48 sessions, notably those where animals seemed to have noticed the contextual difference—rendering it a conservative exclusion criterion for a context classification task. Besides behavior, we further accounted for the time-into-experiment, as temporal drift in neurons can further inflate the classification accuracy in a blocked structure (first VR, then Replay, VR, then Replay) present in this experimental setup (**Figure 3.2**A, see paper methods). In some sessions, covariates alone are predictive of context despite corrections (**Figure 3.2**C). Neural population activity in both SCs and SCim, however, significantly increases classification accuracy (**Figure 3.2**C; see paper for significance tests), with stronger effects in intermediate layers (see paper for significance tests).

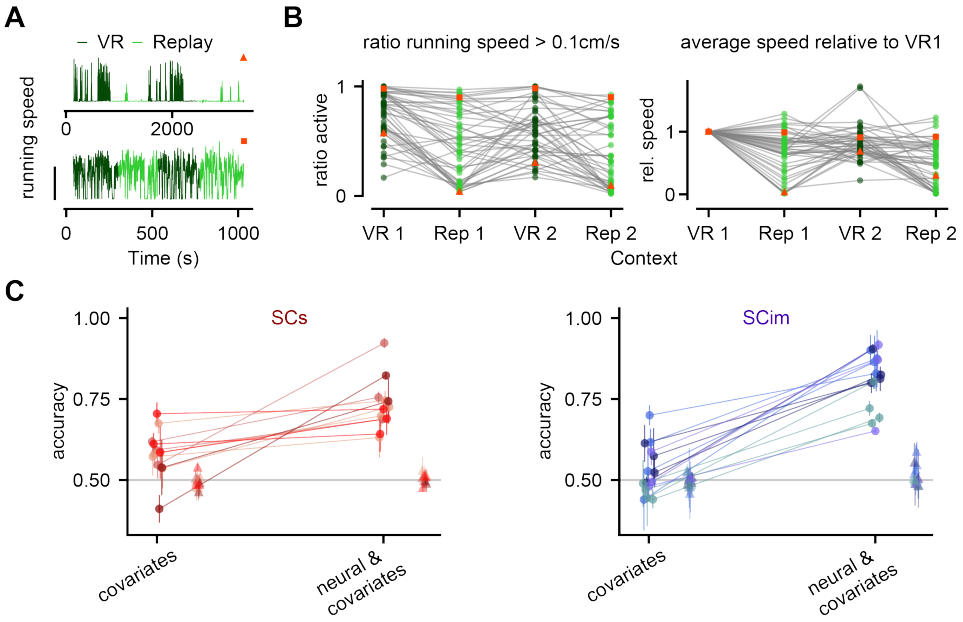


Figure 3.2: Characterization of context-dependent running speed modulation and differential representation in neural populations of SCs and SCim A. Temporal dynamics of locomotion behavior in two example sessions. (top) Session in which the animal ceased locomotion upon exposure to Replay trials (light green), and resumed locomotion when reintroduced into VR (dark green). (bottom) Session in which locomotion speed was similar during VR and Replay. Scale bar, 20 cm/s. **B.** (left) Fraction of time in each condition where animal locomotion speed was at least 0.1 cm/s. Data represent 48 sessions in 14 animals. Red squares and triangles indicate example sessions in A. (right) Average locomotion speed in each condition relative to that in the first exposure to VR. **C.** Test classification accuracy of 12 SCs sessions (red, left) and 15 SCim sessions (blue, right) when including various behavioral and experimental covariates, such as speed, time into the experiment, or the position on the track, or when including neural activity as well as those covariates. Circles indicate the mean performance in each session, and triangles indicate the mean of shuffled controls for each session. Symbols of the same hue are sessions from the same animal. Panels and legends adapted from Zucca, Schulz et al. (2025).

In summary, we have demonstrated that behavioral profiles vary drastically between the two different contexts, requiring careful curation and selection of the data to avoid inflating neural decoding accuracy. Despite all efforts to account for confounds in the data selection process, we have demonstrated that some predictive power remains when classifying context from non-neural covariates. The main challenge of this work, therefore, lay in carefully crafting inclusion criteria and analyses that allowed us to assess if neural activity reflects differences in context not driven by behavioral changes. As such, this study highlights common challenges when moving toward unconstrained behaviors and non-standard experimental paradigms. In conclusion, we found that SC activity can distinguish between visual motion arising from an animal's own movement and motion caused by an external agent.

Contributions Contributions in the publication are listed according to an adapted CRediT system, reflecting the separation between the experimental work and subsequent computational analyses. The work is co-authored by Stefano Zucca, Auguste Schulz (myself), Pedro J. Gonçalves, Jakob H. Macke, Aman B. Saleem, and Samuel G. Solomon. Stefano Zucca and I are **shared first authors**, with a clear division between experimental and computational contributions.

The experiments were conceptualized and conducted by Stefano Zucca, Aman Saleem, and Samuel Solomon before Pedro Gonçalves, Jakob Macke, and I joined the project. For the computational and analytical aspects of the work, I performed formal analyses and generated visualizations for both neural and behavioral data analyses. In particular, I conceptualized and conducted the logistic regression analyses, and prepared all related figures and quantitative analyses, under the supervision of Pedro Gonçalves and Jakob Macke, with feedback from all co-authors. I co-wrote the original draft of the manuscript, focusing on the computational aspects, and contributed to the review and editing process alongside all co-authors. My contributions are featured in two of the four main paper figures and serve as the basis for the main result displayed in the visual abstract. Here, I have only included figures for which I was responsible for data analysis, model setup, and visualization.

3.2 Modeling Conditional Distributions of Neural and Behavioral Data with Masked Variational Autoencoders

Motivation We began this work by asking: can we simulate all possible behaviors of an animal that are likely, given a specific neural activity trace? And conversely, what types of neural patterns are likely while an animal is carrying out a specific, potentially complex behavior?

In statistical terms, these questions target the conditional distributions of neural activity given behavior and vice versa, commonly referred to as neural encoding and decoding (**Figure 3.3**A, left). The same holds true for other prediction problems arising in neuroscience, such as predicting unobserved behavioral time series from observed behaviors or neural activity of one brain region from another region (**Figure 3.3**A, middle). Critically, such predictions are ideally accompanied by uncertainty estimates, informing users if the predictions are trustworthy (**Figure 3.3**A, right). As such, almost all modern analyses in systems neuroscience can be described as calculating conditional distributions. Given the size and complexity of modern datasets, another common and often required analysis step is dimensionality reduction and inferring latent representations of neural activity or behavior, e.g. [39, 41, 43, 47, 48, 56, 90].

While progress on these separate fronts has been remarkable, methods that can simultaneously achieve encoding, decoding, *and* dimensionality reduction—unified in one model—could open up new analyses and further our understanding of the

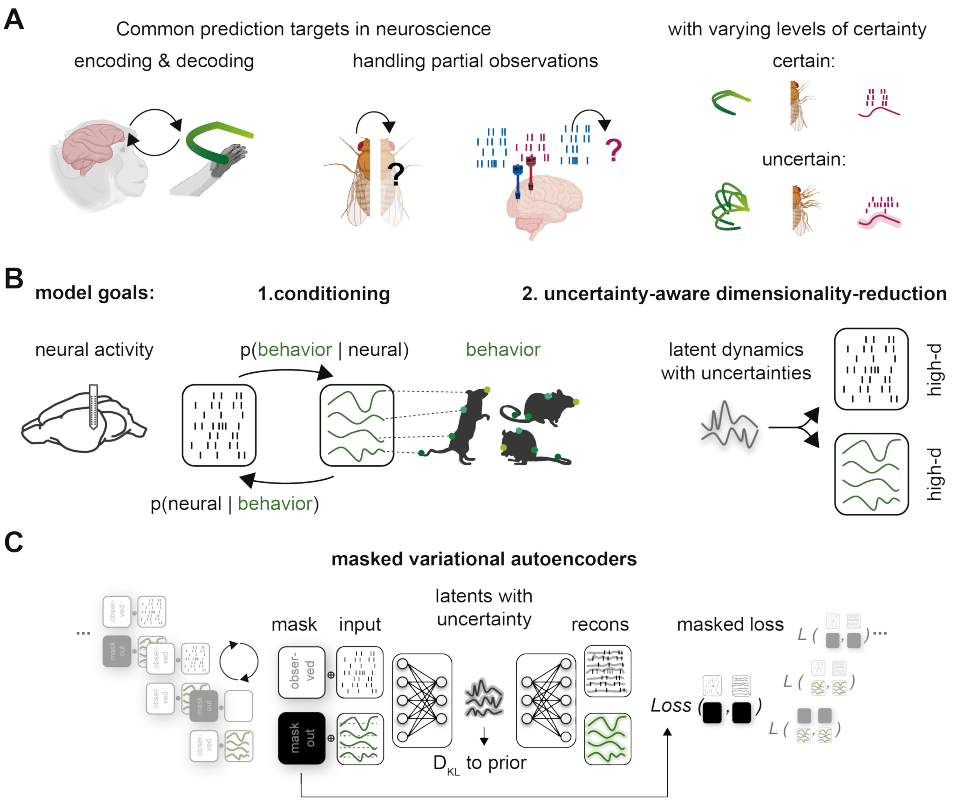


Figure 3.3: How can we model conditional distributions of high-dimensional neural and behavioral data? A. Schematic of common prediction targets in neuroscience: encoding neural activity from behavior or decoding behavior from neural activity, unobserved behavior from observed behavior, or activity in one neural population given another one. Depending on how constrained the prediction target is by the conditioning variable, the predictions have varying levels of associated uncertainty. (Created with BioRender.com) **B.** We aim to address two modeling goals simultaneously: conditioning, e.g., neural activity given behavior or vice versa, and joint dimensionality reduction that accounts for uncertainties. **C.** We demonstrate how to unify these two model desiderata with a multi-modality masked variational autoencoder approach. Panels and legends adapted from Schulz et al. (2025).

neural mechanisms of complex behavior. To this end, the goal of this work was to develop a method that fulfills two model desiderata simultaneously: joint, uncertainty-aware dimensionality reduction of two or more modalities and modeling respective conditional distributions of one modality given the other (**Figure 3.4**B).

Method To address these model desiderata in a single model, we focused on variational autoencoders (VAEs; [69, 70]). VAEs are generative models that infer a low-dimensional latent distribution and are capable of jointly modeling different data modalities [116], making them well-suited for neural and behavioral data. However,

vanilla VAEs do not fulfill the second criterion of conditioning on data subsets, in particular, not on complex time series such as behavioral or neural population sequences. To address this issue, we propose a more general approach to estimating conditional distributions in VAEs for neural-behavioral data: modeling the distribution of an unobserved data subset given an observed data subset $p(\text{ unobserved } | \text{ observed })$. This reformulation allows us to draw inspiration from the missingness literature, targeting such distributions [116, 134].

To capture such conditionals, we modify the training scheme and loss of classical VAEs (**Figure 3.3**C). Similar to Nazábal et al. [116] and Collier, Nazábal, and Williams [134], we mask specified subsets for the desired conditionals during training and calculate the reconstruction loss L solely on observed values. Concretely, we either impute masked continuous values, such as the behavior with the mean value, or discrete neural activity with zeros, but importantly, we provide the information of what has been masked to the VAE through an adjusted masked loss. During training, we alternate between the specified masking patterns, allowing us to query different conditions at test time.

Results First, we validated our masked VAE approach on a synthetic dataset sampled from a Gaussian Latent Variable Model (GLVM), for which we can analytically calculate ground-truth conditional and posterior distributions. We contrast our masked VAEs with naive (vanilla) VAEs that are trained on all observed data. Naive VAEs fail to capture the true (black) conditionals—unobserved x_i^{unobs} given the observed x^{obs}, while masked VAEs capture them perfectly (**Figure 3.4**A, top). The discrepancy arises from differences when inferring the latent (posterior) distribution: masked VAEs, but not naive, can infer the correct posterior when half of the x dimensions are unobserved (**Figure 3.4**A, bottom, posterior for one test sample, B: posterior variance across model runs). Both the posterior variance that fails to increase when some inputs are unobserved in naive VAEs and simulation based calibration (SBC; [135, 136]) checks reveal that naive VAEs are not just wrong but confidently so (**Figure 3.4**B,C). The conditional predictions of the masked VAE are accurate and well-calibrated, demonstrating that structured masking is sufficient to target correct distributions in such a well-specified Gaussian example.

Next, we demonstrated on a high-dimensional behavioral dataset of freely walking flies collected by our collaborators [137] that we can sample unobserved key point trajectories given the remaining behavioral key points and thus that masked training can be applied to time series (not shown here, see the paper).

For modeling cortical data of macaques performing a continuous reach task [138], we specify masks for encoding and decoding, masking either neural activity or behavior (**Figure 3.4**D). Masked mean reconstructions of behavioral traces conditioned on neural activity are more accurate than naive predictions across many model seeds (**Figure 3.4**E, left). Again, SBC checks demonstrate that the masked but not naive approach is well calibrated (**Figure 3.4**E, right). After applying different masks during training, we can query the exact same VAE for the encoding distribution,

generating samples of possible firing rates during queried behaviors (**Figure 3.4**F). As such, this approach unifies encoding and decoding in a single model.

Uncertainty about the inferred latent state should increase with increasing levels of unobserved data. We assessed whether latent variables capture such an increase in latent uncertainty when only partial neural recordings with varying levels of masked data were passed to the VAE. We found that masked VAEs, but not naive VAEs, indeed increase the latent uncertainty of the most informative latents (**Figure 3.4**G).

In conclusion, we introduce a VAE-based method for providing uncertainty-based estimates of conditionals, bridging generative modeling and classical encoding and decoding analyses in neuroscience. Additionally, our work highlights a commonly overlooked yet crucial aspect of analyzing neural and behavioral data: the role of uncertainty estimates.

Contributions Contributions are listed in the publication according to the CRediT system. This publication was co-authored by Auguste Schulz (myself), Julius Vetter, Richard Gao, Daniel Morales, Victor Lobato-Rios, Pavan Ramdya, Pedro Gonçalves, and Jakob Macke. I am the **first author** of this work.

One of the datasets on which we demonstrated the applicability of our masked VAE approach had not been published previously. It was collected at EPFL in the group of Pavan Ramdya and kindly provided and published alongside our manuscript. To clarify my contributions, I separate the contributions related to data collection at EPFL from the development of the machine learning method and only outline the latter here: I conceptualized the project with the help of Pedro Gonçalves and Jakob Macke, who provided supervision throughout the project. I developed the method, wrote all the code (software), conducted all computational experiments, validated them, and performed the formal analysis. I led the investigation with the help of Pedro Gonçalves and Jakob Macke. I was solely responsible for managing the resources and curating the data to fit the machine learning pipeline developed for this project. I wrote the original draft of the manuscript and reviewed and edited it with the help of all co-authors. I made all the figures and visualizations for the paper. Together with Jakob Macke, I was responsible for the entire project administration. Jakob Macke provided funding resources.

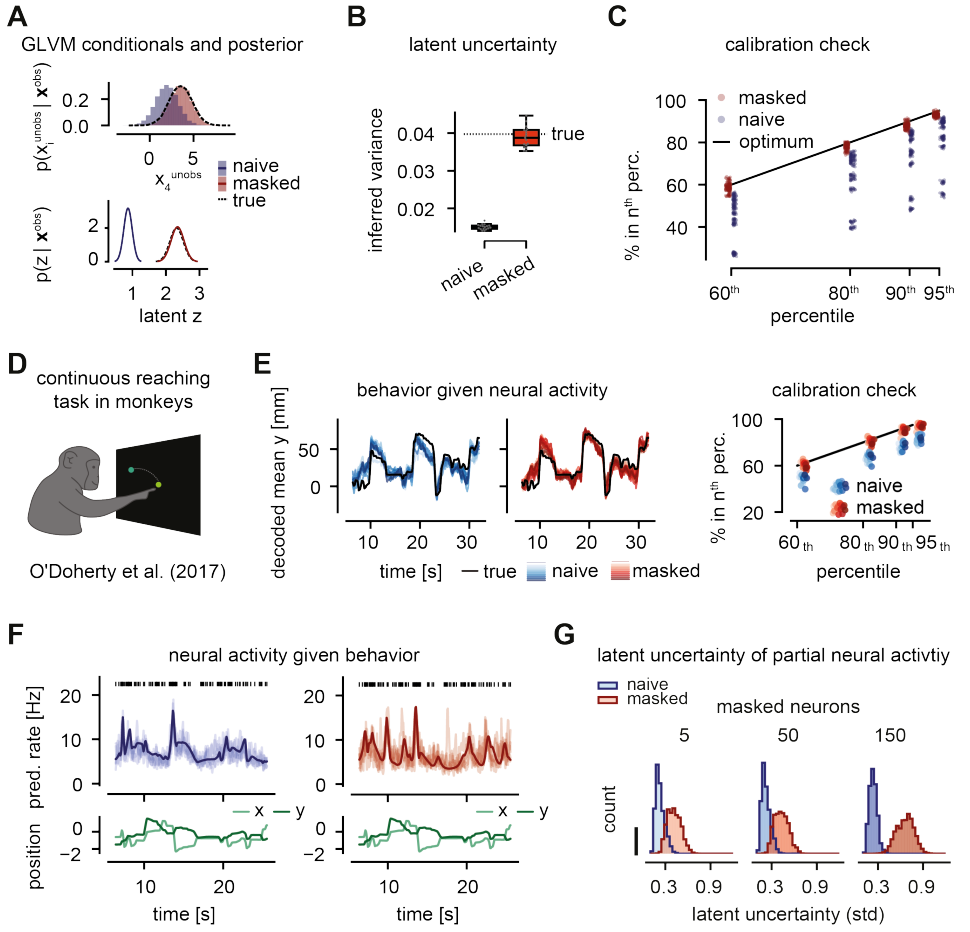


Figure 3.4: Masked VAEs capture the true conditional and posterior distributions in a GLVM and outperform naive VAEs on real data. A. (top) Samples of the conditional distribution of an unobserved x dimension given all observed dimensions inferred by masked (red) and naive (blue) VAEs, compared to the true distribution (black). Here, 50% of the values are masked. (bottom) Corresponding inferred and analytical (true) posterior distributions over latent z given only the observed dimensions of one test sample. **B.** Inferred posterior variance, represented as the mean over test samples, across multiple instantiations (seeds) compared to the true posterior variance (dotted line). **C.** Calibration checks of predicted conditional distributions for all masked x-dimensions across multiple model seeds for masked and naive VAEs. Optimal calibration in grey. **D.** Cortical electrophysiology recordings are collected while a monkey performs self-paced, continuous reaches on an 8x8 grid. **E.** Decoding distribution: (left) True behavioral vertical reach trajectories and mean predictions for the naive (blue) and masked (red) VAEs for multiple model seeds (hue). (right) Calibration checks for these reach trajectories and corresponding VAE samples across multiple model seeds. Optimal calibration in grey. **F.** Encoding distribution: Sampled rate predictions and mean rate prediction from both models (naive blue, masked red) given the standardized behavioral trajectory (bottom, green). **G.** Latent uncertainty: Distribution of latent uncertainty per time-step at varying masking levels (masking of 5, 50, or 150 neurons). The scale bar represents 300 counts. Panels and legends adapted from Schulz et al. (2025).

3.3 Latent Diffusion for Neural Spiking Data

Motivation Denoising Diffusion Probabilistic Models (DDPMs) have become state-of-the-art for generating realistic samples in various domains, ranging from image generation to sound and video synthesis [73]. Their ability to flexibly condition generation on external variables makes them particularly interesting in neuroscience and our quest to link neural activity and behavior. Thus, we asked if we could also leverage DDPMs for modeling neural spiking activity conditioned on behavioral variables, building on the masked VAE approach introduced above.

As mentioned previously, neural activity is high-dimensional, yet often has an underlying low-dimensional structure. This characteristic of neural activity had not been incorporated in DDPMs for neural time series [123]. Ideally, we want a model that is flexible and generic, such that it can generate diverse sets of spiking data from different subjects and species, accounts for underlying low-dimensional structures in high-dimensional neural data, *and* can be flexibly conditioned on behavior or task variables (**Figure 3.5**A).

Method Thus, we proposed a new method, Latent Diffusion for Neural Spiking data, or LDNS for short. LDNS is a generative model for population spiking data that fulfills all three model desiderata (**Figure 3.5**B). We first infer smooth low-dimensional latents and then train a diffusion model, that can be flexibly conditioned on behavioral covariates, directly on these latents. As such, LDNS combines the strength of neural dimensionality reduction [39] with that of diffusion-based generation [73, 122, 123].

Concretely, we first train a regularized autoencoder, as introduced above, to compress the spiking data into a low-dimensional, time-point-matched continuous latent space. We apply temporal smoothness priors for neighboring time points, which lead to smoothly varying extracted latent time series, and introduce bidirectional structured state space layers (S4) to efficiently account for temporal dependencies in the data [106]. On these latents, we then train diffusion models which can either generate such latents unconditionally or can be trained to allow for conditional generation (**Figure 3.5**). Importantly, we also use S4-based denoiser models, which can be unrolled in time, to allow for longer time series generation than the sequence length on which LDNS was trained. We use scalar conditioning and time series conditioning through concatenating the behavioral time series we want to condition on with the latents (see section 2.3.3).

Results We first evaluated the performance of LDNS on a synthetic spiking dataset (Lorenz attractor, **Figure 3.6**A) where we have access to the ground-truth firing rates and latents. We demonstrated that LDNS can not only accurately capture the firing rate distributions but also generate faithful latent samples that reflect the geometry of the Lorenz attractor (**Figure 3.6**B, left). The architectural choice of S4 layers allows for length generalization, as revealed by preserved geometry when

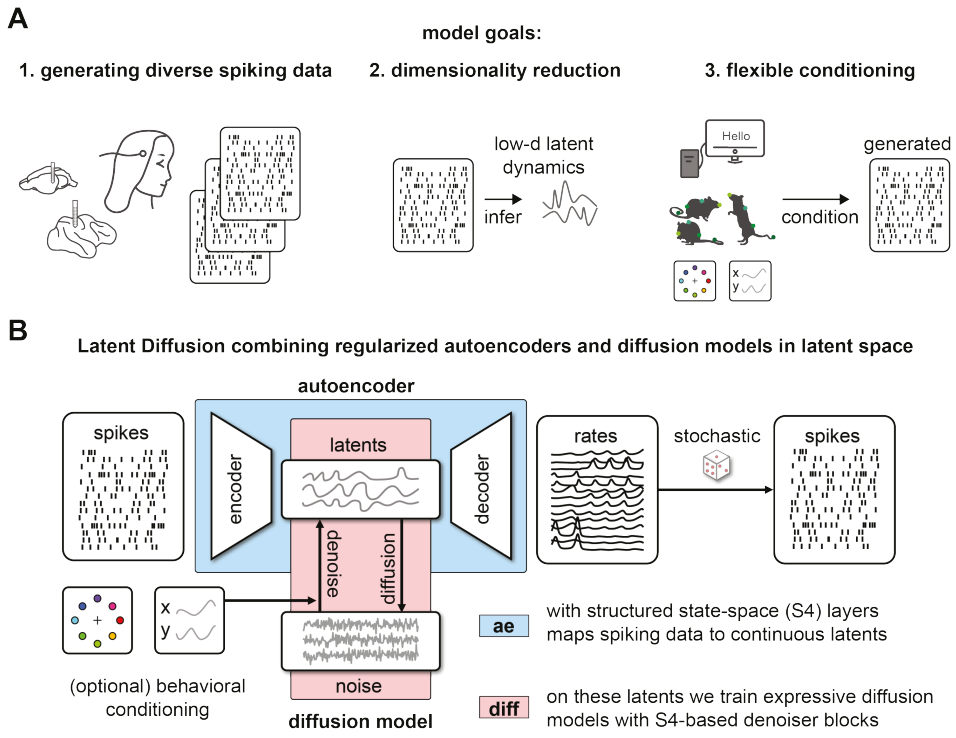


Figure 3.5: How can we generate realistic neural spiking activity conditioned on behavior? A. Schematic of three model desiderata we aim to address in one model: 1. generating diverse spiking data, 2. dimensionality reduction, and 3. flexible conditioning in a behavior or stimulus-dependent manner. **B.** Schematic of LDNS, which combines a regularized autoencoder and flexible diffusion models that act on smooth, low-dimensional latents. Panels and legends adapted from Kapoor, Schulz et al. (2024).

sampling a time series 16 times the length of the training sequences (**Figure 3.6**B, right).

To assess if LDNS is also capable of capturing real electrophysiological recordings, we applied it to recordings from monkey motor cortex recorded during a maze reach task ([92]; **Figure 3.6**C). LDNS samples closely resemble the data (**Figure 3.6**D), which is reflected in matched population-level statistics, e.g., population spike count histograms (**Figure 3.6**E) or pairwise correlations, and single-neuron statistics, e.g., mean or std of inter-spike-intervals (**Figure 3.6**F).

Following other successful variational autoencoder-based approaches for neural spiking data, e.g., [56, 59, 61], the LDNS autoencoder is trained with Poisson log-likelihood. As such, all statistical dependencies are mediated by the latent state and do not allow for neuron-specific, or "private", dynamics. Thus, we learn an additional autoregressive observation model that can account for spike history (sh; [4, 48]).

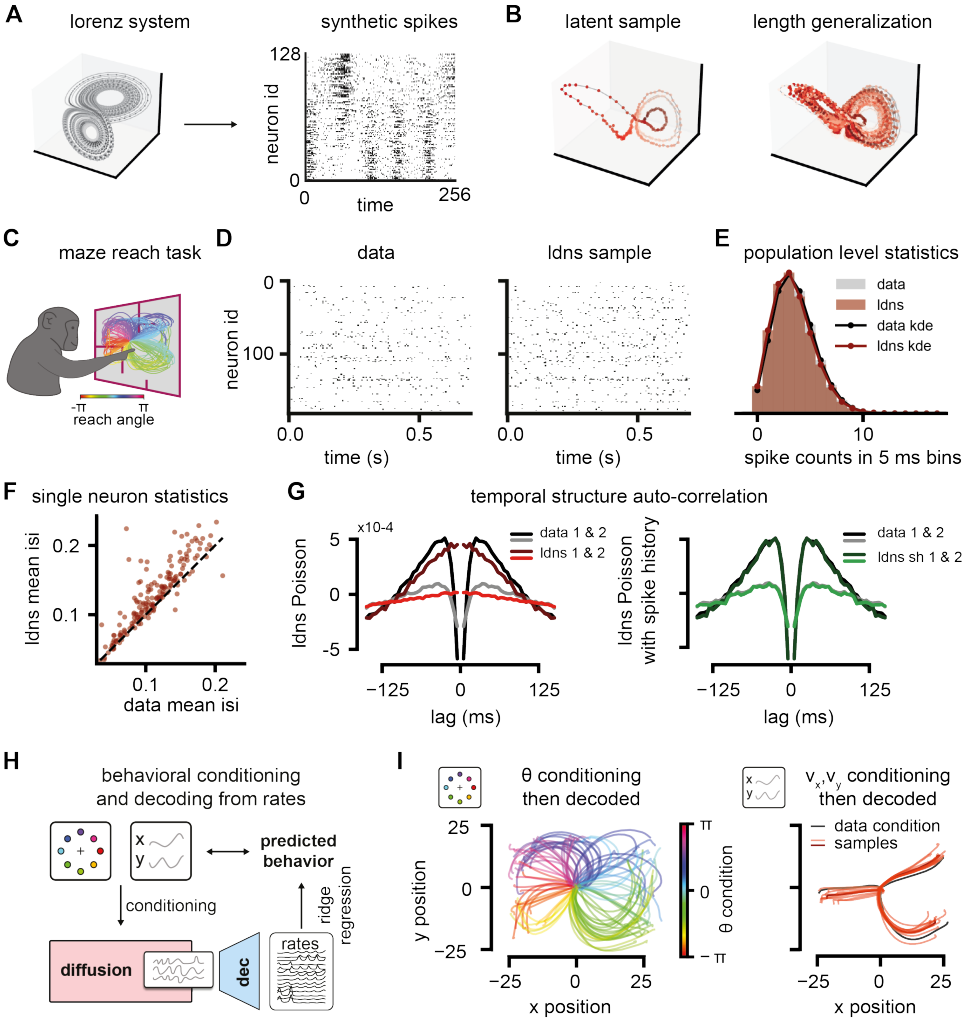


Figure 3.6: Generating synthetic and realistic cortical spiking activity flexibly conditioned on behavior using LDNS A. Synthetic spiking data from an underlying Lorenz system with a Poisson observation model. **B.** Plotted trace of sampled latents from the DDPM (256 bins training length, left) and 16× the original training length (right), demonstrating length generalization. **C.** Schematic of the monkey reach task from the center through a maze leading to diverse reach movements with a fixed trial length. **D.** Depiction of cortical trial data and indistinguishable generated data sampled from LDNS. **E.** Population spike count histogram of data and unconditionally generated LDNS samples. **F.** Mean inter-spike-interval of data and LDNS samples to evaluate single neuron statistics. **G.** Temporal autocorrelation before and after training an expressive observation model that accounts for spike history effects. **H.** Conditional generation paradigm and closed-loop behavioral evaluation scheme. **I.** Decoded behavior from conditionally generated LDNS samples when (left) conditioned on reach angle θ and (right) conditioned on velocity trajectories. Panels and legends adapted from Kapoor, Schulz et al. (2024).

After equipping LDNS with such an observation model, the temporal auto-correlation structure of the data is perfectly captured (**Figure 3.6**G, green vs. red Poisson). Moreover, other single-neuron, as well as population-level statistics, also improve, reflecting an overall increase in realism, which, as we demonstrate in the paper, can even be leveraged for alternative methods, like LFADS [55, 139]. Comparing LDNS to other methods, such as LFADS [56, 139], pi-VAE [61], and TNDM [59], reveals that LDNS(sh) is on par or better in terms of sampling fidelity than previous approaches.

Next, we assessed the ability of conditional LDNS to generate realistic neural activity conditioned on behavioral covariates of varying complexity: the reach angle or entire velocity time series (**Figure 3.6**H). We first train a linear decoder (ridge-regression, following [82]) which maps autoencoder firing rates to behavior. Then, we tested the ability to generate neural time series conditioned on the reach angle θ: From the generated samples of neural activity, we decode, using the same linear decoder, realistic behavior that is consistent with the reach angle θ_{cond} used for conditioning (color bar) and overall reach kinematics (**Figure 3.6**I, left).

Finally, an even more challenging task is the ability to mimic an entire experiment and ask what the neural activity would have looked like if the monkey had performed a specific hypothetical movement. To this end, we train a diffusion model and condition on entire velocity traces (**Figure 3.6**I, right). Velocity-conditioned LDNS is able to produce different samples of neural activity that are consistent with, but not exact copies of, the reach trajectories of the held-out trials (grey), given as the conditioning covariate.

In summary, LDNS is a flexible method that allows for both high-fidelity diffusion-based (conditional) sampling of neural population activity and access to time-aligned low-dimensional representations. Such generated data (conditioned on unseen behavioral conditions) could be useful for augmentation and improving the training of, e.g., brain-computer interface decoding models.

Contributions Contributions are reported according to the university's percentage-based system for co-authored publications. This publication was co-authored by Jaivardhan Kapoor, Auguste Schulz (myself), Julius Vetter, Felix Pei, Richard Gao, and Jakob H. Macke. Jaivardhan Kapoor and I are **shared first authors**, with the ordering decided by a coin toss.

The conception of this project builds on several prior research directions within the lab of Jakob Macke. It emerged from the combination of work on diffusion models for continuous neural time series by Julius Vetter, co-advised by Richard Gao; my own research on latent variable models for linking spiking neural population activity and behavior; and prior work by Jaivardhan Kapoor on latent diffusion models for medical imaging. As such, 30% of scientific ideas can be attributed to me, the same for Jaivardhan Kapoor, 20% to Julius Vetter, and 10% respectively to Richard Gao and Jakob Macke. For this project, no new data was collected and data generation here refers to a corresponding task in machine learning research, namely writing the

software/implementation of the algorithm. Following this definition, I contributed 40% to that and 50% to the analysis with the reverse order for my shared first author. Julius Vetter and Felix Pei each contributed with 5% to the software/implementation. Julius Vetter and Richard Gao each contributed with 5% to the analysis. I had a key role in writing the original draft of the publication and in editing it, with 40% attributed to me and Jaivardhan Kapoor respectively, and 5% each to the remaining 4 authors. These percentages are subjective but were agreed upon by all authors. While they cannot fully capture the complexity of collaborative work, they reflect our shared sense of the relative contributions. In the results section of this thesis, I have included only those figures where I was directly involved in architectural and training choices, tuning of model parameters, and generating the final visualizations.

4

Discussion

The overarching theme of this thesis has been modeling the statistical dependencies between neural activity and behavior using classical discriminative, latent variable, and deep generative machine learning approaches. Despite decades of work on developing powerful latent variable frameworks to link brain activity with behavior, few of these methods have seen widespread adoption in neuroscience. In part, this can be attributed to the vastly different data types and questions in neuroscience. However, it is also due to the growing disparity between method developers (machine learning for neuroscience) and the experimental neuroscientists these tools are meant to serve. To this date, experimental neuroscientists predominantly employ interpretable, linear analysis methods. Yet, the scale and complexity of modern datasets in neuroscience require the adoption of flexible, often non-linear approaches. Given the high variability of both neural and behavioral data, probabilistic treatments are also desirable, which further adds to model complexity. Adoption of such methods is often hindered by the lack of a shared 'modeling language', and by the fact that connections between established statistical tools in neuroscience and modern, deep-learning-based approaches are rarely made explicit.

Addressing this gap is a central contribution of this thesis. I presented three approaches for joint statistical modeling of neural and behavioral data, ranging from classical to deep learning–based methods. Using a conventional discriminative decoding approach, we showed that neural population activity differs between different experimental contexts beyond what is already accounted for by behavioral covariates that change between contexts. Next, we clarified the links between classical encoding and decoding methods with probabilistic dimensionality reduction and deep generative models using a VAE-based approach, enabling sampling from both encoding and decoding distributions. Finally, we explored whether we could also leverage diffusion models for modeling neural activity conditioned on behavior and how to further enhance the realism of generated neural spiking data. To this end, we introduced a latent diffusion approach to do so efficiently and accurately.

In this discussion chapter, I first outline the motivation for jointly modeling neural activity and behavior. I then discuss the types of scientific insight that can be gained from statistical and, particularly, generative modeling approaches. Finally, I address how foundation models may help overcome some of the limitations of current statistical models and how the work presented here could have benefited from or, in the case of the masked VAE work, has inspired foundation model approaches.

For project-specific discussions and limitations, I refer to the respective sections in the individual publications. Limitations shared across all three approaches or those common to statistical models of neural and behavioral data are discussed in the corresponding sections below.

4

4.1 Joint Models of Neural and Behavioral Data

Neural activity both drives and is shaped by behavior. The traditional separation of encoding, predicting neural activity from stimuli or behavior, and decoding analyses, which predict external or internal variables from neural activity, can obscure this inherently bidirectional dependency [3, 36]. Incorporating bidirectionality should not mean abandoning inductive biases about causality and the directionality of information flow. On the contrary, joint modeling approaches can, under appropriate assumptions, enable the investigation of directionality and potential changes thereof over the course of an experiment.

There are several other reasons for the need for carefully designed models that jointly consider neural activity and behavior, including factors that may not initially appear relevant to a given task: Recent findings in rodents show that spontaneous behavioral correlates are widespread and represented across nearly all brain areas [66–68].[1] As a result, even subtle task-induced changes in behavior can confound interpretations of neural activity if behavior is not explicitly accounted for in an analysis [38]. Carefully crafted joint models can partially mitigate spurious correlations among task variables, neural activity, and behavior, reducing the risk of inadvertently capturing "Clever Hans" effects [141].[2] We thus built a multivariate model that included both neural and behavioral features to account for potential confounding factors when predicting context from neural activity. Our analysis indeed revealed that some behavioral features alone were predictive of context, underscoring the importance of comparing neural decoding performance against behavioral baselines to avoid overestimating neural contributions (**Figure 3.1**C, right). It should be noted, however, that we

[1] Using a linear regression model that combined video-based movement features with task variables, during a decision-making task in mice, Musall et al. [66] showed that uninstructed, spontaneous movements were highly predictive of cortex-wide neural activity. In fact, these spontaneous movements accounted for more variance than task variables or instructed movements. However, more recent work in primates suggests that this relationship may not generalize across species [140].

[2] The "Clever Hans" effect refers to a horse that appeared to perform arithmetic but was later shown to respond to subtle, unintentional cues from humans rather than understanding the task itself [141].

only recorded a limited set of behaviors, meaning many remain unaccounted for.

One limitation shared with the other presented approaches is that they provide only correlational evidence. Causal insights require experimental manipulation or the development of mechanistic causal models. With the exception of simpler organisms such as Drosophila melanogaster, which have seen exciting developments toward this goal [142–144], mechanistic full-brain-behavior models remain far out of reach for most animal models.

Another major challenge of modeling neural-behavioral interactions lies in the multitude of relevant timescales [36, 145–147]. None of the presented works explicitly addressed this issue, as all modeled interactions were near instantaneous within a small time window. Deep neural networks, such as those used in the VAE and LDNS projects, could be extended to enable the extraction of relevant dynamical regimes through specialized embedding networks. Future work could focus on effective ways to incorporate events from the distant past that still influence internal states, computational dynamics, and behavior. For behavior, such events could potentially date back years.

Conversely, none of the models presented here capture information on the millisecond timescale, e.g., information carried by the exact timing of spikes. Similar to many latent variable and decoding models [55, 82, 148], they operate on almost rate-like representations or binned spikes, neglecting precise timing information that is critical for plasticity and learning [149].

Depending on the scientific question, these limitations may hinder scientific insight and motivate further methodological development. In general, neuroscientists often struggle to recognize the potential of statistical advances. For generative models of neural and behavioral data in particular, the benefits are not always immediately apparent, which warrants further discussion.

4.2 Generative Latent Variable Approaches

'But what did we actually learn about the brain?' is arguably one of the most common questions method developers in neuroscience face when presenting their work to experimentalists. Method development represents a central step in advancing neuroscience, with progress in data analysis often preceding conceptual discovery. We presented a variety of tasks and analyses that demonstrate the purpose and usefulness of generative approaches for extending encoding and decoding. As such, the developed methods may provide a foundation for future neuroscientific analyses and ultimately biological insight.

Generative models capture full probability distributions, not just point estimates [69–71, 73, 77], enabling the exploration of variability and uncertainty at both the latent and individual feature levels. A key advantage of modeling uncertainty is that it can signal when, for example, a decoded movement or predicted activity pattern should not be trusted, making such approaches particularly powerful in

many practical applications or scientific inquiries. In the VAE project, for example, we demonstrated that masked VAEs applied to neural data during monkey reach behaviors have the desired property of increasing latent uncertainty when predictions are likely incorrect (**Figure 3.4**G, right).

While LDNS can be understood as a powerful generative encoding model that approximates the conditional distribution of cortical activity given reach behavior, masked VAEs enable the capture of both encoding and decoding distributions within a single unified model. Both approaches allow for generating samples from the respective distributions. This is often more intuitive and informative than summary metrics such as R-squared or MSE, especially when modeling degenerate, chaotic systems and multi-modal or time series distributions [150–152]. For example, conditional samples of LDNS illustrated the diversity of generated neural activity and absence of a one-to-one mapping from reach movements to brain activity (**Figure 3.6**I, right). This fundamental neural-behavioral property would likely be obscured in various summary statistics, and certainly in approaches that target only the mean or mode of a distribution.

Generative models can provide additional insight into the structure of neural systems when certain modeling choices, model class, or architectural choices allow for capturing a distribution, while others do not [see e.g., 153]. Meaningful evaluation and such comparisons of generative models require quantitative criteria that define what it means to capture a distribution. Because no standardized benchmarks currently exist for generative neural models, we developed custom metrics for both LDNS and the masked VAE approach to assess population- and neuron-level properties. Future work should aim to establish shared benchmarks [similar to 82, 154, for neural generation], adopting expressive time-series similarity measures [151]. Automated network configuration and learning algorithm selection [155, 156] may further aid in optimizing architectures once such benchmarks are in place and will likely boost the performance of all presented methods.

A remaining limitation is that the generalization ability of the presented methods has not been systematically assessed. Current results are limited to within-dataset performance, and datasets are likely too small to achieve transfer across subjects, brain regions, or species. Achieving such robustness may require new modeling paradigms that can learn representations shared across datasets.

4.3 Toward Foundation Models of Neural and Behavioral Data

Foundation models in systems neuroscience aim to address this challenge by large-scale pretraining across subjects, tasks, and modalities [157–163]. The promise of foundation models lies in their ability to extract neural representations that are useful across a variety of downstream tasks, potentially bringing us closer to understanding how the brain represents external stimuli, internal processes, and behavior. Recent successes with tabular foundation models using in-context learning

indicate that similar strategies could benefit small datasets for neural encoding and decoding studies [164, 165]. For example, the superior colliculus study presented here could have benefited from pretraining to mitigate overfitting and move beyond the single-session logistic-regression classifiers used in this work.

Although the models developed in this thesis were not trained at such scale, both the masked VAE and LDNS projects align conceptually with the goals of foundation models. The masked VAE integrates multiple tasks, i.e., both encoding and decoding within a single probabilistic framework, and incorporates neural and behavioral information through its masking scheme. This unifying treatment of neural and behavioral data has since inspired powerful transformer-based foundation models such as NEDS [162], which extend this principle to large-scale multitask settings.

Similarly, the latent diffusion framework introduced in LDNS has several properties desirable for foundation models. Its S4-based architecture efficiently captures long-range temporal dependencies while maintaining parallelizable computation, making it suitable for scaling to large datasets. The combination of structured state-space layers with DDPMs offers an effective architectural framework for foundation models in systems neuroscience.

Future work toward neural foundation models will require careful design of inductive biases to capture both fine temporal precision and broad contextual structure, while maintaining computational feasibility. The smaller-scale models presented here may inform the development of scalable yet efficient foundation models for neural and behavioral data.

4.4 Conclusion

Overall, this thesis emphasizes the importance of modeling neural activity and behavior jointly, particularly as the field advances in studying interactions during unconstrained, natural behaviors. Probabilistic generative approaches enable the quantification of uncertainty in such variable data and allow for generating realistic samples for hypothesis generation and data augmentation. The work presented here has already inspired follow-up studies and highlights the need for standardized benchmarks to advance generative methods for linking neural activity and behavior. Combined with advances and improved accessibility of pretraining techniques, the work presented in this thesis could help scale classical analyses to the size and complexity of modern datasets and enable experimentalists to gain new insights from their data.

Bibliography

[1] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. San Francisco, USA: W. H. Freeman and Company, 1982.

[2] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT press, 2005.

[3] Liam Paninski, Jonathan Pillow, and Jeremy Lewi. "Statistical models for neural encoding, decoding, and optimal stimulus design". In: *Progress in Brain Research*. Ed. by Paul Cisek, Trevor Drew, and John F. Kalaska. Vol. 165. Computational Neuroscience: Theoretical Insights into Brain Function. Elsevier, Jan. 2007, pp. 493–507.

[4] Jonathan W. Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M. Litke, E. J. Chichilnisky, and Eero P. Simoncelli. "Spatio-temporal correlations and visual signalling in a complete neuronal population". en. In: *Nature* 454.7207 (Aug. 2008), pp. 995–999.

[5] L Paninski and JP Cunningham. "Neural data science: accelerating the experiment-analysis-theory cycle in large-scale neuroscience". In: *Current Opinion in Neurobiology*. Neurotechnologies 50 (June 2018), pp. 232–241.

[6] Nikolaus Kriegeskorte and Pamela K Douglas. "Interpreting encoding and decoding models". In: *Current Opinion in Neurobiology*. Machine Learning, Big Data, and Neuroscience 55 (Apr. 2019), pp. 167–179.

[7] Mackenzie Weygandt Mathis, Adriana Perez Rotondo, Edward F. Chang, Andreas S. Tolias, and Alexander Mathis. "Decoding the brain: From neural representations to mechanistic models". In: *Cell* 187.21 (2024), pp. 5814–5832.

[8] J. A. Nelder and R. W. M. Wedderburn. "Generalized Linear Models". In: *Journal of the Royal Statistical Society. Series A (General)* 135.3 (1972), pp. 370–384.

[9] Wilson Truccolo, Uri T. Eden, Matthew R. Fellows, John P. Donoghue, and Emery N. Brown. "A Point Process Framework for Relating Neural Spiking Activity to Spiking History, Neural Ensemble, and Extrinsic Covariate Effects". In: *Journal of Neurophysiology* 93.2 (2005). PMID: 15356183, pp. 1074–1089.

[10] Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. "Performance-optimized hierarchical models predict neural responses in higher visual cortex". In: *Proceedings of the national academy of sciences* 111.23 (2014), pp. 8619–8624.

[11] Joshua I. Glaser, Ari S. Benjamin, Raeed H. Chowdhury, Matthew G. Perich, Lee E. Miller, and Konrad P. Kording. "Machine Learning for Neural Decoding". In: *eNeuro* 7.4 (July 2020).

[12] Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. "Learnable latent embeddings for joint behavioural and neural analysis". en. In: *Nature* 617.7960 (May 2023), pp. 360–368.

[13] David H Hubel and Torsten N Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *The Journal of physiology* 160.1 (1962), p. 106.

[14] Michael M Merzenich, Paul L Knight, and G Linn Roth. "Representation of cochlea within primary auditory cortex in the cat". In: *Journal of neurophysiology* 38.2 (1975), pp. 231–249.

[15] Charles Bruce, Robert Desimone, and Charles G Gross. "Visual properties of neurons in a polysensory area in superior temporal sulcus of the macaque." In: *Journal of neurophysiology* 46.2 (1981), pp. 369–384.

[16] Doris Y Tsao, Winrich A Freiwald, Roger BH Tootell, and Margaret S Livingstone. "A cortical region consisting entirely of face-selective cells". In: *Science* 311.5761 (2006), pp. 670–674.

[17] Sam Norman-Haignere, Nancy G Kanwisher, and Josh H McDermott. "Distinct cortical pathways for music and speech revealed by hypothesis-free voxel decomposition". In: *neuron* 88.6 (2015), pp. 1281–1296.

[18] Il Memming Park, Miriam L. R. Meister, Alexander C. Huk, and Jonathan W. Pillow. "Encoding and decoding in parietal cortex during sensorimotor decision-making". In: *Nature neuroscience* 17.10 (Oct. 2014), pp. 1395–1403.

[19] Apostolos P Georgopoulos, Andrew B Schwartz, and Ronald E Kettner. "Neuronal population coding of movement direction". In: *Science* 233.4771 (1986), pp. 1416–1419.

[20] Mark M Churchland, Gopal Santhanam, and Krishna V Shenoy. "Preparatory activity in premotor and motor cortex reflects the speed of the upcoming reach". In: *Journal of neurophysiology* 96.6 (2006), pp. 3130–3146.

[21] Kunal K Ghosh, Laurie D Burns, Eric D Cocker, Axel Nimmerjahn, Yaniv Ziv, Abbas El Gamal, and Mark J Schnitzer. "Miniaturized integration of a fluorescence microscope". In: *Nature methods* 8.10 (2011), pp. 871–878.

[22] Tsai-Wen Chen, Trevor J Wardill, Yi Sun, Stefan R Pulver, Sabine L Renninger, Amy Baohan, Eric R Schreiter, Rex A Kerr, Michael B Orger, Vivek Jayaraman, et al. "Ultrasensitive fluorescent proteins for imaging neuronal activity". In: *Nature* 499.7458 (2013), pp. 295–300.

[23] James J. Jun et al. "Fully integrated silicon probes for high-density recording of neural activity". en. In: *Nature* 551.7679 (Nov. 2017), pp. 232–236.

[24] Nicholas A Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, et al. "Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings". In: *Science* 372.6539 (2021), eabf4588.

[25] Misha B. Ahrens, Michael B. Orger, Drew N. Robson, Jennifer M. Li, and Philipp J. Keller. "Whole-brain functional imaging at cellular resolution using light-sheet microscopy". en. In: *Nature Methods* 10.5 (May 2013), pp. 413–420.

[26] Nicholas James Sofroniew, Daniel Flickinger, Jonathan King, and Karel Svoboda. "A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging". In: *eLife* 5 (June 2016), e14472.

[27] Saskia E. J. de Vries et al. "A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex". en. In: *Nature Neuroscience* 23.1 (Jan. 2020), pp. 138–151.

[28] Joshua H. Siegle et al. "Survey of spiking in the mouse visual system reveals functional hierarchy". en. In: *Nature* 592.7852 (Apr. 2021), pp. 86–92.

[29] International Brain Laboratory, Dora Angelaki, Brandon Benson, Julius Benson, Daniel Birman, Niccolò Bonacchi, Kcénia Bougrova, Sebastian A Bruijns, Matteo Carandini, Joana A Catarino, et al. "A brain-wide map of neural activity during complex behaviour". In: *Nature* 645.8079 (2025), p. 177.

[30] Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning". en. In: *Nature Neuroscience* 21.9 (Sept. 2018), pp. 1281–1289.

[31] Semih Günel, Helge Rhodin, Daniel Morales, João Campagnolo, Pavan Ramdya, and Pascal Fua. "DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult Drosophila". In: *eLife* 8 (Oct. 2019), e48571.

[32] Talmo D. Pereira, Diego E. Aldarondo, Lindsay Willmore, Mikhail Kislin, Samuel S.-H. Wang, Mala Murthy, and Joshua W. Shaevitz. "Fast animal pose estimation using deep neural networks". en. In: *Nature Methods* 16.1 (Jan. 2019), pp. 117–125.

[33] John W. Krakauer, Asif A. Ghazanfar, Alex Gomez-Marin, Malcolm A. MacIver, and David Poeppel. "Neuroscience Needs Behavior: Correcting a Reductionist Bias". In: *Neuron* 93.3 (2017), pp. 480–490.

[34] Peiran Gao and Surya Ganguli. "On simplicity and complexity in the brave new world of large-scale neuroscience". In: *Current opinion in neurobiology* 32 (2015), pp. 148–155.

[35] Eric Jonas and Konrad Paul Kording. "Could a Neuroscientist Understand a Microprocessor?" en. In: *PLOS Computational Biology* 13.1 (Jan. 2017), e1005268.

[36] Gordon J Berman. "Measuring behavior across scales". In: *BMC biology* 16.1 (2018), p. 23.

[37] Zhe Sage Chen and Bijan Pesaran. "Improving scalability in systems neuroscience". en. In: *Neuron* 109.11 (June 2021), pp. 1776–1790.

[38] Carsen Stringer and Marius Pachitariu. "Analysis methods for large-scale neuronal recordings". In: *Science* 386.6722 (2024), eadp7429.

[39] John P Cunningham and Byron M Yu. "Dimensionality reduction for large-scale neural recordings". In: *Nature neuroscience* 17.11 (Nov. 2014), pp. 1500–1509.

[40] Gordon J. Berman, Daniel M. Choi, William Bialek, and Joshua W. Shaevitz. "Mapping the stereotyped behaviour of freely moving fruit flies". eng. In: *Journal of the Royal Society, Interface* 11.99 (2014), p. 20140672.

[41] Alexander B. Wiltschko, Matthew J. Johnson, Giuliano Iurilli, Ralph E. Peterson, Jesse M. Katon, Stan L. Pashkovski, Victoria E. Abraira, Ryan P. Adams, and Sandeep Robert Datta. "Mapping Sub-Second Structure in Mouse Behavior". en. In: *Neuron* 88.6 (Dec. 2015), pp. 1121–1135.

[42] Eleanor Batty, Matthew Whiteway, Shreya Saxena, Dan Biderman, Taiga Abe, Simon Musall, Winthrop Gillis, Jeffrey Markowitz, Anne Churchland, John P Cunningham, Sandeep R Datta, Scott Linderman, and Liam Paninski. "BehaveNet: nonlinear embedding and Bayesian neural decoding of behavioral videos". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019, pp. 15706–15717.

[43] Kevin Luxem, Petra Mocellin, Falko Fuhrmann, Johannes Kürsch, Stephanie R. Miller, Jorge J. Palop, Stefan Remy, and Pavol Bauer. "Identifying behavioral structure from deep variational embeddings of animal motion". en. In: *Communications Biology* 5.1 (2022), pp. 1–15.

[44] Karl F.R.S. Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572.

[45] Dmitry Kobak, Wieland Brendel, Christos Constantinidis, Claudia E Feierstein, Adam Kepecs, Zachary F Mainen, Xue-Lian Qi, Ranulfo Romo, Naoshige Uchida, and Christian K Machens. "Demixed principal component analysis of neural population data". In: *eLife* 5 (Apr. 2016), e10989.

[46] Anne C Smith and Emery N Brown. "Estimating a state-space model from point process observations". In: *Neural computation* 15.5 (2003), pp. 965–991.

[47] Byron M. Yu, John P. Cunningham, Gopal Santhanam, Stephen I. Ryu, Krishna V. Shenoy, and Maneesh Sahani. "Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity". In: *Journal of Neurophysiology* 102.1 (July 2009), pp. 614–635.

[48] Jakob H Macke, Lars Buesing, John P Cunningham, Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. "Empirical models of spiking in neural populations". In: *Advances in Neural Information Processing Systems*. Vol. 24. Curran Associates, Inc., 2011, pp. 1350–1358.

[49] Biljana Petreska, Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. "Dynamical segmentation of single trials from population neural data". In: *Advances in Neural Information Processing Systems*. Vol. 24. Curran Associates, Inc., 2011, pp. 756–764.

[50] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. "Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Vol. 54. Proceedings of Machine Learning Research. PMLR, 2017, pp. 914–922.

[51] Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. "Linear dynamical neural population models through nonlinear embeddings". In: *Advances in neural information processing systems* 29 (2016).

[52] Yuan Zhao and Il Memming Park. "Variational Latent Gaussian Process for Recovering Single-Trial Dynamics from Population Spike Trains". In: *Neural Computation* 29.5 (May 2017), pp. 1293–1316.

[53] Anqi Wu, Nicholas A Roy, Stephen Keeley, and Jonathan W Pillow. "Gaussian process based nonlinear latent structure discovery in multivariate spike train data". In: *Advances in neural information processing systems* 30 (2017).

[54] Kristopher Jensen, Ta-Chu Kao, Jasmine Stone, and Guillaume Hennequin. "Scalable Bayesian GPFA with automatic relevance determination and discrete noise models". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 10613–10626.

[55] David Sussillo, Rafal Jozefowicz, L. F. Abbott, and Chethan Pandarinath. *LFADS - Latent Factor Analysis via Dynamical Systems*. 2016.

[56] Chethan Pandarinath, Daniel J. O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D. Stavisky, Jonathan C. Kao, Eric M. Trautmann, Matthew T. Kaufman, Stephen I. Ryu, Leigh R. Hochberg, Jaimie M. Henderson, Krishna V. Shenoy, L. F. Abbott, and David Sussillo. "Inferring single-trial neural population dynamics using sequential auto-encoders". In: *Nature Methods* 15.10 (Oct. 2018), pp. 805–815.

[57] Lea Duncker and Maneesh Sahani. "Temporal alignment and latent Gaussian process factor inference in population spike trains". In: *Advances in Neural Information Processing Systems*. 2018.

[58] Qi She and Anqi Wu. "Neural dynamics discovery via gaussian process recurrent neural networks". In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 454–464.

[59] Cole Hurwitz, Akash Srivastava, Kai Xu, Justin Jude, Matthew Perich, Lee Miller, and Matthias Hennig. "Targeted Neural Dynamical Modeling". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 29379–29392.

[60] Marine Schimel, Ta-Chu Kao, Kristopher T. Jensen, and Guillaume Hennequin. "iLQR-VAE : control-based learning of input-driven dynamics with applications to neural data". en. In: *International Conference on Learning Representations*. Oct. 2021.

[61] Ding Zhou and Xue-Xin Wei. "Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-VAE". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 7234–7247.

[62] Omid G. Sani, Hamidreza Abbaspourazad, Yan T. Wong, Bijan Pesaran, and Maryam M. Shanechi. "Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification". en. In: *Nature Neuroscience* 24.1 (Jan. 2021), pp. 140–149.

[63] Moein Khajehnejad, Forough Habibollahi, Richard Nock, Ehsan Arabzadeh, Peter Dayan, and Amir Dezfouli. "Neural Network Poisson Models for Behavioural and Neural Spike Train Data". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 10974–10996.

[64] Manuel Brenner, Florian Hess, Georgia Koppe, and Daniel Durstewitz. "Integrating Multimodal Data for Joint Generative Modeling of Complex Dynamics". In: *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. Proceedings of Machine Learning Research. PMLR, 2024, pp. 4482–4516.

[65] Rabia Gondur, Usama Bin Sikandar, Evan Schaffer, Mikio Christian Aoi, and Stephen L. Keeley. "Multi-modal Gaussian Process Variational Autoencoders for Neural and Behavioral Data". In: *International Conference on Learning Representations*. 2024.

[66] Simon Musall, Matthew T. Kaufman, Ashley L. Juavinett, Stefan Gluf, and Anne K. Churchland. "Single-trial neural dynamics are dominated by richly varied movements". In: *Nature Neuroscience* 22 (2019), pp. 1677–1686.

[67] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Charu Bai Reddy, Matteo Carandini, and Kenneth D. Harris. "Spontaneous behaviors drive multidimensional, brainwide activity". In: *Science* (2019).

[68] Nicholas A Steinmetz, Peter Zatka-Haas, Matteo Carandini, and Kenneth D Harris. "Distributed coding of choice, action and engagement across the mouse brain". In: *Nature* 576.7786 (2019), pp. 266–273.

[69] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *International Conference on Learning Representations*. 2014.

[70] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 2014, pp. 1278–1286.

[71] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc., 2014.

[72] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In: *International Conference on Machine Learning*. 2015.

[73] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[74] Stefano Zucca, Auguste Schulz, Pedro J Gonçalves, Jakob H Macke, Aman B Saleem, and Samuel G Solomon. "Visual loom caused by self-movement or object-movement elicits distinct responses in mouse superior colliculus". In: *Current Biology* (2025).

[75] Auguste Schulz, Julius Vetter, Richard Gao, Daniel Morales, Victor Lobato-Rios, Pavan Ramdya, Pedro J. Gonçalves, and Jakob H. Macke. "Modeling conditional distributions of neural and behavioral data with masked variational autoencoders". In: *Cell Reports* 44.3 (2025).

[76] Jaivardhan Kapoor, Auguste Schulz, Julius Vetter, Felix C. Pei, Richard Gao, and Jakob H. Macke. "Latent Diffusion for Neural Spiking Data". In: *Advances in Neural Information Processing Systems* (2024).

[77] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[78] E.R. Kandel. *Principles of Neural Science, Fifth Edition*. McGraw-Hill's AccessMedicine. McGraw-Hill Education, 2013. ISBN: 9780071390118.

[79] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[80] Gustav Fritsch and Eduard Hitzig. "Uber die elektrische Erregbarkeit des Grosshirns". In: *Archiv für Anatomie, Physiologie und wissenschaftliche Medicin* 1870 (1870), pp. 300–332.

[81] Arthur E Hoerl and Robert W Kennard. "Ridge regression: applications to nonorthogonal problems". In: *Technometrics* 12.1 (1970), pp. 69–82.

[82] Felix C Pei, Joel Ye, David M. Zoltowski, Anqi Wu, Raeed Hasan Chowdhury, Hansem Sohn, Joseph E O'Doherty, Krishna V. Shenoy, Matthew Kaufman, Mark M Churchland, Mehrdad Jazayeri, Lee E. Miller, Jonathan W. Pillow, Il Memming Park, Eva L Dyer, and Chethan Pandarinath. "Neural Latents Benchmark '21: Evaluating latent variable models of neural population activity". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021.

[83] Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. "On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks". In: *International Conference on Learning Representations*. 2022.

[84] Alison I. Weber and Jonathan W. Pillow. "Capturing the Dynamical Repertoire of Single Neurons with Generalized Linear Models". In: *Neural Computation* (2017).

[85] Konstantin F Willeke, Paul G Fahey, Mohammad Bashiri, Laura Pede, Max F Burg, Christoph Blessing, Santiago A Cadena, Zhiwei Ding, Konstantin-Klemens Lurz, Kayla Ponder, et al. "The Sensorium competition on predicting large-scale mouse primary visual cortex activity". In: *arXiv preprint arXiv:2206.08666* (2022).

[86] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. "UMAP: Uniform Manifold Approximation and Projection". In: *Journal of Open Source Software* 3.29 (2018), p. 861.

[87] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605.

[88] Xiuye Chen, Yu Mu, Yu Hu, Aaron T. Kuan, Maxim Nikitchenko, Owen Randlett, Alex B. Chen, Jeffery P. Gavornik, Haim Sompolinsky, Florian Engert, and Misha B. Ahrens. "Brain-wide Organization of Neuronal Activity and Convergent Sensorimotor Transformations in Larval Zebrafish". In: *Neuron* 100.4 (2018), 876–890.e5.

[89] Carsen Stringer, Lin Zhong, Atika Syeda, Fengtong Du, Maria Kesa, and Marius Pachitariu. "Rastermap: a discovery method for neural population recordings". In: *Nature Neuroscience* 28.1 (2025), pp. 201–212.

[90] Maneesh Sahani. *Latent variable models for neural data analysis*. California Institute of Technology, 1999.

[91] Ofer Mazor and Gilles Laurent. "Transient Dynamics versus Fixed Points in Odor Representations by Locust Antennal Lobe Projection Neurons". In: *Neuron* 48.4 (Nov. 2005), pp. 661–673.

[92] Mark M. Churchland, John P. Cunningham, Matthew T. Kaufman, Justin D. Foster, Paul Nuyujukian, Stephen I. Ryu, and Krishna V. Shenoy. "Neural population dynamics during reaching". eng. In: *Nature* 487.7405 (July 2012), pp. 51–56.

[93] Mohammad Reza Keshtkaran and Chethan Pandarinath. "Enabling hyperparameter optimization in sequential autoencoders for spiking neural data". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[94] I. T. Jolliffe. *Principal Component Analysis*. en. Springer Series in Statistics. New York: New York: Springer, 1986.

[95] Alex H Williams, Tony Hyun Kim, Forea Wang, Saurabh Vyas, Stephen I Ryu, Krishna V Shenoy, Mark Schnitzer, Tamara G Kolda, and Surya Ganguli. "Unsupervised discovery of demixed, low-dimensional neural dynamics across multiple timescales through tensor component analysis". In: *Neuron* 98.6 (2018), pp. 1099–1115.

[96] Juan A. Gallego, Matthew G. Perich, Lee E. Miller, and Sara A. Solla. "Neural Manifolds for the Control of Movement". In: *Neuron* 94.5 (June 2017), pp. 978–984.

[97] Cole Hurwitz, Nina Kudryashova, Arno Onken, and Matthias H Hennig. "Building population models for large-scale neural recordings: Opportunities and pitfalls". In: *Current opinion in neurobiology* 70 (2021), pp. 64–73.

[98] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum Likelihood from Incomplete Data Via the EM Algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (Dec. 1977), pp. 1–22.

[99] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". en. In: *Journal of Basic Engineering* 82.1 (Mar. 1960), pp. 35–45.

[100] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

[101] Hava T Siegelmann and Eduardo D Sontag. "On the computational power of neural nets". In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 440–449.

[102] Ken-ichi Funahashi and Yuichi Nakamura. "Approximation of dynamical systems by continuous time recurrent neural networks". In: *Neural networks* 6.6 (1993), pp. 801–806.

[103] Sepp Hochreiter. "Untersuchungen zu dynamischen neuronalen Netzen". In: *Diploma, Technische Universität München* 91.1 (1991), p. 31.

[104] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (1997), pp. 1735–1780.

[105] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734.

[106] Albert Gu, Karan Goel, and Christopher Ré. "Efficiently modeling long sequences with structured state spaces". In: *International Conference on Learning Representations* (2022).

[107] A. Tustin. "A method of analysing the behaviour of linear systems in terms of time series". In: *Journal of the Institution of Electrical Engineers - Part IIA: Automatic Regulators and Servo Mechanisms* 94 (1 1947), pp. 130–142.

[108] Sebastian Bischoff et al. "A practical guide to sample-based statistical distances for evaluating generative models in science". In: *Transactions on Machine Learning Research* (2024).

[109] Robert E. Kass et al. "Computational Neuroscience: Mathematical and Statistical Perspectives". In: *Annual Review of Statistics and Its Application* 5.Volume 5, 2018 (2018), pp. 183–214.

[110] Diederik P. Kingma and Max Welling. "An Introduction to Variational Autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[111] Jakub Tomczak and Max Welling. "VAE with a VampPrior". In: *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. Ed. by Amos Storkey and Fernando Perez-Cruz. Vol. 84. Proceedings of Machine Learning Research. PMLR, 2018, pp. 1214–1223.

[112] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. "Hyperspherical Variational Auto-Encoders". In: *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)* (2018).

[113] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. "DRAW: A Recurrent Neural Network For Image Generation". In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, 2015, pp. 1462–1471.

[114] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. "A Recurrent Latent Variable Model for Sequential Data". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*. 2016, pp. 2980–2988.

[115] Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. "Dynamical Variational Autoencoders: A Comprehensive Review". In: *Foundations and Trends® in Machine Learning* 15.1-2 (2021), pp. 1–175.

[116] Alfredo Nazábal, Pablo M. Olmos, Zoubin Ghahramani, and Isabel Valera. "Handling incomplete heterogeneous data using VAEs". en. In: *Pattern Recognition* 107 (2020), p. 107501.

[117] Calvin Luo. "Understanding Diffusion Models: A Unified Perspective". In: *arXiv preprint arXiv:2208.11970* (2022).

[118] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. "Structured Denoising Diffusion Models in Discrete State-Spaces". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 17981–17993.

[119] Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q Weinberger. "Latent Diffusion for Language Generation". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc., 2023, pp. 56998–57025.

[120] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. "WaveGrad: Estimating Gradients for Waveform Generation". In: *International Conference on Learning Representations*. 2021.

[121] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. "Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8857–8868.

[122] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. "CSDI: Conditional score-based diffusion models for probabilistic time series imputation". In: *Advances in Neural Information Processing Systems* (2021).

[123] Julius Vetter, Jakob H. Macke, and Richard Gao. "Generating realistic neurophysiological time series with denoising diffusion probabilistic models". English. In: *Patterns* 5.9 (Sept. 2024).

[124] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. "Zero-Shot Text-to-Image Generation". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8821–8831.

[125] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Lit, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Raphael Gontijo-Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. "Photorealistic text-to-image diffusion models with deep language understanding". In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. New Orleans, LA, USA: Curran Associates Inc., 2022. ISBN: 9781713871088.

[126] Prafulla Dhariwal and Alexander Nichol. "Diffusion Models Beat GANs on Image Synthesis". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 8780–8794.

[127] Jonathan Ho and Tim Salimans. "Classifier-Free Diffusion Guidance". In: *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*. 2021.

[128] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022).

[129] Zijiao Chen, Jiaxin Qing, Tiange Xiang, Wan Lin Yue, and Juan Helen Zhou. "Seeing beyond the brain: Conditional diffusion model with sparse masked modeling for vision decoding". In: *Computer Vision and Pattern Recognition* (2023).

[130] Minkai Xu, Alexander Powers, R. Dror, Stefano Ermon, and J. Leskovec. "Geometric Latent Diffusion Models for 3D Molecule Generation". In: *International Conference on Machine Learning* (2023).

[131] Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Scholkopf. "From Variational to Deterministic Autoencoders". In: *International Conference on Learning Representations*. 2020.

[132] Paul Dean, Peter Redgrave, and Gw. Westby. "Event or emergency? Two response systems in the mammalian superior colliculus". In: *Trends in Neurosciences* 12 (1989), pp. 137–147.

[133] Tom Wheatcroft, Aman B. Saleem, and Samuel G. Solomon. "Functional organisation of the mouse superior colliculus". In: *Frontiers in Neural Circuits* 16 (2022), p. 792959.

[134] Mark Collier, Alfredo Nazábal, and Christopher K. I. Williams. "VAEs in the Presence of Missing Data". In: *ICML Workshop on the Art of Learning with Missing Values (Artemiss)*. 2020.

[135] Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. *Validating Bayesian Inference Algorithms with Simulation-Based Calibration*. Apr. 2018.

[136] Samantha R Cook, Andrew Gelman, and Donald B Rubin. "Validation of Software for Bayesian Models Using Posterior Quantiles". en. In: *Journal of Computational and Graphical Statistics* 15.3 (Sept. 2006), pp. 675–692.

[137] Auguste Schulz, Julius Vetter, Richard Gao, Daniel Morales, Victor Lobato-Rios, Pavan Ramdya, Pedro Goncalves, and Jakob Macke. *Walking behavior of flies (Drosophila melanogaster)*. Zenodo, Apr. 2024. URL: https://doi.org/10.5281/zenodo.11002776.

[138] Joseph E. O'Doherty, Mariana M. B. Cardoso, Joseph G. Makin, and Philip N. Sabes. *Nonhuman Primate Reaching with Multichannel Sensorimotor Cortex Electrophysiology*. May 2017.

[139] Mohammad Reza Keshtkaran, Andrew R. Sedler, Raeed H. Chowdhury, Raghav Tandon, Diya Basrai, Sarah L. Nguyen, Hansem Sohn, Mehrdad Jazayeri, Lee E. Miller, and Chethan Pandarinath. "A large-scale neural network training framework for generalized estimation of single-trial population dynamics". en. In: *Nature Methods* 19.12 (Dec. 2022), pp. 1572–1577.

[140] Bharath Chandra Talluri, Incheol Kang, Adam Lazere, Katrina R. Quinn, Nicholas Kaliss, Jacob L. Yates, Daniel A. Butts, and Hendrikje Nienborg. "Activity in primate visual cortex is minimally driven by spontaneous movements". In: *Nature Neuroscience* 26 (Nov. 2023). Publisher: Nature Publishing Group, pp. 1953–1959.

[141] Oskar Pfungst. *Das Pferd des Herrn von Osten (Der Kluge Hans). Ein Beitrag zur experimentellen Tier- und Menschen-Psychologie*. Leipzig: Johann Ambrosius Barth, 1907.

[142] Janne K Lappalainen, Fabian D Tschopp, Sridhama Prakhya, Mason McGill, Aljoscha Nern, Kazunori Shinomiya, Shin-ya Takemura, Eyal Gruntman, Jakob H Macke, and Srinivas C Turaga. "Connectome-constrained networks predict neural activity across the fly visual system". In: *Nature* 634.8036 (2024), pp. 1132–1140.

[143] Sibo Wang-Chen, Victor Alfred Stimpfling, Thomas Ka Chung Lam, Pembe Gizem Özdil, Louise Genoud, Femke Hurtak, and Pavan Ramdya. "NeuroMechFly v2: simulating embodied sensorimotor control in adult Drosophila". In: *Nature Methods* (2024), pp. 1–10.

[144] Roman Vaxenburg, Igor Siwanowicz, Josh Merel, Alice A Robie, Carmen Morrow, Guido Novati, Zinovia Stefanidi, Gert-Jan Both, Gwyneth M Card, Michael B Reiser, et al. "Whole-body physics simulation of fruit fly locomotion". In: *Nature* (2025), pp. 1–3.

[145] Stefan J Kiebel, Jean Daunizeau, and Karl J Friston. "A hierarchy of timescales and the brain". In: *PLoS computational biology* 4.11 (2008), e1000209.

[146] Roxana Zeraati, Anna Levina, Jakob H Macke, and Richard Gao. "Neural timescales from a computational perspective". In: *arXiv preprint arXiv:2409.02684* (2024).

[147] Caleb Weinreb, Jonah E Pearl, Sherry Lin, Mohammed Abdal Monium Osman, Libby Zhang, Sidharth Annapragada, Eli Conlin, Red Hoffmann, Sofia Makowska, Winthrop F Gillis, et al. "Keypoint-MoSeq: parsing behavior by linking point tracking to pose dynamics". In: *Nature Methods* 21.7 (2024), pp. 1329–1339.

[148] Joel Ye and Chethan Pandarinath. "Representation learning for neural population activity with Neural Data Transformers". In: *Neurons, Behavior, Data analysis, and Theory* 5.3 (Aug. 11, 2021), pp. 1–18.

[149] Guo-qiang Bi and Mu-ming Poo. "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type". In: *Journal of neuroscience* 18.24 (1998), pp. 10464–10472.

[150] Gerald M Edelman and Joseph A Gally. "Degeneracy and complexity in biological systems". In: *Proceedings of the national academy of sciences* 98.24 (2001), pp. 13763–13768.

[151] Georgia Koppe, Hazem Toutounji, Peter Kirsch, Stefanie Lis, and Daniel Durstewitz. "Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fMRI". In: *PLOS Computational Biology* 15.8 (Aug. 2019). Ed. by Leyla Isik, e1007263.

[152] Christophe Bernard. "Brain's Best Kept Secret: Degeneracy". en. In: *eNeuro* 10.11 (Nov. 2023).

[153] Poornima Ramesh, Mohamad Atayi, and Jakob H Macke. "Adversarial training of neural encoding models on population spike trains". In: *Real Neurons & Hidden Units: Future directions at the intersection of neuroscience and artificial intelligence @ NeurIPS 2019* (2019).

[154] Brianna M. Karpowicz et al. "Few-shot Algorithms for Consistent Neural Decoding (FALCON) Benchmark". In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang. Vol. 37. Curran Associates, Inc., 2024, pp. 76578–76615.

[155] Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. "Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms". In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '13. Chicago, Illinois, USA: Association for Computing Machinery, 2013, pp. 847–855. ISBN: 9781450321747.

[156] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. "Efficient and Robust Automated Machine Learning". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015.

[157] Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael Mendelson, Blake Richards, Matthew Perich, Guillaume Lajoie, and Eva Dyer. "A unified, scalable framework for neural population decoding". In: *Advances in Neural Information Processing Systems* 36 (2024).

[158] Joel Ye, Jennifer Collinger, Leila Wehbe, and Robert Gaunt. "Neural data transformer 2: multi-context pretraining for neural spiking activity". In: *Advances in Neural Information Processing Systems* 36 (2024).

[159] Yizi Zhang, Yanchen Wang, Donato Jiménez-Benetó, Zixuan Wang, Mehdi Azabou, Blake Richards, Renee Tung, Olivier Winter, Eva Dyer, Liam Paninski, et al. "Towards a" universal translator" for neural dynamics at single-cell, single-spike resolution". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 80495–80521.

[160] Mehdi Azabou, Krystal Xuejing Pan, Vinam Arora, Ian Jarratt Knight, Eva L Dyer, and Blake Aaron Richards. "Multi-session, multi-task neural decoding from distinct cell-types and brain regions". In: *The Thirteenth International Conference on Learning Representations*. 2024.

[161] Eric Y Wang, Paul G Fahey, Zhuokun Ding, Stelios Papadopoulos, Kayla Ponder, Marissa A Weis, Andersen Chang, Taliah Muhammad, Saumil Patel, Zhiwei Ding, et al. "Foundation model of neural activity predicts response to new stimulus types". In: *Nature* 640.8058 (2025), pp. 470–477.

[162] Yizi Zhang, Yanchen Wang, Mehdi Azabou, Alexandre Andre, Zixuan Wang, Hanrui Lyu, International Brain Laboratory, Eva L Dyer, Liam Paninski,

and Cole Lincoln Hurwitz. "Neural Encoding and Decoding at Scale". In: *Forty-second International Conference on Machine Learning*. 2025.

[163] Joel Ye, Fabio Rizzoglio, Adam Smoulder, Hongwei Mao, Xuan Ma, Patrick Marino, Raeed Chowdhury, Dalton Moore, Gary Blumenthal, William Hockeimer, et al. "A Generalist Intracortical Motor Decoder". In: *bioRxiv* (2025).

[164] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. "Tabpfn: A transformer that solves small tabular classification problems in a second". In: *arXiv preprint arXiv:2207.01848* (2022).

[165] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. "Accurate predictions on small data with a tabular foundation model". In: *Nature* 637.8045 (2025), pp. 319–326.

A

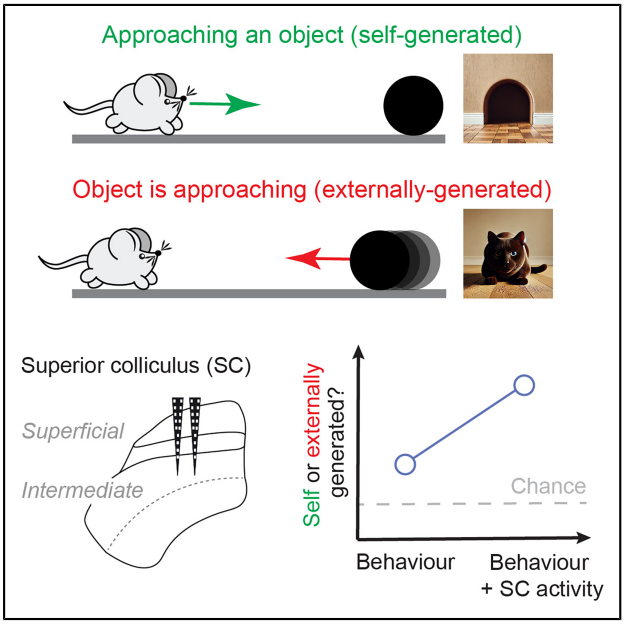
Appendix: Publications

Report

Current Biology

Visual loom caused by self-movement or object-movement elicits distinct responses in mouse superior colliculus

Graphical abstract



Authors

Stefano Zucca, Auguste Schulz, Pedro J. Gonçalves, Jakob H. Macke, Aman B. Saleem, Samuel G. Solomon

Correspondence

s.solomon@ucl.ac.uk

In brief

Zucca, Schulz, et al. show that mice discriminate between self- and object-generated motion in virtual reality. Simultaneous neural recordings from superior colliculus show that while vision dominates superficial layers, intermediate layers integrate vision and locomotion and better discriminate these motion contexts.

Highlights

- Superficial (SCs) and intermediate (SCim) superior colliculus were recorded in VR
- Vision dominated SCs, while SCim was modulated by both vision and locomotion
- Mice changed behavior when vision did not match expectations from self-motion
- Population activity differed when vision matched self-motion, particularly in SCim

Zucca et al., 2025, Current Biology *35*, 4241–4250
September 8, 2025
https://doi.org/10.1016/j.cub.2025.07.013

CellPress

Current Biology

CellPress
OPEN ACCESS

Report

Visual loom caused by self-movement or object-movement elicits distinct responses in mouse superior colliculus

Stefano Zucca,[1,2,7] Auguste Schulz,[3,7] Pedro J. Gonçalves,[3,4,5] Jakob H. Macke,[3,6] Aman B. Saleem,[1,7] and Samuel G. Solomon[1,7,8,*]

[1]Institute of Behavioural Neuroscience and Department of Experimental Psychology, University College London, 26 Bedford Way, London WC1H 0AP, UK
[2]Department of Life Sciences and Systems Biology (DBIOS), University of Turin, via Accademia Albertina 13, 10123 Turin, Italy
[3]Machine Learning in Science, University of Tübingen and Tübingen AI Center, Maria-von-Linden-Str. 6, 72076 Tübingen, Germany
[4]VIB-Neuroelectronics Research Flanders (NERF) and imec, Remisebosweg 1, 3001 Leuven, Belgium
[5]Department of Computer Science and Department of Electrical Engineering, KU Leuven, Oude Markt 13, 3000 Leuven, Belgium
[6]Max Planck Institute for Intelligent Systems, Max-Planck-Ring 4, 72076 Tübingen, Germany
[7]These authors contributed equally
[8]Lead contact
*Correspondence: s.solomon@ucl.ac.uk
https://doi.org/10.1016/j.cub.2025.07.013

SUMMARY

The meaning of a visual image depends on context—a mouse sees an expanding visual stimulus when approaching a dark refuge or when a cat approaches them, and distinguishing between the two is a matter of life and death. The superior colliculus (SC) is an evolutionarily ancient hub essential for survival behaviors like approach and avoidance of threats.[1,2] We therefore combined virtual reality and neural recordings to ask whether matching visual stimuli to self-motion alters behavior and neural activity in SC. We first measured locomotion behavior and neural activity while animals approached an object in virtual reality or while the same object loomed at them. In both contexts, vision dominated activity in the superficial layers of SC (SCs), whereas locomotion had more influence on activity in the intermediate layers (SCim). In addition, animals instinctively slowed their locomotion when nearing the object or when the object neared them. To directly test animals' ability to distinguish self from object motion, we replayed the visual images generated during object approach. Locomotion behavior often changed during replay, showing animals can determine whether visual motion is matched to their self-movement. Further, decoders trained on locomotion behavior or on population activity in SC, particularly in SCim, were able to reliably discriminate self-movement and object movement contexts. We conclude that both mouse behavior and SC activity distinguish the context of visual motion and can thus discriminate motion arising from an animal's own movement and that of an external agent.

RESULTS

To establish how vision and locomotion are integrated during self-movement or object movement, we measured activity in the superior colliculus (SC) of mice immersed in a virtual reality (VR) environment. Mice were head-restrained over a treadmill and could move along a 100-cm-long platform in the VR environment by moving on the treadmill (Figures 1A and 1B; Videos S1 and S2). We used 32-channel multielectrode arrays to measure spiking activity from populations of single units in the superficial layers of SC (SCs) (30 sessions in 7 mice) and intermediate layers (SCim) (35 sessions in 9 mice) (Figure 1C).

Response of neurons in SCs and SCim to visual objects looming from ahead

Before evaluating SC responses to visual stimulation generated by self-movement, we first measured responses to an object that loomed at the animal from directly ahead of them. On each trial, a dark, round object (8 cm in diameter) appeared at the distant end of the platform and then moved toward the animal at a speed of 100 cm/s, expanding from an initial 4.6 degrees of visual angle to the entire display. We analyzed responses where activity was consistent across trials (see STAR Methods), yielding 672 units in SCs and 583 in SCim (Figure 1D). To allow comparison of activity between neurons, we normalized (Z scored) each unit's response by its activity across all stimuli. We found that the edge of the looming object increased activity in most SCs units. Responses in SCim were more diverse, such that activity increased in some SCim units but decreased in others. Consequently, activity was, on average, higher ($p << 0.001$, Student's t test) in SCs (mean Z score 0.85, SD 1.03) than in SCim (0.07, SD 0.56).

Vigorous response to the edge of the looming object in SCs is consistent with previous measurements that show highly responsive, spatially localized visual receptive fields in most SCs neurons (e.g., De Franceschi and Solomon,[3] Dräger and

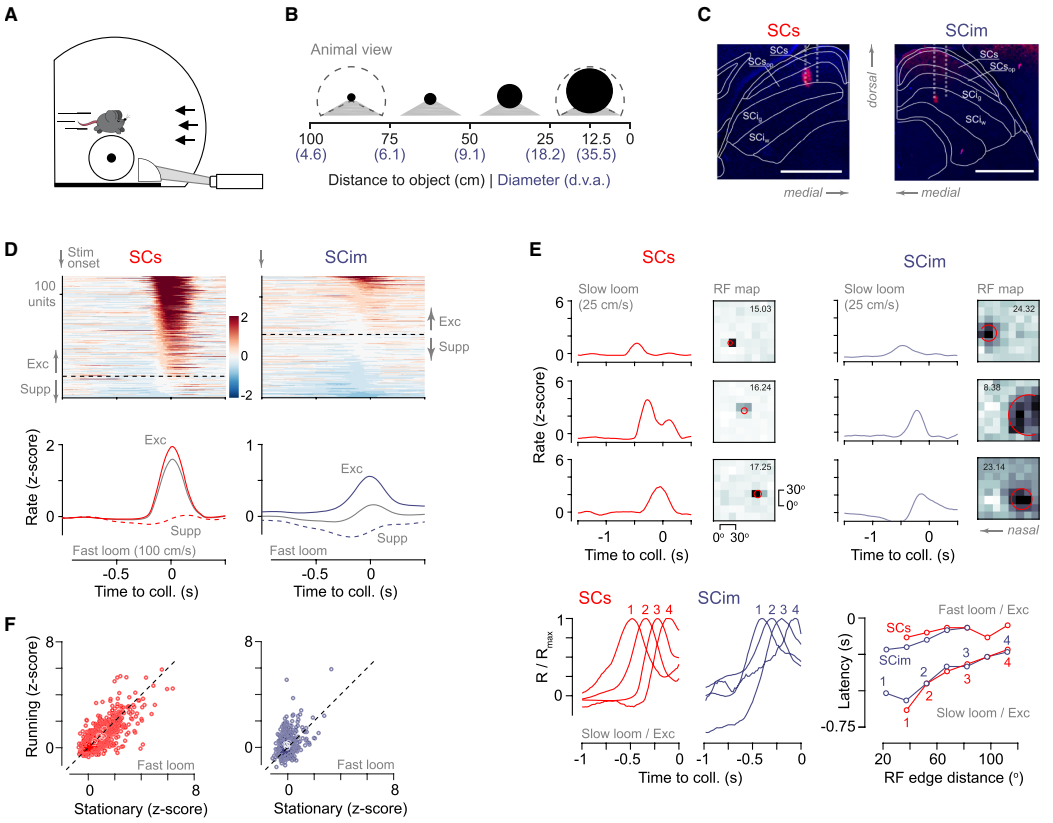


Figure 1. SC responses to looming object combine visual and self-motion signals

(A) Animals were head-restrained above a treadmill in an immersive virtual reality environment.

(B) A round, black object loomed toward the animal at constant speed. Schematics indicate approximate view of the virtual object to the animal; dashed line indicates visual limit of the virtual environment. Numbers in brackets indicate the size of the 8-cm virtual object in degrees of visual angle (d.v.a.) at different distances.

(C) Extracellular recordings were made from SCs and SCim using high-density probes. Images of sections through SC showing red fluorescence deposited by a DiI-coated electrode. Inferred electrode tracks are indicated by vertical lines. SCs image is flipped on the horizontal axis. Scale bars, 1 mm.

(D) Visual response is strongest in SCs. (Upper) Responses (*Z* scores) of consistent units to object loom (100 cm/s), sorted by response in 0.2 s preceding collision. (Lower) Average response over all units in SCs or SCim (gray lines) and for units where looms increased ("Exc") or decreased ("Supp") activity.

(E) Visual receptive fields predict timing of responses to the object. (Upper) Example units showing response to slow object loom (25 cm/s) and receptive field maps for black squares. Receptive field maps are flipped on the horizontal axis so that the nasal visual field is on the left and temporal visual field is on the right. Darker indicates larger positive weight at that location. Red circle indicates estimates of location and size (1 SD of the best-fitting Gaussian) of excitatory receptive fields in each case. (Lower left) Average responses to the slow loom for units where the looming object increased activity. Units were grouped by "edge distance," the visual angle from the center of the object to the nasal edge of the receptive field (at 1 SD from its center); the four lines show average response in each of four of the groups. (Lower right) Latency to 60% of peak response, as a function of edge distance. Units with receptive fields that extend more nasally (edge distance closer to 0) respond earlier to the object. Numerals indicate relevant curves in the lower left plot from which the indices were derived.

(F) Locomotion has a more consistent impact on SCim. Response to fast object looms over the 0.2 s preceding collision, in SCs and SCim, during stationary and running conditions. White points indicate means for individual animals.

See also Figure S1 and Videos S1 and S2.

Hubel,[4] Gale and Murphy,[5] Ito et al.,[6] and Wang et al.[7]). Conversely, fewer neurons in SCim show measurable visual receptive fields,[6,8,9] potentially explaining the reduced population response to the visual object in SCim. To understand the relationship between responses to the object and receptive fields, we determined receptive field maps by presenting flashed black and white squares at different positions in the visual field and fit the maps for each unit with two-dimensional Gaussians. In these recordings, we could extract receptive fields for at least one luminance polarity in most SCs units (607/711, 85%; see

STAR Methods) and fewer SCim units (230/830, 28%) (examples in Figure 1E). Most of these units responded to the black squares, and we found that receptive field size (SD of the Gaussian) for black squares was smaller in SCs (mean 7.3 degrees, SD 3.9, n = 569) than SCim (mean 14.4, SD 6.2, n = 219; $p < 0.001$, Student's t test).

Visual neurons should respond when the object's edge passes through their receptive field, and receptive field location should therefore predict the relative timing of neural responses to the loom stimulus. We tested this prediction in units with a receptive field map, which also increased activity in response to the looming object. We pooled units in which the nasal edge of the receptive field (at 1 SD of the fitted Gaussian) was at a similar location and found the latency at which their average response reached 60% of the maximum. Because objects looming at 100 cm/s expanded very rapidly, latency to response was similar across the visual field (Figure 1E). We therefore also presented objects that approached at slower speed (25 cm/s; Figure 1E; see also Figure S1). At slower speeds, we found that responses occurred earlier among neurons with more nasal receptive fields, in both SCs and SCim. The temporal pattern of responses to an object looming from ahead was therefore largely consistent with that expected from neurons' receptive field locations.

Self-movement modulates response to looming visual objects

Locomotion influenced the response of SC neurons to objects looming toward the animal. Animals moved sporadically during presentation of the looming object, and so we defined each trial as either "running" (locomotion speed at least 2 cm/s) or "stationary" (less than 2 cm/s) and calculated average activity in each condition over the 0.2 s preceding object collision. We found that locomotion was associated with either increases or decreases in the activity in SCs neurons, with no net effect (Figure 1F; mean difference in Z scores of −0.05, SD 0.80; F(lme:state) = 0.78, p = 0.378). In contrast, locomotion consistently increased activity in SCim, by a mean 0.23 (SD 0.72; F(lme:state) = 42.39, $p << 0.001$; SCs vs. SCim: F(lme:state*layer) = 17.54, $p < 0.001$). The differences between SCs and SCim persisted across a range of measurement epochs (Figure S1D) and when speed of locomotion was constrained to ensure overlapping speed ranges in SCs and SCim measurement sessions (Figure S1E). To determine whether locomotion also modulated activity in the absence of a visual stimulus, we analyzed activity during the presentation of a blank gray screen (epochs of which were interleaved among the visual stimuli). We found the same result: locomotion increased activity more in SCim (μ 0.30, SD 0.69) than SCs (μ 0.02, SD 0.40; F(lme:layer*state) = 30.59, $p << 0.001$). Our measurements therefore reveal that neurons in SC process both visual and locomotion signals: visual response is strongest in SCs, but locomotion more consistently increases SCim response.

Vision and self-motion signals also converge during object approach

Visual looming also arises when an animal approaches an object. To characterize response to looms caused by self-movement, we made a simple change to the VR environment. On each trial, the same dark object still appeared at the distant end of the platform, but now it stayed there. As animals moved along the platform, their self-movement brought them closer to the object; the upshot was that the object again loomed in the visual field, but now that looming was caused by self-movement. As time to reach the object varied between trials, we analyzed neuronal activity as a function of distance between animal and object.

We found a consistent pattern of response in SCs as animals approached the object in VR: population neuronal activity increased to a peak and then subsided (Figures 2A and 2B). Object approach produced a different pattern of responses in SCim. Some neurons showed a distinct peak in activity as the animal neared the object. Other neurons showed a distinct reduction in activity at similar positions. In most SCim neurons, however, we saw a ramp in activity (either increase or decrease) as animals moved along the platform. Activity patterns in these neurons usually showed a peak or trough as the animal neared the object. Overall, we saw a consistent pattern of response in SCs and a diversity of responses in SCim.

Locomotion speed influenced activity during object approach, particularly in SCim. SCim activity depended on locomotion speed throughout the trial, even early in the trial when the object was still distant (examples in Figure 2B). SCs activity, by contrast, was similar during both slower and faster approaches. We used ridge regression to establish the relative contribution[10] of visual and self-movement signals to SC activity during object approach. With these regressions, we estimated the amount of variance in neural activity that could be explained by the distance between animal and object (equivalently, the location of object edges in the visual field) or by speed (see STAR Methods; in these experiments visual speed is coupled to locomotion speed). To assess the relative contribution of each factor, we normalized the cross-validated explained variance attained when using either speed or distance, to that attained when using both. Note that these normalized values can be greater than one due to the cross-validation. We found a variety of responses across the population, such that some neurons were more dependent on distance to object, whereas others were more dependent on speed (Figure 2C). Neurons in SCs showed stronger dependence on distance (median normalized explained variance = 0.706) compared with speed (0.256; n = 627). Neurons in SCim instead showed stronger dependence on speed (median = 0.888) compared with distance (0.402; n = 468; SCs vs. SCim: F(lme:model*layer) = 7.82, p = 0.005).

Vision drives instinctive locomotion behaviors

Mice were free to choose how they moved along the virtual platform. Locomotion behavior was, however, dependent on vision and stereotypical: mice consistently slowed down as they approached the object (Figure 3A). This slow-down behavior could be observed on the first day of exposure to an object and then became more pronounced as animals gained experience and ran faster down the platform (Figure 3C). Slow-down behavior was not limited to black objects—in separate experiments, we found that animals showed the slow-down behavior when they instead encountered a white object (Figure S2) and continued to show slow-down behavior when the white object was subsequently replaced with a black one.

How does locomotion behavior differ when visual stimuli match or conflict with that predicted by self-movement? To

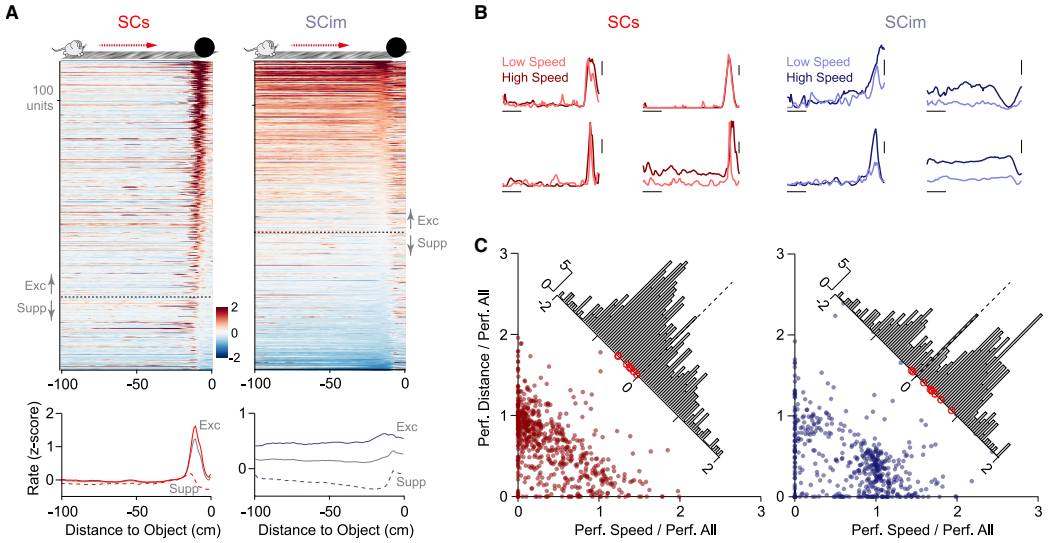


Figure 2. SCs is dominated by vision, and SCim is dominated by self-motion, during object approach

(A) Patterns of activity were stereotypical in SCs, but diverse in SCim, as animals approached a stationary object at the end of the virtual platform. (Upper) Response (*Z* scores) of consistent units, sorted by response in the 0–10 cm (SCs) or 10–20 cm (SCim) preceding collision. Color bar indicates unit response in *Z* scores. (Lower) Average response over all units in SCs or SCim (gray lines) or over units in which object approach increased (Exc) or decreased (Supp) activity.

(B) Speed of approach had more impact on activity in SCim and less impact in SCs. Example units showing average activity during trials of high locomotion speed (darker colors; 5 fastest trials) or low locomotion speed (lighter colors; 5 slowest trials). Horizontal scale bar, 20 cm; vertical scale bar, 1 *Z* score.

(C) Relative contribution of locomotion speed and distance-to-object to activity in individual units in SCs (left) and SCim (right). Each point represents the performance ("Perf."; explained variance) of a ridge regression model when only locomotion speed (abscissa) or distance (ordinate) was allowed to vary, normalized to the performance of the model when both were allowed to vary. These normalized values could be greater than one due to the cross-validation. Histograms show the difference between these performance indices. Red circles on the histogram axis indicate means for individual animals.

See also Videos S1 and S2.

address this, we stored the sequences of visual images generated as an animal approached the object and replayed those sequences later in the session. This allowed us to compare two conditions: locomotion behavior during VR, where the visual stimulus matched that predicted by self-movement, and during replay, where it did not. We found that behavior changed during the replay condition, such that animals often ran slower or even ceased moving during one or both replay epochs (Figures 3E and 3F, Wilcoxon signed-rank test; $p < 0.02$ for all conditions relative to the previous condition, $n = 48$; see also Figure S3). We note that different animals showed different patterns of behavior in VR and replay conditions—some animals ran faster than others and some had more periods of inactivity during replay (Figure S3). When animals ran during the replay condition, their patterns of movement resembled that in the VR condition—animals slowed down when the object was close (Figures 3B and 3D). Similarly, presentation of independently looming objects (as in Figure 1) could evoke slow-down behavior at each of the loom speeds we tested (Figure S1C).

The similar slow-downs observed for VR and replay conditions, and looming objects, suggest that slow-downs are an instinctive sensorimotor response to a looming visual stimulus (at least in head-restrained animals), whether that looming arises from self-movement or object movement. The changes in locomotion behavior between VR and replay conditions, however, demonstrate that mice can discriminate whether the visual experience is a result of their own movement or not.

SC activity is sensitive to match between vision and self-movement

We hypothesized that patterns of neural activity in SC, particularly SCim, depend on whether the visual stimulus is matched to self-movement. Because the temporal pattern of visual stimulation when presenting the looming object as in Figure 1 (where object movement was smooth) could be very different to that during object approach (where object movement depended on animal speed), we tested the hypothesis by analyzing behavior and neural responses during the VR and replay conditions described above. We divided each trial into non-overlapping 0.2 s epochs and performed logistic regression to classify each measurement epoch as arising from VR or replay conditions, using behavioral and neural variables (Figure 4A, see STAR Methods). Some animals stopped running during replay of the visual stimulus and, consistent with the overall impact of locomotion (shown in Figure 2), SCim activity often reduced dramatically when animals simply stopped moving (Figure S4). Thus, to provide an additional test of the hypothesis, we restricted the analyses to sessions in which animals showed a

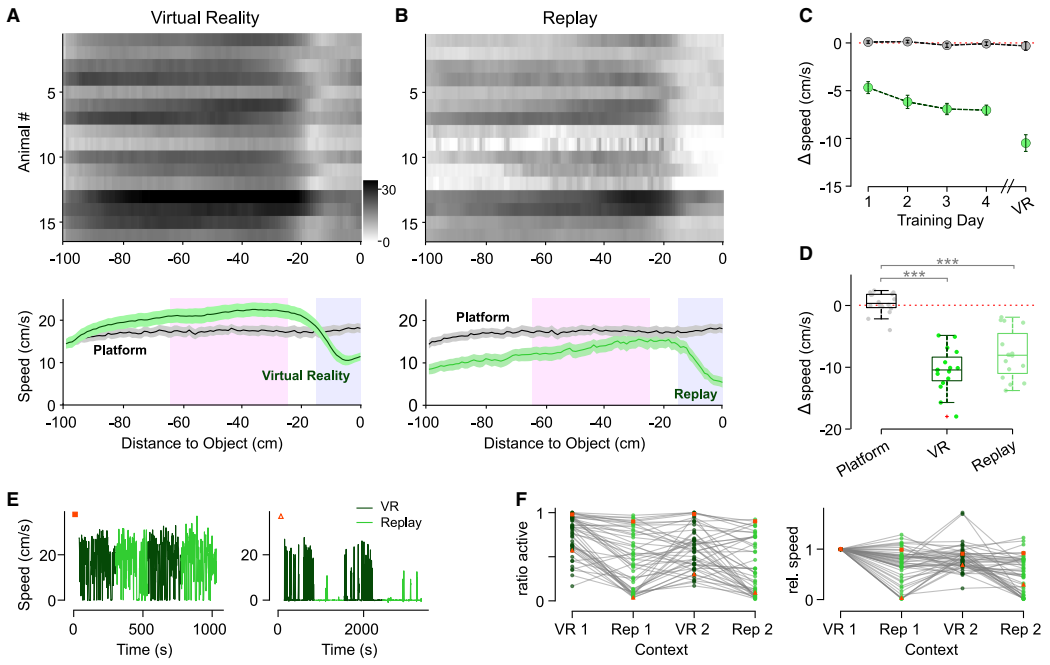


Figure 3. Vision and sensorimotor context drive instinctive locomotion behaviors

(A) (Top) Animals reduce locomotion speed (slow down) as they approach an object in virtual reality (VR). Each row of the image represents the locomotion speed of one animal, averaged across all VR trials in all post-training sessions. Color bar indicates animal speed (cm/s). (Bottom) Slow-down behavior was present during object approach but not when the object was absent (platform trials). Mean and SEM of locomotion speed across animals. Shaded regions indicate locations used for estimates of locomotion speed used in (C) and (D).

(B) Same as (A) but during replay of object approach trials, where the visual stimulus was not predicted by current locomotion behavior. Slow-down behavior remained but overall locomotion speed was reduced during replay.

(C) Slow-down behavior emerged in the earliest stages of exposure to virtual objects. Each point shows locomotion speed (mean and SEM across animals) when animals were near the object (blue-gray regions in A and B), relative to that at larger distances (pink regions in A and B). Equivalent measurements were made during platform trials. The last point shows the data from the post-training sessions shown in (A).

(D) Comparison of slow-down behavior in platform, VR, and replay trials post training. Small symbols show individual animals. VR and replay conditions were significantly different to platform condition ($p << 0.0001$ in both cases) but not significantly different to each other ($p = 0.484$). Boxes show the median and 25th–75th percentiles.

(E) Temporal dynamics of locomotion behavior in two example sessions. (Left) Session in which locomotion speed was similar during VR (dark green) and replay (light green) blocks of trials. (Right) Session in which the animal ceased locomotion upon exposure to replay trials and resumed locomotion when re-introduced into VR.

(F) (Left) Fraction of time in each condition where animal locomotion speed was at least 0.1 cm/s. Data represent 48 sessions in 14 animals. Red squares and triangles indicate example sessions in E. (Right) Average locomotion speed in each condition, relative to that in the first exposure to VR.

See also Figures S2 and S3.

similar range of locomotion in the VR and replay conditions and when the animal was within 50 cm of the object (see STAR Methods).

We first measured classification performance when the regression model was blind to neural activity. Even in sessions where overall locomotion was similar, logistic regression was able to predict whether the animal was in a VR or a replay condition from combinations of behavioral variables, including locomotion speed, distance to the object, and time into the experiment (Wilcoxon signed-rank test; covariates: $p < 0.001$, $n = 12$ SCs sessions; $p < 0.001$, $n = 15$ SCim sessions; Figures 4B and 4C). Classification with locomotion speed alone was sometimes sufficient (speed: SCs sessions, $p = 0.034$; SCim sessions, $p = 0.277$). Time into the experiment also had some predictive power (time: SCs sessions, $p < 0.001$; SCim sessions, $p = 0.035$). Above chance classification persisted in many sessions, even after constraining average time into the experiment and locomotion speed ranges (Figure 4D) between VR and replay conditions (constrained covariates: SCs sessions, $p = 0.009$; SCim sessions, $p = 0.389$; Figures 4B and 4C). We tested the classifier after shuffling the trials between VR and replay conditions and found that performance dropped to chance levels for all variables tested (constrained covariates: SCs sessions, $p = 0.204$; SCim sessions, $p = 0.470$).

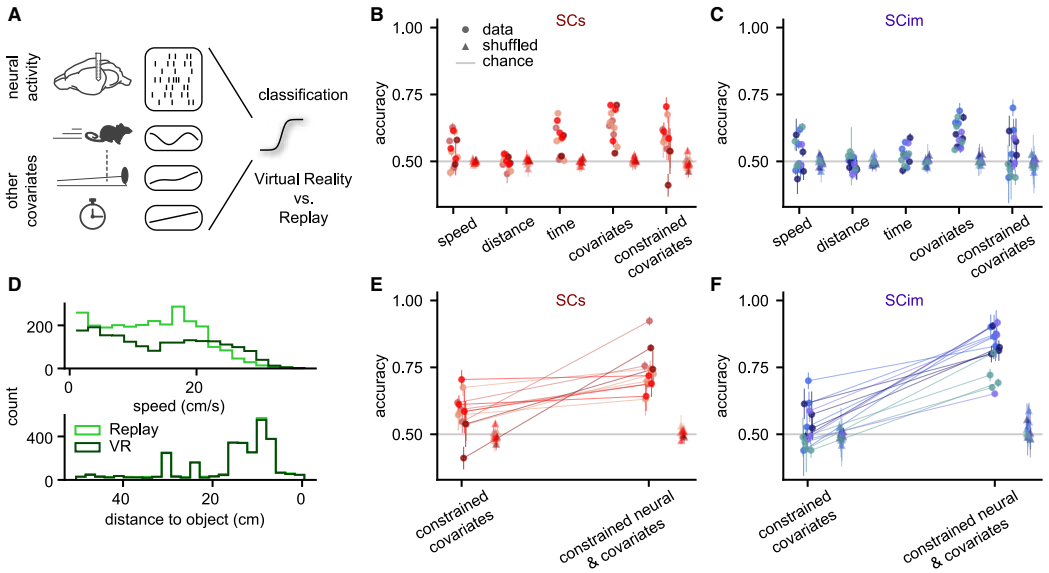


Figure 4. Neural activity in SC differs between virtual reality and replay conditions

(A) Logistic regression was used to classify epochs of time as belonging to virtual reality (VR) or replay given neural activity from SCs or SCim and/or other covariates, including running speed, distance to object, or time into the experiment.

(B) Test classification accuracy of 12 SCs sessions for individual covariates (speed, distance, and time) and all covariates combined. "Constrained covariates" indicates classification performance when using a constrained subset of the data (confining analyses to samples of VR and replay conditions with the same speed range, the same average time into the experiment, and constraining analyses to distances within 50 cm of the object). Circles indicate mean performance in each session, and triangles indicate mean of shuffled controls for each session. Mean and SD were obtained from the test sets across four train-test splits. Symbols of the same hue are sessions from the same animal.

(C) Same as (B) but for 15 SCim sessions.

(D) (Top) Distributions of locomotion speed differ in VR and replay conditions. (Bottom) Distributions of distance-to-object are similar in VR and replay.

(E) Test classification accuracy of SCs sessions when including all constrained covariates or when including neural activity as well as those covariates. Symbols of the same hue are sessions from the same animal.

(F) Same as (E) but for SCim sessions.

See also Figures S4 and S5.

We next tested whether including neural activity during these measurement epochs would improve model performance beyond that of the constrained covariates alone (Figures 4E and 4F). Using recorded spiking activity of populations of neurons alongside behavior, we calculated the mean firing rates of each neuron in the same epochs. We trained the classifier on the constrained subsets of the data for which measurement time and locomotion speed ranges were similar across VR and replay conditions (see STAR Methods, Figure S5). We found that neural activity in SCim improved classifier performance (mean ± SD: 56.1 ± 23.0 % increase, $p < 0.001$, $n = 15$ sessions; Figure 4F). Neural activity in SCs also improved classifier performance (28.8 ± 27.7% increase, $p < 0.001$; $n = 12$ sessions; Figure 4E) but to a lesser extent (SCim vs. SCs; $p = 0.005$, Mann-Whitney U test: 32.0). We confirmed through control analyses that these results were not dependent on the choice of constraints used, and conclusions remained unchanged when epochs with eye movements were excluded from the analyses (Figure S5). Thus, neural activity in SC, particularly SCim, depends on whether the visual stimulus is matched to that expected from the animal's locomotion.

DISCUSSION

We found that neuronal activity in SCs was dominated by vision but that vision and self-movement signals contribute to neuronal activity in SCim for both objects looming from ahead of the animal and during object approach. Animals instinctively slowed down when an object was near and further altered their behavior when visual experience did not match that expected from their locomotion. This difference between visual experience and expectation from locomotion behavior was also represented in population neural activity in SC, particularly in SCim.

Mice often ceased moving during replay of the sequence of images they had previously generated while approaching an object. The behavior likely reflects their instinctive ability to predict and account for the visual stimulation brought about by their self-movement. VR reality allowed us to generate a mismatch

between predicted and actual visual stimulus, and mice were sensitive to this discrepancy. Neural activity in SC was also sensitive to this mismatch. Although our analyses revealed behavioral differences between VR and replay conditions, SC activity could still distinguish between the two conditions when behavioral differences were accounted for. Although we cannot rule out the contribution of behaviors we are not measuring,[11,12] we controlled for the behavioral variables we did record as thoroughly as possible (e.g., through session selection, balancing time into the experiment, and speed range). We, therefore, infer that both animals and neural activity in SC, particularly SCim, can tell the difference between visual stimuli produced by object movement and self-movement.

Previous work has described subpopulations of neurons in primary visual cortex that are sensitive to brief disruptions (mismatch events) between predicted and actual visual stimulus (Keller et al.[13]; but see Muzzu and Saleem[14]). Similarly, response of neurons in postrhinal (POR) and laterointermediate (LI) visual cortical areas (but not primary visual cortex) to a small moving spot depends on whether the spot's movement is coupled to animal locomotion.[15] These latter cortical responses appear to depend on a subpopulation of SCs neurons ("wide-field cells"), and the pooled calcium signal of populations of wide-field SCs neurons is increased when a small spot's movement is uncoupled (either slower or faster) from animal movement.[15] Wide-field cells, which project from SCs to the pulvinar (lateral posterior nucleus), show weak responses to large visual stimuli like the object we presented.[5,16] Because our recordings were not designed to specify finer laminar structure within SCs or SCim, we do not know the extent of the contribution from wide-field cells to the observed differences in VR and replay conditions.

SC has access to visual sensory signals directly from retina[17] and via visual cortices[18] and can access locomotion-related and high-level contextual signals from multiple brain regions.[19,20] The different layers of SC have different anatomical connectivity (see Wheatcroft et al.,[2] Basso et al.,[21] Isa et al.,[22] Liu et al.,[23] and May[24] for reviews). SCs receives strong visual input from retina,[17] primary visual cortex, and some secondary visual areas,[18] consistent with our observations that most of the activity in SCs could be accounted for by the pattern of visual stimulation. By contrast, SCim receives weak, if any, retinal input, and cortical visual inputs are mainly derived from other secondary visual areas[18] that may form a "dorsal" stream through mouse visual cortex. Consistent with this anatomical connectivity, visual responses were weaker in SCim than in SCs, though we may have detected more visual responses in SCim if we had measured responses to a wider variety of stimuli, including looming discs centered on the receptive field of the neurons.[6,9,25] Receptive fields in SCim (where we could recover them) were also larger than those of most SCs neurons and similar in size to wide-field and "horizontal" SCs cell classes.[3,5,6,9,16,26–28] This may be related to the greater positional invariance reported in SCim.[9] In addition, we found substantial impact of locomotion on population activity in SCim, in line with limited previous work (Ito et al.[6]; but see Chen et al.[25]), and no net impact of locomotion on SCs population activity (see also Ito et al.,[6] Chen et al.,[25] Savier et al.,[29] Schröder et al.,[30] and Li et al.[31]). Note that, as the black expanding sphere elicited the animals' pupillary light reflex, we were unable to assess the relative contribution of locomotion and arousal in these measurements. The stronger impact of locomotion in SCim is also consistent with anatomical connectivity: SCim receives multi-modal sensory input from auditory and somatosensory pathways (e.g., Ito et al.[26] and Xie et al.[32]), as well as multiple non-sensory signals from subcortical and cortical areas (see Benavidez et al.[19] and Doykos et al.[20]), including inputs from retrosplenial cortex, prefrontal cortex, and striatum, among others (e.g., Campagner et al.,[33] Lee et al.,[34] and Ritter et al.[35]). These anatomical connections of SCim also make it an ideal site for establishing the VR context and broadcasting those signals.

In freely moving mice, overhead looming objects usually induce escape when a refuge is present and freezing behavior when a refuge is absent.[36–38] Looming objects in the frontal visual field can also induce escape to refuge, though usually after initial freezing behavior.[39] Our behavioral measurements show that head-restrained mice slow down when presented with looming objects in the frontal visual field, whether or not that visual loom is matched to their locomotion. Other visual behaviors that are readily elicited in untrained, head-restrained animals include "vidgetting" and locomotion changes produced by novel grating patterns,[40,41] "locomotion arrest" produced by flashed lights (Liang et al.[42]; see also Roseberry and Kreitzer[43]), and "burrow ingress" produced by overhead looms.[44] Slow-down behavior adds to this repertoire of instinctive behaviors in head-restrained animals; whether there is a relationship between them needs to be established.

Animals explored the virtual environment at moderate locomotion speeds of about 20 cm/s. The instinctive slow-downs we observed when animals were near an object may be adaptive, providing both time and motor state for a greater range of subsequent behaviors. Objects and agents (conspecifics or potential threat or prey) approaching at slow speed will evoke a wave of activity across SC during this period (Figure 1E), analysis of which could allow animals to choose among appropriate approach or avoidance behaviors (cf. Campagner et al.[33]). Striking predators will likely move more quickly (e.g., Garland[45]), and a rapidly approaching immediate threat will near simultaneously activate a large fraction of visually responsive neurons in both SCs and SCim (Figure 1D). The different spatiotemporal dynamics of population activity during slower object approach and fast object loom may therefore help allow rapid avoidance behaviors in the presence of immediate threat, while reducing the likelihood of false alarms.

Our measurements leveraged immersive VR to afford tight experimental control, while retaining expression of instinctive behaviors. Our findings demonstrate that both animals and neurons in SC can distinguish whether current visual experience matches that expected from the animal's actions. The convergence of visual and locomotion signals onto SC, and the fact that SC is well placed to instruct behavior, indicate that contextual variation in SC activity could be important in allowing animals to shape their behavior.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Samuel Solomon (s.solomon@ucl.ac.uk).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- Source data have been deposited on FigShare at https://doi.org/10.5522/04/29469893 and are publicly available.
- All code has been deposited on FigShare at https://doi.org/10.5522/04/29469893 and is publicly available.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

ACKNOWLEDGMENTS

This work was supported by the Biotechnology and Biological Sciences Research Council (BBSRC; BB/R004765/1), by the UKRI Frontier Research Grant (EU underwrite; EP/Y024656/1), and by the Human Frontier Science Program (RGY0076/2018). This work was also supported by the German Research Foundation (DFG) through Germany's Excellence Strategy (EXC-Number 2064/1, PN 390727645) and SFB 1233, the German Federal Ministry of Education and Research (Tübingen AI Center, FKZ: 01IS18039). A.S. is a member of the International Max Planck Research School for Intelligent Systems (IMPRS-IS). We thank Sarah Ruediger and Sylvia Schröder for comments on the manuscript and Henrik Singmann and Edward Horrocks for advice on analyses.

AUTHOR CONTRIBUTIONS

This work was conceptualized by S.Z., A.B.S., and S.G.S.; experiments were performed by S.Z.; formal analysis and visualization by S.Z., A.S., and S.G.S.; formal logistic regression analyses and visualization by A.S., P.J.G., and J.H.M.; original draft writing by S.Z., A.S., A.B.S., and S.G.S.; review and feedback by all authors; supervision and funding acquisition by S.G.S., A.B.S., and J.H.M.

DECLARATION OF INTERESTS

The authors declare no competing interests.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- EXPERIMENTAL MODEL AND SUBJECT DETAILS
- METHOD DETAILS
  - Surgery and recording
  - Visual stimulation
  - Spike sorting and clustering
  - Eye-tracking
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - Receptive field analysis
  - Response during object loom and object approach
  - Contribution of speed and distance-to-object
  - Response consistency
  - Linear mixed-effects analyses
  - Logistic regression
  - Classifier details
  - Inclusion criteria, constraints and control analyses
  - Code

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.cub.2025.07.013.

Received: February 3, 2025
Revised: June 3, 2025
Accepted: July 4, 2025
Published: July 31, 2025

REFERENCES

1. Dean, P., Redgrave, P., and Westby, G.W. (1989). Event or emergency? Two response systems in the mammalian superior colliculus. Trends Neurosci. *12*, 137–147. https://doi.org/10.1016/0166-2236(89)90052-0.
2. Wheatcroft, T., Saleem, A.B., and Solomon, S.G. (2022). Functional Organisation of the Mouse Superior Colliculus. Front. Neural Circuits *16*, 792959. https://doi.org/10.3389/fncir.2022.792959.
3. De Franceschi, G., and Solomon, S.G. (2018). Visual response properties of neurons in the superficial layers of the superior colliculus of awake mouse. J. Physiol. *596*, 6307–6332. https://doi.org/10.1113/JP276964.
4. Dräger, U.C., and Hubel, D.H. (1975). Responses to visual stimulation and relationship between visual, auditory, and somatosensory inputs in mouse superior colliculus. J. Neurophysiol. *38*, 690–713. https://doi.org/10.1152/jn.1975.38.3.690.
5. Gale, S.D., and Murphy, G.J. (2014). Distinct representation and distribution of visual information by specific cell types in mouse superficial superior colliculus. J. Neurosci. *34*, 13458–13471. https://doi.org/10.1523/JNEUROSCI.2768-14.2014.
6. Ito, S., Feldheim, D.A., and Litke, A.M. (2017). Segregation of Visual Response Properties in the Mouse Superior Colliculus and Their Modulation during Locomotion. J. Neurosci. *37*, 8428–8443. https://doi.org/10.1523/JNEUROSCI.3689-16.2017.
7. Wang, L., Sarnaik, R., Rangarajan, K., Liu, X., and Cang, J. (2010). Visual receptive field properties of neurons in the superficial superior colliculus of the mouse. J. Neurosci. *30*, 16573–16584. https://doi.org/10.1523/JNEUROSCI.3305-10.2010.
8. González-Rueda, A., Jensen, K., Noormandipour, M., de Malmazet, D., Wilson, J., Ciabatti, E., Kim, J., Williams, E., Poort, J., Hennequin, G., et al. (2024). Kinetic features dictate sensorimotor alignment in the superior colliculus. Nature *631*, 378–385. https://doi.org/10.1038/s41586-024-07619-2.
9. Lee, K.H., Tran, A., Turan, Z., and Meister, M. (2020). The sifting of visual information in the superior colliculus. eLife *9*, e50678. https://doi.org/10.7554/eLife.50678.
10. Saleem, A.B., Ayaz, A., Jeffery, K.J., Harris, K.D., and Carandini, M. (2013). Integration of visual motion and locomotion in mouse visual cortex. Nat. Neurosci. *16*, 1864–1869. https://doi.org/10.1038/nn.3567.
11. Musall, S., Kaufman, M.T., Juavinett, A.L., Gluf, S., and Churchland, A.K. (2019). Single-trial neural dynamics are dominated by richly varied movements. Nat. Neurosci. *22*, 1677–1686. https://doi.org/10.1038/s41593-019-0502-4.
12. Stringer, C., Pachitariu, M., Steinmetz, N., Reddy, C.B., Carandini, M., and Harris, K.D. (2019). Spontaneous behaviors drive multidimensional, brainwide activity. Science *364*, 255. https://doi.org/10.1126/science.aav7893.
13. Keller, G.B., Bonhoeffer, T., and Hübener, M. (2012). Sensorimotor mismatch signals in primary visual cortex of the behaving mouse. Neuron *74*, 809–815. https://doi.org/10.1016/j.neuron.2012.03.040.
14. Muzzu, T., and Saleem, A.B. (2021). Feature selectivity can explain mismatch signals in mouse visual cortex. Cell Rep. *37*, 109772. https://doi.org/10.1016/j.celrep.2021.109772.
15. Brenner, J.M., Beltramo, R., Gerfen, C.R., Ruediger, S., and Scanziani, M. (2023). A genetically defined tecto-thalamic pathway drives a system of superior-colliculus-dependent visual cortices. Neuron *111*, 2247–2257. e7. https://doi.org/10.1016/j.neuron.2023.04.022.
16. Hoy, J.L., Bishop, H.I., and Niell, C.M. (2019). Defined Cell Types in Superior Colliculus Make Distinct Contributions to Prey Capture Behavior in the Mouse. Curr. Biol. *29*, 4130–4138.e5. https://doi.org/10.1016/j.cub.2019.10.017.

17. Ellis, E.M., Gauvain, G., Sivyer, B., and Murphy, G.J. (2016). Shared and distinct retinal input to the mouse superior colliculus and dorsal lateral geniculate nucleus. J. Neurophysiol. *116*, 602–610. https://doi.org/10.1152/jn.00227.2016.

18. Wang, Q., and Burkhalter, A. (2013). Stream-related preferences of inputs to the superior colliculus from areas of dorsal and ventral streams of mouse visual cortex. J. Neurosci. *33*, 1696–1705. https://doi.org/10.1523/JNEUROSCI.3067-12.2013.

19. Benavidez, N.L., Bienkowski, M.S., Zhu, M., Garcia, L.H., Fayzullina, M., Gao, L., Bowman, I., Gou, L., Khanjani, N., Cotter, K.R., et al. (2021). Organization of the inputs and outputs of the mouse superior colliculus. Nat. Commun. *12*, 4004. https://doi.org/10.1038/s41467-021-24241-2.

20. Doykos, T.K., Gilmer, J.I., Person, A.L., and Felsen, G. (2020). Monosynaptic inputs to specific cell types of the intermediate and deep layers of the superior colliculus. J. Comp. Neurol. *528*, 2254–2268. https://doi.org/10.1002/cne.24888.

21. Basso, M.A., Bickford, M.E., and Cang, J. (2021). Unraveling circuits of visual perception and cognition through the superior colliculus. Neuron *109*, 918–937. https://doi.org/10.1016/j.neuron.2021.01.013.

22. Isa, T., Marquez-Legorreta, E., Grillner, S., and Scott, E.K. (2021). The tectum/superior colliculus as the vertebrate solution for spatial sensory integration and action. Curr. Biol. *31*, R741–R762. https://doi.org/10.1016/j.cub.2021.04.001.

23. Liu, X., Huang, H., Snutch, T.P., Cao, P., Wang, L., and Wang, F. (2022). The Superior Colliculus: Cell Types, Connectivity, and Behavior. Neurosci. Bull. *38*, 1519–1540. https://doi.org/10.1007/s12264-022-00858-1.

24. May, P.J. (2006). The mammalian superior colliculus: laminar structure and connections. Prog. Brain Res. *151*, 321–378. https://doi.org/10.1016/S0079-6123(05)51011-2.

25. Chen, H., Savier, E.L., DePiero, V.J., and Cang, J. (2021). Lack of Evidence for Stereotypical Direction Columns in the Mouse Superior Colliculus. J. Neurosci. *41*, 461–473. https://doi.org/10.1523/JNEUROSCI.1155-20.2020.

26. Ito, S., Si, Y., Litke, A.M., and Feldheim, D.A. (2021). Nonlinear visuoauditory integration in the mouse superior colliculus. PLOS Comput. Biol. *17*, e1009181. https://doi.org/10.1371/journal.pcbi.1009181.

27. Li, Y.T., and Meister, M. (2023). Functional cell types in the mouse superior colliculus. eLife *12*, e82367. https://doi.org/10.7554/eLife.82367.

28. Wang, L., Herman, J.P., and Krauzlis, R.J. (2022). Neuronal modulation in the mouse superior colliculus during covert visual selective attention. Sci. Rep. *12*, 2482. https://doi.org/10.1038/s41598-022-06410-5.

29. Savier, E.L., Chen, H., and Cang, J. (2019). Effects of Locomotion on Visual Responses in the Mouse Superior Colliculus. J. Neurosci. *39*, 9360–9368. https://doi.org/10.1523/JNEUROSCI.1854-19.2019.

30. Schröder, S., Steinmetz, N.A., Krumin, M., Pachitariu, M., Rizzi, M., Lagnado, L., Harris, K.D., and Carandini, M. (2020). Arousal Modulates Retinal Output. Neuron *107*, 487–495.e9. https://doi.org/10.1016/j.neuron.2020.04.026.

31. Li, C., Kühn, N.K., Alkislar, I., Sans-Dublanc, A., Zemmouri, F., Paesmans, S., Calzoni, A., Ooms, F., Reinhard, K., and Farrow, K. (2023). Pathway-specific inputs to the superior colliculus support flexible responses to visual threat. Sci. Adv. *9*, eade3874. https://doi.org/10.1126/sciadv.ade3874.

32. Xie, Z., Wang, M., Liu, Z., Shang, C., Zhang, C., Sun, L., Gu, H., Ran, G., Pei, Q., Ma, Q., et al. (2021). Transcriptomic encoding of sensorimotor transformation in the midbrain. eLife *10*, e69825. https://doi.org/10.7554/eLife.69825.

33. Campagner, D., Vale, R., Tan, Y.L., Iordanidou, P., Pavón Arocas, O., Claudi, F., Stempel, A.V., Keshavarzi, S., Petersen, R.S., Margrie, T.W., et al. (2023). A cortico-collicular circuit for orienting to shelter during escape. Nature *613*, 111–119. https://doi.org/10.1038/s41586-022-05553-9.

34. Lee, J., Wang, W., and Sabatini, B.L. (2020). Anatomically segregated basal ganglia pathways allow parallel behavioral modulation. Nat. Neurosci. *23*, 1388–1398. https://doi.org/10.1038/s41593-020-00712-5.

35. Ritter, A., Habusha, S., Givon, L., Edut, S., and Klavir, O. (2024). Prefrontal control of superior colliculus modulates innate escape behavior following adversity. Nat. Commun. *15*, 2158. https://doi.org/10.1038/s41467-024-46460-z.

36. De Franceschi, G., Vivattanasarn, T., Saleem, A.B., and Solomon, S.G. (2016). Vision Guides Selection of Freeze or Flight Defense Strategies in Mice. Curr. Biol. *26*, 2150–2154. https://doi.org/10.1016/j.cub.2016.06.006.

37. Wei, P., Liu, N., Zhang, Z., Liu, X., Tang, Y., He, X., Wu, B., Zhou, Z., Liu, Y., Li, J., et al. (2015). Processing of visually evoked innate fear by a non-canonical thalamic pathway. Nat. Commun. *6*, 6756. https://doi.org/10.1038/ncomms7756.

38. Yilmaz, M., and Meister, M. (2013). Rapid innate defensive responses of mice to looming visual stimuli. Curr. Biol. *23*, 2011–2015. https://doi.org/10.1016/j.cub.2013.08.015.

39. Solomon, S.G., Janbon, H., Bimson, A., and Wheatcroft, T. (2023). Visual spatial location influences selection of instinctive behaviours in mouse. R. Soc. Open Sci. *10*, 230034. https://doi.org/10.1098/rsos.230034.

40. Cooke, S.F., Komorowski, R.W., Kaplan, E.S., Gavornik, J.P., and Bear, M.F. (2015). Visual recognition memory, manifested as long-term habituation, requires synaptic plasticity in V1. Nat. Neurosci. *18*, 262–271. https://doi.org/10.1038/nn.3920.

41. Papanikolaou, A., Rodrigues, F.R., Holeniewska, J., Phillips, K.G., Saleem, A.B., and Solomon, S.G. (2022). Plasticity in visual cortex is disrupted in a mouse model of tauopathy. Commun. Biol. *5*, 77. https://doi.org/10.1038/s42003-022-03012-9.

42. Liang, F., Xiong, X.R., Zingg, B., Ji, X.Y., Zhang, L.I., and Tao, H.W. (2015). Sensory Cortical Control of a Visually Induced Arrest Behavior via Corticotectal Projections. Neuron *86*, 755–767. https://doi.org/10.1016/j.neuron.2015.03.048.

43. Roseberry, T., and Kreitzer, A. (2017). Neural circuitry for behavioural arrest. Philos. Trans. R. Soc. Lond. B Biol. Sci. *372*, 20160197. https://doi.org/10.1098/rstb.2016.0197.

44. Fink, A.J., Axel, R., and Schoonover, C.E. (2019). A virtual burrow assay for head-fixed mice measures habituation, discrimination, exploration and avoidance without training. eLife *8*, e45658. https://doi.org/10.7554/eLife.45658.

45. Garland, T.J. (1983). The relation between maximal running speed and body mass in terrestrial mammals. J. Zool. *199*, 157–170. https://doi.org/10.1111/j.1469-7998.1983.tb02087.x.

46. Pachitariu, M., Sridhar, S., Pennington, J., and Stringer, C. (2024). Spike sorting with Kilosort4. Nat. Methods *21*, 914–921. https://doi.org/10.1038/s41592-024-02232-7.

47. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. *12*, 2825–2830.

48. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2.

49. Siegle, J.H., López, A.C., Patel, Y.A., Abramov, K., Ohayon, S., and Voigts, J. (2017). Open Ephys: an open-source, plugin-based platform for multi-channel electrophysiology. J. Neural Eng. *14*, 045003. https://doi.org/10.1088/1741-2552/aa5eea.

50. Lopes, G., Bonacchi, N., Frazão, J., Neto, J.P., Atallah, B.V., Soares, S., Moreira, L., Matias, S., Itskov, P.M., Correia, P.A., et al. (2015). Bonsai: an event-based framework for processing and controlling

data streams. Front. Neuroinform. *9*, 7. https://doi.org/10.3389/fninf.2015.00007.

51. Lopes, G., Farrell, K., Horrocks, E.A., Lee, C.Y., Morimoto, M.M., Muzzu, T., Papanikolaou, A., Rodrigues, F.R., Wheatcroft, T., Zucca, S., et al. (2021). Creating and controlling visual environments using BonVision. eLife *10*, e65541. https://doi.org/10.7554/eLife.65541.

52. Shang, C., Chen, Z., Liu, A., Li, Y., Zhang, J., Qu, B., Yan, F., Zhang, Y., Liu, W., Liu, Z., et al. (2018). Divergent midbrain circuits orchestrate escape and freezing responses to looming stimuli in mice. Nat. Commun. *9*, 1232. https://doi.org/10.1038/s41467-018-03580-7.

53. Zhao, X., Liu, M., and Cang, J. (2014). Visual cortex modulates the magnitude but not the selectivity of looming-evoked responses in the superior colliculus of awake mice. Neuron *84*, 202–213. https://doi.org/10.1016/j.neuron.2014.08.037.

54. Rossant, C., Kadir, S.N., Goodman, D.F.M., Schulman, J., Hunter, M.L.D., Saleem, A.B., Grosmark, A., Belluscio, M., Denfield, G.H., Ecker, A.S., et al. (2016). Spike sorting for large, dense electrode arrays. Nat. Neurosci. *19*, 634–641. https://doi.org/10.1038/nn.4268.

55. Sibille, J., Gehr, C., Benichov, J.I., Balasubramanian, H., Teh, K.L., Lupashina, T., Vallentin, D., and Kremkow, J. (2022). High-density electrode recordings reveal strong and specific connections between retinal ganglion cells and midbrain neurons. Nat. Commun. *13*, 5218. https://doi.org/10.1038/s41467-022-32775-2.

56. Fritsch, F.N., and Butland, J. (1984). A method for constructing local monotone piecewise cubic interpolants. SIAM journal on scientific and statistical computing *5* (2), 300–304.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Chemicals, peptides, and recombinant proteins		
Isofluorane	Primal Critical Care	CAS #26675-46-7
KwikCast	World Precision Instrument	KWIK-CAST
Deposited data		
Data and code for data analysis post-processing and analysis	This Paper	https://doi.org/10.5522/04/29469893
Experimental models: Organisms/strains		
Mouse: C57BL6/J	Charles River Laboratories	RRID: IMSR_JAX:000664
Software and algorithms		
MATLAB2021b & MATLAB2022a	Mathworks	https://www.mathworks.com/
Kilosort 2	Pachitariu et al.[46]	https://github.com/jamesjun/Kilosort2
Python	Python Software Foundation	https://www.python.org/
Sklearn	Pedregosa et al.[47]	https://github.com/scikit-learn/scikit-learn
SciPy	Virtanen et al.[48]	https://scipy.org/citing-scipy/
OpenEphys	Siegle et al.[49]	https://open-ephys.org/
Bonsai 4.3	Lopes et al.[50]	https://bonsai-rx.org/
BonVision	Lopes et al.[51]	https://bonvision.github.io/info/Home/#

EXPERIMENTAL MODEL AND SUBJECT DETAILS

All experiments were performed in accordance with the Animals (Scientific Procedures) Act 1986 (United Kingdom) and Home Office (United Kingdom) approved project and personal licences. Mice (n = 16 C57BL6/J male wildtype, age 12-16 weeks) were housed in groups of maximum five under a 12-hour light/dark cycle, with free access to food and water. All electrophysiological recordings were carried out during the dark phase of the cycle.

METHOD DETAILS

Surgery and recording

Mice were anaesthetised with isoflurane and the skull exposed under aseptic conditions. A custom-built stainless-steel metal plate was attached to the skull with dental cement and a metal screw was implanted over the somatosensory cortex on the right hemisphere, for future use as a reference electrode. The skull above SC was left accessible. Animals were allowed to recover from surgery for at least seven days; analgesia was provided for at least the first three days. Mice were then habituated to the experimental apparatus, as described in Visual Stimulus section below. Following habituation, mice were briefly re-anesthetized with isoflurane and a craniotomy was performed over the SC on the right hemisphere, centred at 0.75 mm lateral to sagittal midline, and at lambda), and the dura was left intact. The craniotomy was sealed with silicon elastomer (KwikCast, World Precision Instruments).

Mice were allowed to recover for at least 4 hours before the first recording session. Multiple, daily recordings (3 - 6 sessions, one session per day) were then made in each animal. In each session, the animal was head-restrained in the apparatus, the craniotomy was exposed and a silicon probe, comprising two shanks each with 16 electrodes in a 'V' formation (spacing 250 μm between shanks, 40 μm between sites, 300 μm total depth; ASSY-37 E-1, Cambridge Neurotech Ltd, Cambridge, UK), was implanted using a vertical micromanipulator (Sensapex, Finland). Electrophysiological signals were acquired using an OpenEphys acquisition board[49] at a rate of 30 kHz. The electrodes were lowered rapidly to a depth of approximately 0.5 mm below the brain surface, and then lowered slowly while displaying a large flickering (2 Hz) checkerboard across the visual field. The depth of the dorsal surface of SC was identified by the appearance of robust neural activity modulated at the stimulus frequency; the electrodes were then lowered further until the tip

was 500 μm below SC surface (for recordings from SCs) or was 800 μm below (for recordings from SCim). At the end of the recording session the electrodes were retracted, and the craniotomy was resealed as above. To confirm electrode targeting the electrode was immersed into a DiI solution before the last recording session. Following that session, mice were deeply anaesthetized and perfused transcardially, and brains were extracted for histological analysis.

Visual stimulation

We used an experimental apparatus described previously.[14] Mice were head-fixed above a polystyrene wheel (radius 10 cm), such that their head was in the centre of a truncated spherical dome. Visual stimuli were produced by shining the light of a projector (Casio Green Slim XJ-A257-UJ DLP; 60 Hz refresh rate), onto the internal surface of the dome via an hemispherical mirror; the projected image spanned 240° azimuth (from -120° to 120°) and 120° elevation (from -30° and 90°) with mean luminance ca. 10 cd.m^2. Mesh-mapped and gamma-calibrated visual stimuli were produced by the package BonVision,[51] in the Bonsai framework.[50] Movement of the polystyrene wheel was sensed with a rotary encoder (2400 pulses/rotation, Kübler, Germany), the output of which was copied to both the OpenEphys acquisition board and to an Arduino connected to the stimulus computer, so as to allow locomotion-dependent updating of the visual scene where required. A synchronising signal was sent to both the OpenEphys board and the stimulus computer. The OpenEphys board also acquired the signals of a photodiode (PDA25K2, Thorlabs Inc., USA) that detected light in a small region of the dome hidden from the animal, and provided additional confirmation of stimulus timing. Video recordings of the eye were obtained by a camera (DMK 27BUR0135, The Imaging Source) equipped with a zoom lens (MLH10X, Computar).

We presented three classes of stimulus in the virtual environment: 1) platform: a static virtual platform formed by a smoothed random visual texture (8 cm wide x 100 cm long, rendered 2 cm below the animal's eye); 2) static object: a round object (8 cm diameter) sitting on the distal end of the platform; 3) looming object: the same round object sitting on the distal end of the platform, but which then moved along the platform towards the animal at one of 12.5, 25, 50 or 100 cm/s, and disappeared after it had collided with the animal (at delays of, respectively, 0.65s, 0.35s, 0.15s and 0.016s). For looms of 100 cm/s, the object appeared on the platform and remained stationary for 1s before moving towards the animal. We note that neural responses to a looming visual object has often been measured after centering the stimulus on a neuron's receptive field (e.g. [9,52,53]), but here objects always loomed from the front of the animal, and the edge of the stimulus therefore always moved across the visual field in the nasal-temporal direction.

Animals were habituated to the virtual environment for 8-13 days (one session per day) before recordings started. Animals did not receive rewards during any of the habituation or recording sessions. On each habituation day the animal was placed in the virtual reality apparatus for up to 30 mins, during which it moved along the virtual platform by running on the treadmill. On each of the last 4 habituation days, the static object was presented at the end of the virtual platform on a subset of randomly interleaved trials. Animals (n = 16) experienced an average 55±30 (mean ± s.d.) trials of the platform, and 52±23 trials with the static object, on each of these days. Each trial was separated by 2s of grey screen. Subsequent recording sessions started with 5 consecutive platform trials, then 40 trials of the static object that were randomly interleaved with 20 trials of a looming object moving at 100 cm/s. In the virtual reality ('VR', or 'closed-loop') condition, the distance travelled on the polystyrene wheel was used by the stimulus generator to control the position of the animal on the virtual platform (Videos S1 and S2). In the 'replay' (or 'open-loop') condition, the movies produced during the VR condition were replayed to the animal, independent of the animal's movement on the treadmill. In 2/16 animals, the 40 VR trials and 20 looming object trials were presented in a single block, and these trials then replayed in a single block. In 14/16 animals, we split the 40 trials so that animals experienced 20 trials in VR then 20 trials in replay condition, and then repeated this process. Subsequently, we presented randomly interleaved trials of a looming object moving at different speeds (30 trials/speed); each trial was preceded by 5s of grey screen. To map receptive fields we then obtained responses to flashed black or white squares (15° wide). On each of 3000 0.1s trials (no interstimulus interval), we presented black or white squares at each of 5 random locations drawn randomly from an 8x8 grid centred in the left hemifield. The stimuli covered -15:105° azimuth, and -30:90° elevation, where 0° is directly ahead of the animal and at eye height.

Spike sorting and clustering

Electrophysiological signals from all recordings in a session were concatenated and processed using Kilosort 2 and Phy.[46,54] We kept for further analysis those units in which minimum inter-spike interval was greater than 1 ms, yielding a total of 828 units in SCs (28±10 per session, mean±S.D.; 30 sessions in 7 animals), and 1024 in SCim (29±8 per session; 35 sessions in 9 animals). Our recordings were not designed to identify putative retinal action potentials.[55] Subsequent analyses were performed in MATLAB R2021b or 2022a. Neural firing rate was resampled to the refresh-rate of the visual environment (60 Hz). Except for receptive field analyses, these binned rates were transformed into z-scores by normalising to the mean and standard deviation of firing rate across all stimulus conditions. To assess running behaviour, locomotion speed was also resampled at 60 Hz and animals were defined as 'stationary' if locomotion speed was less than 2 cm/s, and as 'running' otherwise.

Eye-tracking

Pupil size and position was estimated on-line using Bonsai video processing functions. Briefly, the image was cropped and thresholded, and the resultant image analysed by *FindContours* and *BinaryRegionAnalysis*. As the object is a black sphere that expands as the animals nears it, the mean luminance of the display reduced with distance to the object. Pupil size naturally increased (due to the pupillary light reflex) when the object was near and therefore, we could not use pupil size as an index of arousal.

QUANTIFICATION AND STATISTICAL ANALYSIS

Statistical tests, unit of analysis and sample sizes are included in the main text or in figure legends when not specified below.

Receptive field analysis

Units were included for further analysis if they produced at least 100 spikes during the stimulus sequence (yielding 711 units in SCs, and 830 in SCim). We analysed responses to black and white stimuli separately. For each unit, a linear regression (function *fitlm* in MATLAB) was conducted between a vector representation of the stimulus contrast at each location, and mean firing rate over the frame. The five parameters of a circular two-dimensional Gaussian (x- and y-location, standard deviation, amplitude, and offset) that best predicted the spatial pattern of regression weights were found using constrained least-squares optimisation (function *lsqcurvefit* in MATLAB). Size (standard deviation) of the receptive field was constrained to be at least 3.75°. Fits were included for further analysis if the predicted location of the receptive field centre was at least 7.5° from the edge of the stimulus grid, and the fits predicted at least 33% of the variance in the regression weights. Of the 711 SCs units, 569 yielded acceptable fits for black stimuli and 351 for white, of which 38 were acceptable only for white; of the 830 SCim units, 219 yielded acceptable fits for black stimuli and 46 for white, of which 11 were acceptable only for white. Receptive fields could be defined for black squares in most of these units. Since the looming object was also black, we used responses to black squares for further analysis. Note we will overestimate receptive field size, particularly in SCs, because receptive field size was poorly constrained when units responded to only one location in the stimulus grid. The model returned receptive field size less than 5 degrees in 240 SCs units, and 10 SCim units.

Response during object loom and object approach

For analyses of responses to object looms in Figure 1, binned and z-scored firing rates on each trial were aligned to the time at which the object collided with the animal in the virtual environment. Units were included for further analysis if they met the criteria for response consistency described below. Response amplitude was calculated as the mean z-score in the 0.2 s (for 100 cm/s objects) or 0.8s (for 25 cm/s) preceding collision. For analyses of responses during object approach in Figure 2, the binned and z-scored firing rate was first smoothed with a Gaussian filter (0.3 s). We then discretized the distance between the animal and object into non-overlapping bins of 1cm, and found the time-bins in each trial where the animal was at each of those distances. We then calculated time- and trial-averaged neural activity, and locomotion speed, at each of the 100 distances-to-object. Units were included for further analysis if they met the criteria for response consistency as below.

Contribution of speed and distance-to-object

To evaluate the relative contribution of locomotion speed and distance-to-object in explaining SC responses in Figure 2C, we used methods previously applied to responses in mouse primary visual cortex.[10] Distance-to-object was discretised into 110 bins; locomotion speed into 20 bins. We fit a linear model to obtain weights for each bin of the predictor variables, i.e. $y_n = w_n X$, where y_n is the response of the n^{th} neuron over time, X is the matrix of predictor variable bins, and w_n is an array of weights acting on the predictor variable bins to capture the response of the n^{th} neuron. We used linear ridge regression to calculate the weights based on the equation: $w'_n = (X^T X + \lambda I)^{-1} X^T y_n$, where λ was the ridge parameter tested (one of 0.001, 0.1, 1, 5, 10). We chose λ to minimise cross-validated explained variance. To calculate cross-validated explained variance, we first calculated the weights using the training data (80% of data) and then used the following equation to calculate performance in the test data:

$$Performance = 1 - \frac{\Sigma(y_n - w_{-w} X)^2}{\Sigma(y_n - \mu_{train})^2}$$

where μ_{train} is the mean of the response in the training period. Performance of each of three models (locomotion speed only, distance-to-object only, or both) was calculated on the best λ of each of the models independently.

Response consistency

For each of 30 iterations we split trials randomly into two equally sized sets and binned activity as a function of time to collision (object loom) or distance-to-object (object approach). We averaged activity in each bin over trials and calculated Pearson's correlation coefficient between the two resultant vectors. We then took one of these sets and randomly shuffled the time- or distance-bins on each trial before creating the average vector, and then calculated the Pearson's correlation with the other unshuffled, trial-averaged vector as above. The procedure was also repeated 30 times and yielded 30 estimates of raw correlations and 900 estimates of shuffle correlations. Units in which the mean raw correlation exceeded 95% of shuffle correlations were considered consistent.

Linear mixed-effects analyses

Where we included behaviour in our analyses of activity or model fits across individual units, our statistical analyses made use of linear mixed-effects (LME) models, using *fitlme* in MATLAB R2022a, and included individual animal as a random effect. The LMEs had the form

$$Y \sim X * layer + (X \mid animal)$$

For analyses of running modulation, Y was (z-scored) activity, and X was State (running or stationary). For analyses of ridge regression model fits, Y was model performance, and X was Model (distance or speed).

Logistic regression

We used logistic regression to ask whether the VR and replay conditions were associated with distinct patterns of neural activity. Classification using logistic regression was performed on epochs of 0.2 s that belonged to either condition. We compared the performance of regressions based on either behavioural and other covariates alone, or when also including neural population activity in SCim or SCs.

Classifier details

We divided data into non-overlapping 0.2 s epochs, for each of which we calculated different features (e.g., locomotion speed, z-scored neural activity, etc.) as predictors. We performed min-max scaling for each feature (neural and covariate) separately to ensure fair feature weighting. We classified epochs as belonging to VR or replay conditions using logistic regression with sparsity-inducing L1-regularisation (C=1.0). For each session, we trained a separate logistic regression model. For cross-validation, we generated four train and test splits (described below) and calculated accuracy as the average over the four test sets (mean and s.d.). In defining training and test epochs, we introduced a buffer period between them to account for any autocorrelations in the neural data, and prevent information leakage between training and test sets. Thus, 40 consecutive training epochs were separated from 10 test epochs by a buffer phase of 5 seconds. In control analyses, we tested classifier performance by shuffling test labels between the VR and replay conditions. Missing values in running speed data were interpolated using a monotonic cubic interpolation method.[56] We applied a moving average filter over 81 time steps to correct for slight jitter in wheel movements, followed by a median filter with a kernel of 51 time steps. Neural activity from units that contained missing values were excluded from further analyses.

Inclusion criteria, constraints and control analyses

We analysed sessions where we ran two repetitions of VR and replay conditions. In our experiments, the distributions of running speed were often very different between VR and replay conditions (Figure 3). As running speed can have a strong influence on neural activity (Figures 1 and 2), we included only sessions with comparable running behaviour in VR and replay sessions. We calculated the fraction of time where the running speed was greater than 2 cm/s ('fraction-running'). Sessions with fraction-running in the first replay condition exceeding 50% of fraction-running in the first VR condition were included in the classification analysis, except for 3 sessions where visual inspection showed unusual running behaviour (highlighted in Figures S5A and S5B). Consequently, our analyses exclude 5 SCs and 16 SCim sessions (Figures S3 and S5C).

We first established classification performance for non-neural features ('covariates': locomotion speed, distance to object, time into the experiment), and combinations of these features. Because these covariates could successfully classify many epochs, we applied feature-specific constraints ('constrained covariates') to minimise their influence when evaluating the contribution of neural activity to classification accuracy, as follows:

1) Given the block design of VR and replay conditions, and the potential for slow changes in neural activity, we balanced the average time into the experiment by including data from the last 2/3rd of the first VR condition, and an equivalent time from the start of the second VR session (Figure S5D).
2) As the strongest variation in neural activity was present as the animal neared the object, we constrained analyses to epochs when the animal was within 50 cm of the object. Including the full distance, or excluding 25 cm closest to the object did not change outcomes (Figures S5E and S5F).
3) To ensure a common range of locomotion speeds across conditions, we excluded epochs with running speeds that were present in only one condition.

To evaluate whether performance accuracy was above chance level for different predictors, we performed a one-sample Wilcoxon signed-rank test comparing the mean accuracy of all sessions to a chance level of 0.5 using the implementation of the scipy.stats package.[48] To assess the difference in prediction performance between models incorporating neural activity and covariates versus covariates alone, we used a paired Wilcoxon signed-rank test. This test compared the mean classification accuracy of all sessions with and without neural activity as a predictor.

Code

(Pre)processing, analysis, and classification code was written in Python using the Sklearn[47] toolbox.

Current Biology, Volume *35*

Supplemental Information

Visual loom caused by self-movement or object-movement elicits distinct responses in mouse superior colliculus

Stefano Zucca, Auguste Schulz, Pedro J. Gonçalves, Jakob H. Macke, Aman B. Saleem, and Samuel G. Solomon

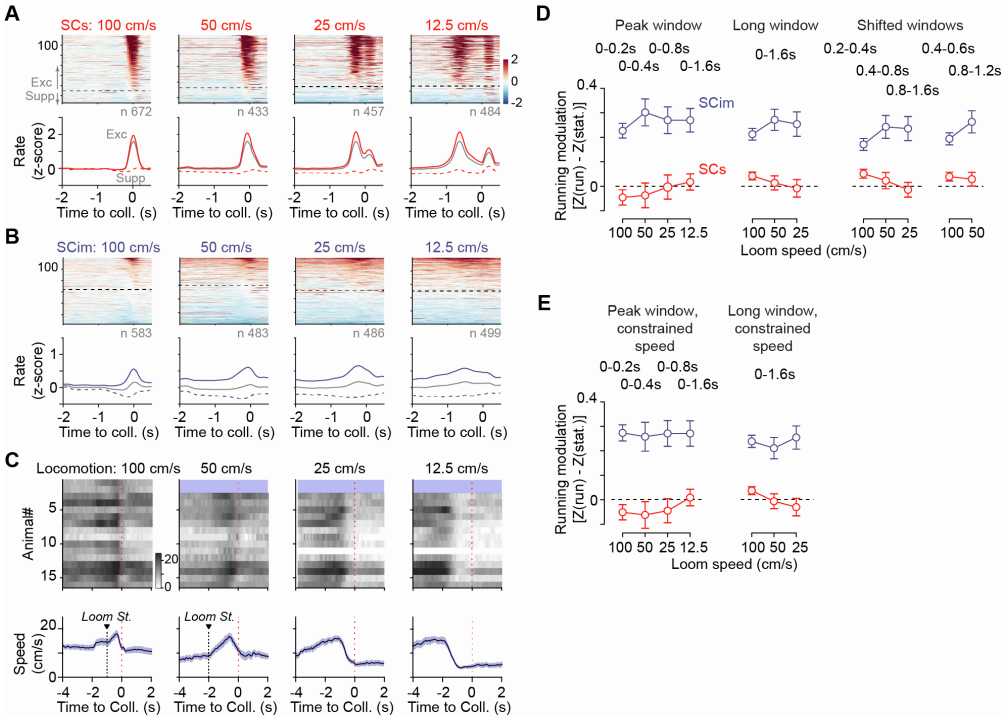


Figure S1. Response of SCs and SCim units to objects looming at different speeds. Related to Figure 1. A. Responses of units in SCs to objects looming at the animal at speeds of 100, 50, 25 and 12.5 cm/s (left to right). Conventions as in Figure 1D. At each speed, only units in which the response at that speed was consistent (see Methods) were included. Responses were sorted by the amplitude of the response in the 0.2 s (100 cm/s), 0.4 s (50 cm/s), 0.8 s (25 cm/s), or 1.6 s (12.5 cm/s) period preceding object collision. The second peak, beginning after 0 s at speeds of 25 and 12.5 cm/s, is a visual response triggered by the disappearance of the object after colliding with the animal in the virtual environment. **B.** Same as A but for units in SCim. **C**. (*upper*): Locomotion speed for each animal, averaged across trials. Red dashed lines show time of collision. Blue shaded area indicates that the first two animals were not exposed to slower looms. Colourbar indicates animal speed in cm/s, and applies to all panels. (*lower*): Average locomotion speed across all animals. Red dashed lines show time of collision, black dashed lines ('Loom St.') indicate when the object started to move in those trials. For 100 cm/s looms the object appeared and then remained stationary for 1 s before starting to move towards the animal. **D.** Difference in activity (in z-scores) between running and stationary states, during the different measurement epochs indicated above each panel. Each datum shows the mean and s.e.m. across units. **E.** Same as (D) but when 'running' trials were constrained to locomotion speeds 7-27 cm/s inclusive. Average locomotion speed among the constrained trials was 16.3 cm/s (s.d. 5.7) in SCs sessions and 16.7 cm/s (s.d. 5.5) in SCim sessions.

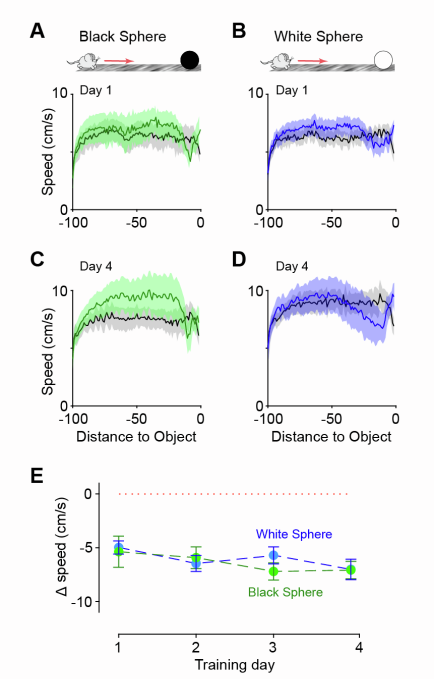


Figure S2. **Similar patterns in locomotor behaviour while approaching black and white objects/ Related to Figure 3. A-D**. Average running speed for 7 animals approaching a black sphere (green lines in A,C) or a white sphere (blue lines in B,D). **A&B**: First day of exposure. **C&D**: Fourth day of exposure. Black lines in A-D show locomotion speed when the sphere was absent ('Platform'). **E.** Reduction in locomotion speed as animals approached black (green lines) or white (blue lines) spheres, as a function of exposure day. Points show mean and s.e.m across animals. There was no significant difference between white and black objects on any of the four days.

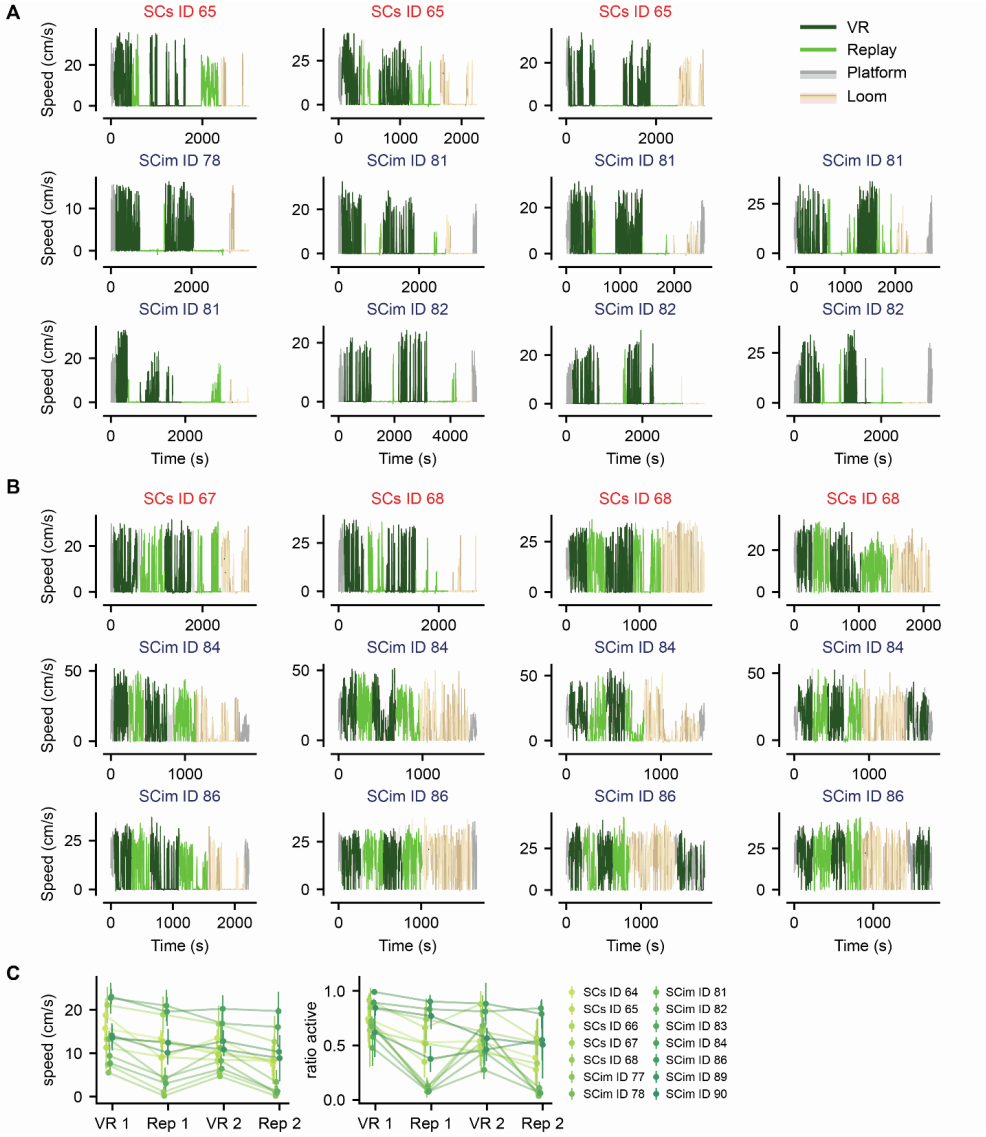


Figure S3. Running behaviour profiles across sessions and animals. Related to Figure 3. A. Temporal profile of locomotion speed for example sessions where overall locomotion behaviour during Replay was very different to that in VR. Each plot shows locomotion as a function of time into the recording session. In each case, line colour indicates condition (VR or Replay; Platform; Looming object). These sessions were excluded from the classification analysis. **B.** Same as A but showing example sessions where overall locomotion behaviour during Replay was similar to that in VR. These sessions were included in the classification analysis. **C.** Variability of running behaviour across animals. (*left*) average locomotion speed for each animal (N = 14). Each point shows mean ± standard deviation across sessions. Conventions as in Figure 3F. (*right*) Average time active for the same animals (fraction of time where locomotion speed was at least 0.1 cm/s).

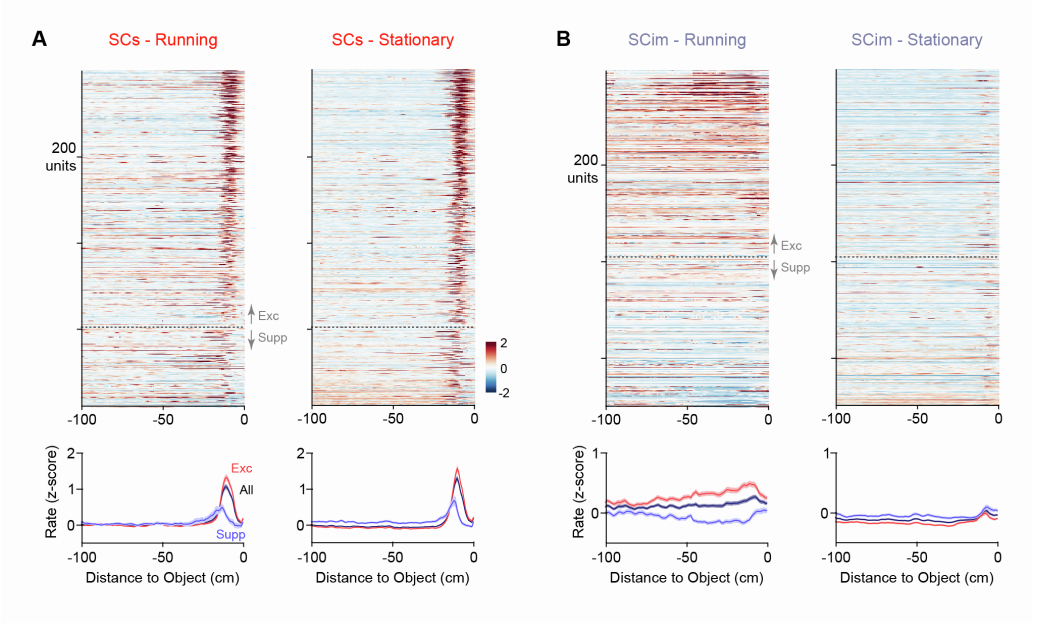


Figure S4. Activity in SCs and SCim during replay of object approach. Related to Figure 4. **A.** Activity of SCs responsive units during replay. Same units as in Figure 2A. Conventions as in Figure 2A. (*upper*) Each replayed trial was classified as presented while the animal was 'running' (average locomotion speed >2 cm/s) or 'stationary' (locomotion speed <=2 cm/s). Activity was averaged over all running trials (left) or all stationary trials (right). Units are ordered by their response in VR presented in Figure 2A, and the dashed line separating 'Exc' and 'Supp' groups is also reproduced from that in VR. (*lower*) Mean ± s.e.m. of activity in running (left) and stationary (right) replay trials, for all units (black), for 'Exc' units defined by VR responses (red) and for 'Supp' units defined by VR responses (blue). **B.** Same as A but for units recorded from SCim. Scale bar in A applies to all images.

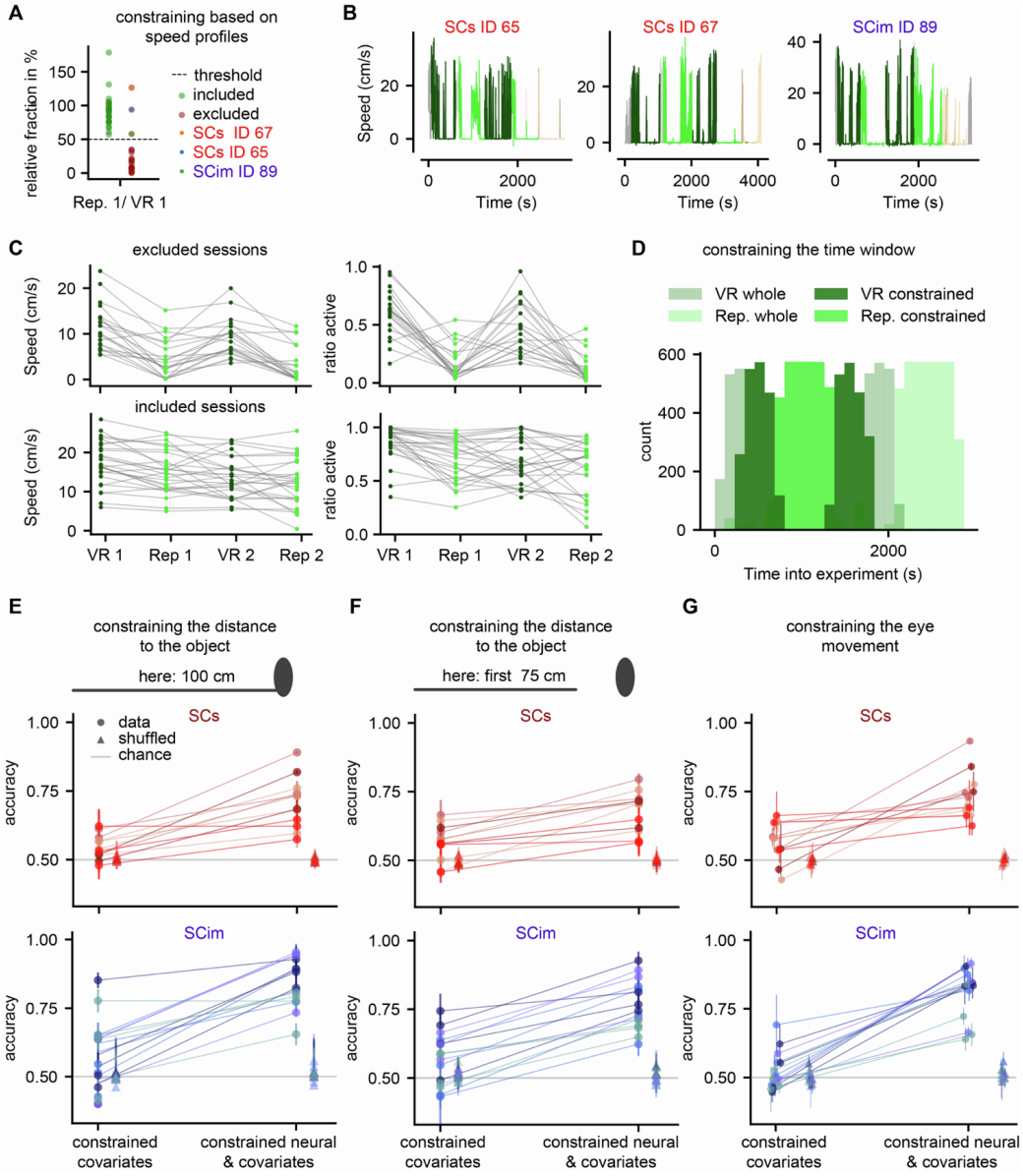


Figure S5. Session selection and data constraints for the classification analysis. Related to Figure 4. A. The 'fraction running' (percent of time when speed > 2cm/s) in Rep. 1 was at least 50% of that in VR 1 in all sessions included in the classification analysis. **B.** Three sessions above the 50% threshold were excluded due to sparse running behaviour (visual inspection). **C.** Mean speed (left) and fraction running (right) of excluded (top) and included (bottom) sessions in the classification analysis. Conventions are identical to those in Figure 3F. **D.** Distribution of time-into-experiment of epochs used in the classification experiment. To ensure that average time-into-experiment was similar in the VR and

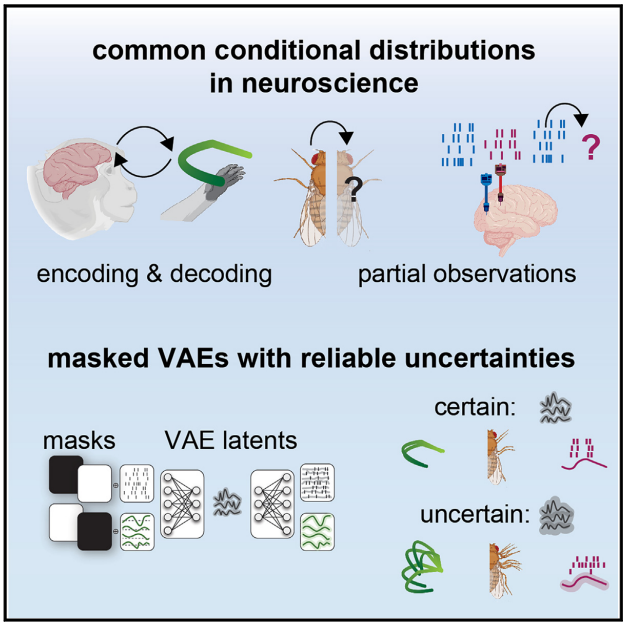
replay conditions, we selected only parts of the first and second VR condition (‘VR constrained) and only the first replay condition (‘Rep. constrained). **E.** Neural activity also improves classification performance when including data along the entire track. Test classification accuracy when considering data of the entire platform (100 cm) of SCs and SCim sessions, considering all (constrained) covariates, or neural activity and these covariates. The same hue marks sessions from the same animal. Chance level is indicated by grey horizontal line. **F.** Same as E, but for the first 75 cm of the platform, excluding the area closest to the object, where the visual stimulation through the sphere was most pronounced. **G.** Constraining data to epochs with minimal eye movement. When excluding eye movements, neural activity in both SCs and SCim still contributed to prediction performance beyond covariates. Conventions as in Figure 4.

Resource

Cell Reports

Modeling conditional distributions of neural and behavioral data with masked variational autoencoders

Graphical abstract



Authors

Auguste Schulz, Julius Vetter, Richard Gao, ..., Pavan Ramdya, Pedro J. Gonçalves, Jakob H. Macke

Correspondence

auguste.schulz@uni-tuebingen.de (A.S.), jakob.macke@uni-tuebingen.de (J.H.M.)

In brief

Schulz et al. demonstrate how neural encoding and decoding can be cast as computing conditional distributions and how to modify variational autoencoders (VAEs) to calculate such distributions. The proposed VAE-masking scheme allows for joint dimensionality reduction of neural and behavioral data and conditional generation of one modality given the other.

Highlights

- We can recast neural encoding and decoding as computing conditional distributions
- We modify VAEs so that they can flexibly compute such conditional distributions
- Our method can be applied to various datasets, tasks, conditions, and levels of missingness
- Our approach results in more reliable uncertainty estimates than standard VAEs

Schulz et al., 2025, Cell Reports *44*, 115338
March 25, 2025
https://doi.org/10.1016/j.celrep.2025.115338

CellPress

Cell Reports

CellPress
OPEN ACCESS

Resource

Modeling conditional distributions of neural and behavioral data with masked variational autoencoders

Auguste Schulz,[1,9,*] Julius Vetter,[1] Richard Gao,[1] Daniel Morales,[2] Victor Lobato-Rios,[2] Pavan Ramdya,[2] Pedro J. Gonçalves,[1,3,4,5,6,8] and Jakob H. Macke[1,7,8,*]

[1]Machine Learning in Science, University of Tübingen & Tübingen AI Center, Tübingen, Germany
[2]Neuroengineering Laboratory, Brain Mind Institute & Interfaculty Institute of Bioengineering, EPFL, Lausanne, Switzerland
[3]VIB-Neuroelectronics Research Flanders (NERF), Leuven, Belgium
[4]Imec, Leuven, Belgium
[5]Department of Computer Science, KU Leuven, Leuven, Belgium
[6]Department of Electrical Engineering, KU Leuven, Leuven, Belgium
[7]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[8]These authors contributed equally
[9]Lead contact
*Correspondence: auguste.schulz@uni-tuebingen.de (A.S.), jakob.macke@uni-tuebingen.de (J.H.M.)
https://doi.org/10.1016/j.celrep.2025.115338

SUMMARY

Extracting the relationship between high-dimensional neural recordings and complex behavior is a ubiquitous problem in neuroscience. Encoding and decoding models target the conditional distribution of neural activity given behavior and vice versa, while dimensionality reduction techniques extract low-dimensional representations thereof. Variational autoencoders (VAEs) are flexible tools for inferring such low-dimensional embeddings but struggle to accurately model arbitrary conditional distributions such as those arising in neural encoding and decoding, let alone simultaneously. Here, we present a VAE-based approach for calculating such conditional distributions. We first validate our approach on a task with known ground truth. Next, we retrieve conditional distributions over masked body parts of walking flies. Finally, we decode motor trajectories from neural activity in a monkey-reach task and query the same VAE for the encoding distribution. Our approach unifies dimensionality reduction and learning conditional distributions, allowing the scaling of common analyses in neuroscience to today's high-dimensional multi-modal datasets.

INTRODUCTION

Recent developments in experimental techniques allow real-time behavioral tracking of animals[1–3] and simultaneous recordings of hundreds of neurons across multiple brain regions.[4–6] Modern datasets in neuroscience are thus increasingly large, high-dimensional,[7,8] and commonly consist of multiple modalities—e.g., neural activity and behavior[9]—that often have highly non-linear relationships.[10] While data collection has changed drastically in the last years, an important goal of systems neuroscience remains the same: understanding how brain activity gives rise to complex behavior.

To gain insights from neural and behavioral data, neuroscientists have developed various neural encoding and decoding models.[11] These tasks should ideally be addressed in a probabilistic manner to account for the inherent variability of neural and behavioral measurements and in order to quantify resulting uncertainty. As experimental neuroscience is moving toward less controlled, unconstrained, multi-modal data collection, this aspect becomes even more relevant. Both probabilistic encoding and decoding tasks can, algorithmically, be boiled down to the task of calculating conditional distributions: encoding studies in neuroscience involve calculating the conditional distribution of neural activity given behavior or other observations such as stimuli.[12] Conversely, for decoding analyses, one needs to calculate the conditional distribution over behavior, given neural activity (Figure 1A, left). Generating interpretable, accessible neuroscientific predictions from complex, high-dimensional data directly is very challenging,[13,14] highlighting the need for tools that can infer low-dimensional representations of high-dimensional neural and behavioral datasets (Figure 1A, right). In short, to gain neuroscientific insights from such complex datasets, our goal is to unify (1) the ability to link neural and behavioral data (i.e., through encoding/decoding models; Figure 1A, left), and (2) joint dimensionality reduction of the data, ideally in a probabilistic and generative manner (Figure 1A, right).

Various dimensionality reduction methods have demonstrated that a substantial fraction of variability both in unconstrained behavior and neural population activity can be captured by a few latent (i.e., unobserved) dimensions.[15–21] This insight has driven the development of various latent variable models

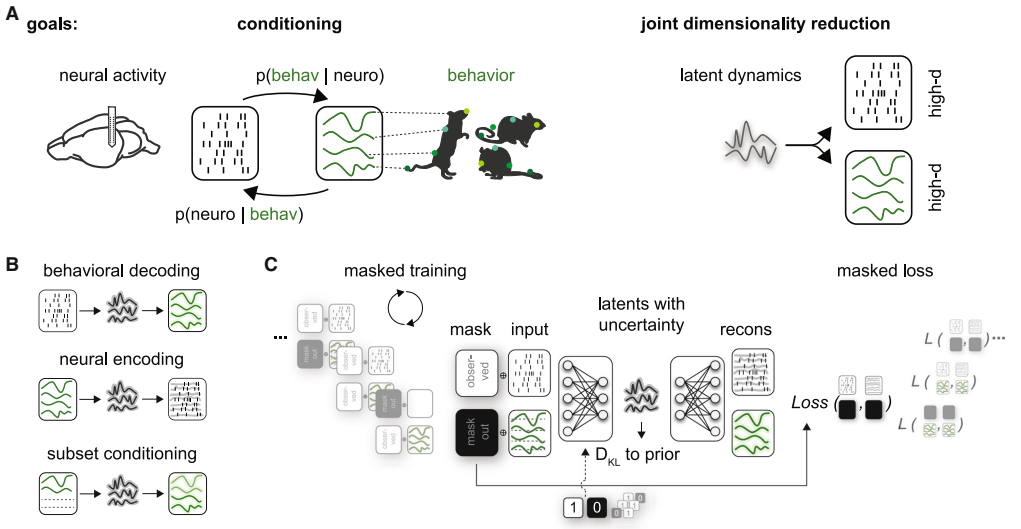


Figure 1. Latent variable models that can deal with conditional distributions arising in large-scale multi-modal datasets

(A) Our approach aims to address two common goals of (neuro)scientific analyses: conditioning, e.g., for linking brain activity (neuro) and behavior (behav) and joint dimensionality reduction of potentially high-dimensional neural and behavioral data.

(B) Conditional distributions arise, e.g., when learning the distribution over behavior given brain activity (behavioral decoding), the distribution over neural activity given behavior (neural encoding), or when analyzing the interaction of data subsets such as different behavioral variables or multiple brain regions.

(C) Masked variational autoencoder training scheme for data of potentially different data types (e.g., count and continuous) with structured masks for modeling conditional distributions. During each training iteration (for which a subset of the training data is passed to the network), one mask is chosen randomly, and the corresponding inputs to the network are replaced by a fixed imputation value (zero or the mean). The reconstruction loss is computed solely on observed data, i.e., for different masks, the reconstruction loss is computed on a distinct subset of the data. Identical to classical VAEs, the training objective combines the (masked) reconstruction loss and a regularization term for the latent space: the Kullback-Leibler divergence D_{KL} of the inferred latent representation and a defined prior (here, standard Gaussian). Optionally, one can pass the binary mask (1, observed; 0, masked) to the encoder network (dashed line, see STAR Methods).

that infer underlying low-dimensional representations from neural data.[16,22–26] Classical methods often require simplifying modeling assumptions, such as (switching) linear dynamics,[27–29] or strictly Gaussian observations,[15] and rely on model-specific optimization schemes (e.g., expectation-maximization algorithms or subspace-identification methods).[19,30] With the rise of deep learning and more flexible optimization schemes, various of these assumptions can be relaxed,[16,20,21] leading to latent variable models that can capture complicated non-linear relationships in the data and underlying low-dimensional dynamics.[31]

One such model class, exploiting deep inference networks, is the variational autoencoder (VAE).[32,33] Inference networks of VAEs take observed data as the input and return a distribution over the latent state. VAEs are, however, often primarily used as tools for dimensionality reduction, where data are compressed and then decompressed. However, VAEs provide a full, probabilistic, non-linear latent variable model that can approximate the whole underlying latent distributions rather than providing only a compressed point estimate. Sequential variants of VAEs can infer latent representations underlying heterogeneous time-series datasets, e.g., consisting of both continuous and count data (spiking),[31,34,35] and are commonly used in analyzing neural and behavioral data.[16,20,23,25,36] However, most VAE-based methods cannot adequately deal with a ubiquitous analysis task in neuroscience: calculating arbitrary conditional distributions p(data subset A|data subset B).[37–39] The reason for this is that inference networks of VAEs typically can only deal with *fully* observed input data, and have no means to model the (additional) uncertainty arising from partial observations. Such conditional distributions do not only arise when estimating behavioral decoding (Figure 1B, top) and neural encoding distributions (Figure 1B, middle) but also become relevant when dealing with partially observed data or when studying interactions between brain regions or tracked body parts (Figure 1B, bottom). An ideal model should correctly estimate how uncertain it is about the inferred underlying representation and its predictions (error bands, Figure 1B). Accurate uncertainty estimates can tell us how constrained one subset is given the other subset and let us reason about their dependencies beyond accuracy or similarity scores. However, in neuroscience, the quality of uncertainty estimates is typically not assessed.

In this work, we present an approach that enables VAEs to accurately model arbitrary data conditionals arising in neuroscience. Specifically, we use a masked-training approach and demonstrate it in a variety of neuroscience applications where we achieve both dimensionality reduction and sampling

of conditional distributions. Furthermore, we propose calibration tests to assess the quality of the generated conditional distributions.

VAEs have been extended to model the distribution of a missing data subset given an observed data subset.[34,39] Conceptually, we build on these approaches and treat the modality we want to learn the conditional distribution over as missing. For example, to capture a behavioral decoding distribution, we modify the loss and training scheme of a VAE during joint training on neural and behavioral data by stochastically masking behavior. Our training approach is not limited to a specific modeling architecture but can be applied to a variety of VAE approaches. We showcase our approach on diverse datasets: sequential and static, multi- and uni-modality, discrete and continuous datasets, each of which has a different VAE architecture. We first validate our approach on a task on which we have access to the ground-truth distributions. We demonstrate that our approach allows for correct inference of low-dimensional latents and accurate predictions. On a high-dimensional behavioral dataset of walking flies, we successfully recover the relationship between different body parts along with uncertainties and obtain realistic samples from the conditional distributions of masked legs. Finally, we showcase the approach in a challenging multi-modal neural and behavioral dataset, where we model encoding and decoding distributions of high-dimensional population activity from primary motor areas and self-paced reach movements.[40] Neural latents extracted from partial neural recordings reveal that our masked VAE approach has the desired property of increasing uncertainty when predictions are likely wrong.

RESULTS

Masked training of variational autoencoders for estimating conditional distributions

To prepare variational autoencoders to deal with conditional distributions commonly arising in neuroscience, we modify the training scheme of classical VAEs. To reiterate, VAEs can approximate the whole underlying latent distribution, often parametrized by a mean and variance, rather than providing point estimates. This aspect is critical for handling partial or masked observations and corresponding uncertainty levels in the latent distribution. During joint training on multiple data subsets, our approach prepares the network for each subset to be structurally masked at test time (Figure 1C). We use the term structured masking to refer to algorithmic masking of data subsets for conditioning to avoid confusion with actual data missingness—e.g., individual input channels that drop in and out. First, we specify the structured masks depending on the desired conditional distributions and specify how often (on average) each mask should be selected during training (Figure 1C, left; see STAR Methods). During each training iteration, a random subset of the training data, often referred to as a mini-batch, is passed to the network, and one mask is chosen randomly. The corresponding masked inputs are replaced by a fixed imputation value (e.g., zero or the mean). We then calculate the reconstruction loss $\mathcal{L}$ solely considering observed subsets (see STAR Methods) in the evidence lower bound (ELBO), which is the optimization target of VAEs.[33,39] The ELBO combines such reconstruction terms with a regularization term in latent space. Here, we compute the Kullback-Leibler divergence D_{KL} between the inferred latent representation and a standard Gaussian prior over the latents. The training objective of masked VAEs, thus, only differs from classical VAEs in terms of the masked reconstruction loss. Passing the mask to the reconstruction loss is the crucial component that instructs masked VAEs to update their weights to appropriately deal with different conditional distributions. Additionally, to make it easier for the network to learn that an input has been masked, one can pass the binary mask to the encoder network, in addition to the masked information in the loss. In cases where dimensions masked during training would, at test time, happen to take the same value as the imputation value (e.g., silent neural populations and zero imputation), the binary mask should always be passed. This allows the VAE to properly disentangle whether these dimensions are masked or observed.

In summary, we propose modeling conditional distributions with VAEs, e.g., for neural encoding and behavioral decoding, by recasting it as a structured masking problem. This approach allows us to sample from a distribution of interest, e.g., to visualize time series of various potential behaviors that are likely given a neural population activity trace and vice versa. As such, our results demonstrate, in various application scenarios, how to perform latent inference and generate samples of data modalities that are unobserved at test time. The generality of this approach allows for applying it to a variety of conditional distributions and variational autoencoder settings.

Inference of conditionals in a tractable Gaussian latent variable model

First, we evaluated whether our training scheme and loss modification allow us to learn the correct distributions of interest on a simulated dataset where we have access to the ground-truth conditional and posterior distributions. This dataset was generated from a Gaussian latent variable model (GLVM) with latent (unobserved) random variable z and data dimensions x, which linearly depend on the latent z (Figure 2A; see STAR Methods). In this illustrative example, we can think of a subset of x as the high-dimensional neural activity, another subset of x as high-dimensional behavior, and the latent variable z as the low-dimensional representation underlying both neural and behavioral data. The inference network infers the distribution over these unobserved latents given a chosen x—i.e., it calculates the posterior distribution $p(z|\text{observed x})$—effectively inverting the data-generation process in a probabilistic way. The strength of the linear coupling and the noise levels of individual x dimensions define how much information about the latent can be gained by observing those x dimensions.

We contrast the masked VAE with a regular VAE trained on all data (referred to as naive training) regarding the capacity to capture data conditionals $p(\text{masked x}|\text{observed x})$ at test time (Figure 2). The naive approach fails to capture the true data distribution (gray), with overly narrow 1D and 2D (marginal) distributions (Figure 2B, left). Masked VAEs, however, can successfully reconstruct observed values (x_i^{obs}) and impute masked ones (x_i^{unobs}) (Figure 2B, right). The reason for this discrepancy lies in the ability to learn the distribution over the latents (posterior distribution) when some of the input data are unobserved.

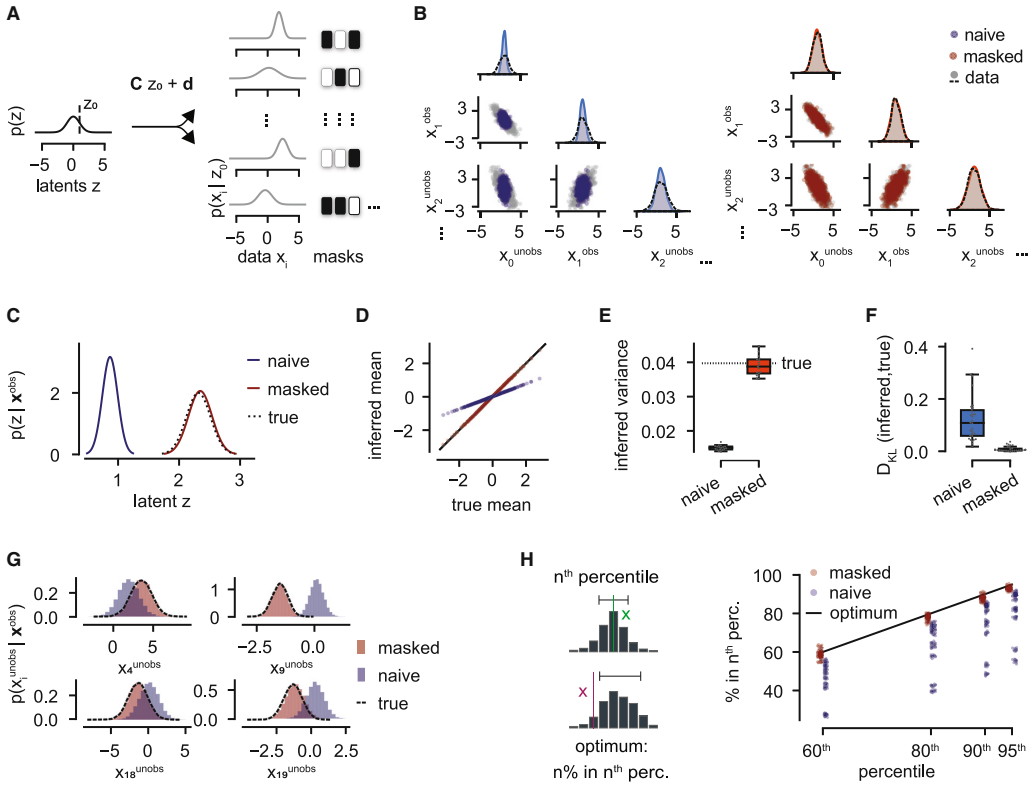


Figure 2. Inference of conditionals in a GLVM

(A) Left: schematic of a GLVM with fixed parameters $\theta = \{C, d, \Lambda\}$, where $\Lambda = \mathrm{diag}(\sigma_i^2)$ for $i \in \{1,20\}$. Right: masks for conditioning specified by the user, where each column is a different mask. Here, 50% of the values are masked.

(B–H) Comparison of our masked-training approach (red) with a vanilla VAE approach (blue, naive) when half of the inputs x are masked out at test time.

(B) Model reconstructions and true test data samples (1D and 2D marginal distributions over x).

(C) Inferred and analytical (true) posterior distributions over latent z given only the observed dimension of one test sample, i.e., $p(z|\mathbf{x}^{\mathrm{obs}})$.

(D) True versus inferred posterior mean given a range of test samples.

(E) Inferred posterior variance across multiple instantiations (seeds) and true posterior variance (dotted line). Data are represented as mean over test samples. Boxplots show the median and lower and upper quartiles.

(F) Average Kullback-Leibler divergence between true and inferred posterior distributions across different GLVM parameters ($\theta = \{C, d, \Lambda\}$) and structured masks. Data representation as in (E).

(G) Conditional distributions over randomly chosen masked x dimensions (see STAR Methods) for the same test sample as in (C).

(H) Left: schematic of statistical calibration, evaluating the quality of uncertainty estimates. Optimal calibration: n% of true data points lie in the n^{th} percentile confidence interval of the sampling distribution. Example of an x within the interval (green, top) and one outside of it (red, bottom). Right: calibration checks of predicted conditional distributions $p(x_i^{\mathrm{unobs}}|\mathbf{x}^{\mathrm{obs}})$ for all masked x dimensions across multiple model seeds. See also Figures S1-S4.

Masked training can perfectly infer the true analytically calculated posterior. In contrast, naive training fails to do so (Figure 2C). The naive network does not detect masked values as such. Hence, its posterior mean inference is biased (Figure 2D), and the posterior variance is too small (Figure 2E). These observed discrepancies between masked and naive VAEs are even more pronounced when more than 50% of the values are masked. Conversely, when only one of the dimensions was masked, the discrepancies were less pronounced (Figure S1).

In short, naive VAE training leads to confidently wrong predictions, while the masked network correctly adjusts the uncertainty about its predictions. This finding generalizes across different parameter sets (C, d, Λ) and masking conditions, and observation noise ranges (Figures 2F, S2, S3, and S4). The higher the overall observation noise, however, the more training samples are required to achieve a good model fit (Figure S3).

One of the advantages of VAEs is that, once trained, sampling from VAEs is straightforward. Thus, we can easily investigate

CellPress
OPEN ACCESS

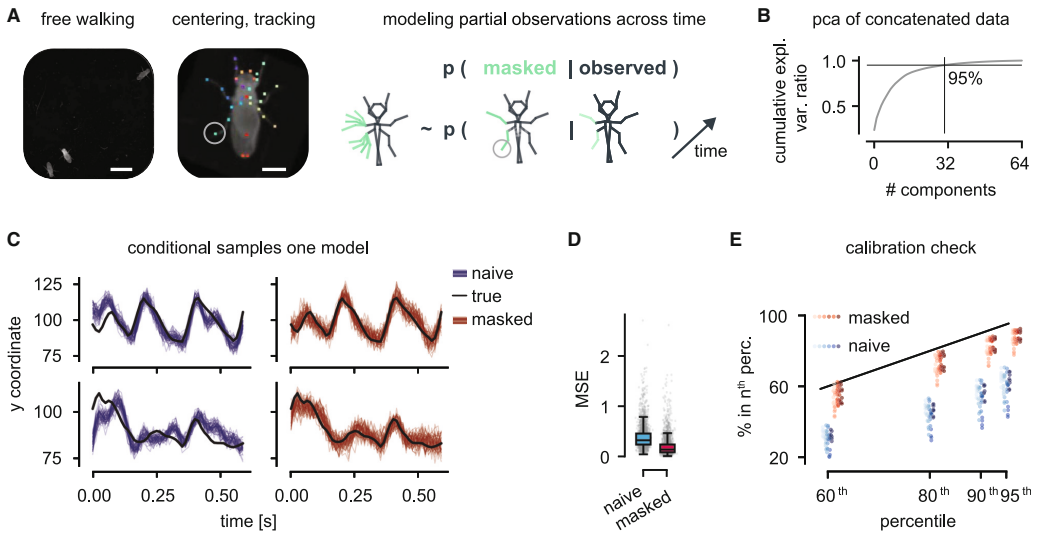


Figure 3. Conditional sampling of masked legs of walking flies

(A) Three flies are filmed from the bottom during walking behavior in a constrained arena. The scale bar represents 5 mm. Cropped video frames of individual flies are centered, aligned for constant head direction, and tracked with DeepLabCut, resulting in a 64-dimensional time series. The scale bar represents 1 mm. Schematic of the target distribution: the conditional distribution over masked left legs given the remaining body key points.

(B) Cumulative explained variance of the number of components when performing principal-component analysis on all concatenated time-series data.

(C) Two example time series (from the test set) of the unobserved limb, marked with a circle in (A). Conditional samples from the masked model in red, naive in blue, and true limb trajectory in black.

(D) Mean squared error (MSE) of mean predictions for the masked limb, averaged across time and test samples, for both training schemes. Boxplots show the median and lower and upper quartiles.

(E) Calibration checks of predicted conditional distributions $p(\text{limb}_i^{\text{unobs}} | \text{limbs}^{\text{obs}})$ for all masked leg key points shown in (A), across multiple model seeds (different hue per seed). Optimal calibration in black.

whether the conditional samples for unseen test data correspond to the conditional distribution we set out to model. Wrongly inferred posterior distributions will likely result in inaccurate conditional samples.

Indeed, for four example masked data dimensions (x_i^{unobs}), the distribution of conditional samples from the naive VAE is wrong, whereas the masked training distribution matches the true (analytical) conditional (Figure 2G).

In real neuroscientific datasets, we do not have access to the true conditional distributions for comparison. For such cases, we propose to evaluate the quality of the inference method and its uncertainty estimates with an adaptation of simulation-based calibration.[41,42] These calibration checks allow us to also evaluate the quality of the uncertainty estimates and thus go beyond the evaluation of mean squared error or log-likelihood of structurally masked values, which informs only about the quality of the mean predictions. More concretely, calibration checks count how often masked test sample values x_i^{unobs} lie in the respective predicted conditional sampling distribution (Figure 2H; see STAR Methods). We found that, for the masked training, predictions are well calibrated, i.e., neither overconfident nor underconfident (values close to the diagonal Figure 2H, red). In contrast, the naive approach is confidently wrong for many test cases (Figure 2H, blue). Overall, these results suggest that masked training allows us to infer both the correct posterior $p(z|\text{only observed x})$ and conditional distributions $p(\text{masked x}|\text{only observed x})$ in a tractable, well-specified example. The results demonstrate the statistical challenges that arise when one aims to use VAEs to perform conditioning and provide a theoretical basis for applying the masked-training approach to neuroscientific data.

Probabilistic conditional modeling of masked key point trajectories in fly walking behavior

Next, we wanted to investigate whether the masked approach is applicable to complex time-series data in neuroscience and can successfully model conditional distributions of scientific interest. In particular, we focused on an experiment that characterizes the (backward) walking behavior of the fruit fly, *Drosophila melanogaster*, and applied our masked-training approach to a sequential VAE developed for this high-dimensional behavioral dataset.

We obtained the dataset by tracking the centroids of individual flies and aligning the video frames such that fly heads are all pointing upward (Figure 3A). We tracked 32 body parts (x, y each) with DeepLabCut,[1] resulting in a 64-dimensional time series (see STAR Methods). To account for the temporal structure

of the data, the VAE's architecture has both convolutional elements and recurrent neural networks based on gated recurrent units,[43] as well as elements for non-linear dimensionality reduction (see STAR Methods). Analogous to the GLVM case, we adapted the masked-training scheme for this sequential VAE to allow for modeling the conditional distribution over a subset of the fly body key points, given the remaining ones (Figure 3A, right, masked legs in green; see STAR Methods). Here, we chose to mask body key points that are crucial for walking behaviors and show characteristic variability during walking: hind claw, hind tibia-tarsal joint, mid tibia tarsus, and mid claw of the left side.

If the model captures the dependence correctly via the compact latent representation, we expect accurate conditional modeling of these masked legs. In the previous example, the variability in the data was captured by one latent variable, but, for experimental data, the underlying dimensionality is unknown. Here, we are dealing with a dataset with high intrinsic dimensionality: almost 32 principal components are required to capture 95% of the variance of the 64-dimensional dataset (Figure 3B). In our sequential VAE, we can exploit temporal dependencies and thus further reduce the dimensionality of the latent space while capturing stereotyped walking behavior (see STAR Methods). Samples from the naive model capture overall trends of the masked left hind claw (circle in Figure 3A) well, particularly during highly periodic walking (Figure 3C, left). However, ground-truth trajectories often deviate from the sampled trajectories (blue). In contrast, masked training produces more faithful predictions and uncertainty estimates (red), reflected in the inclusion of the ground truth for most time steps (Figure 3C, right). This is also captured by a lower average mean-squared error (MSE) of the masked VAE test predictions (averaged across time) of the masked key point shown in Figure 3C. However, MSE alone does not immediately reveal a substantial performance boost through masked training (Figure 3D). The difference, however, becomes clear when inspecting the uncertainty estimates: when the naive approach is wrong, it is confidently wrong (large deviations from the diagonal in Figure 3E), while the masked approach is better calibrated.

We conclude that our masked-training methodology is indeed applicable to time-series datasets and allows us to faithfully model the conditional distributions of masked body key points given the remaining ones. Masked training leads to better uncertainty estimates—it allows the network to know better when it does not know.

Decoding continuous reaches from neural population activity

To be effective for neuroscientific research, our method should be able to deal with data types and modalities that commonly arise in neuroscience. Therefore, we implemented the masked-training scheme for a classic monkey-reach task, which is particularly challenging due to its continuous instead of trial-based structure (using publicly available data from O'Doherty et al.,[40]; Figure 4A, left). We focus on the behavioral decoding distributions, i.e., the conditional distribution of x- and y-reach directions given activity traces of >200 neurons (Figure 4A, right).

The monkey reaches toward an indicated light target on an 8 × 8 grid, leading to movements of different lengths, heterogeneous angles, and velocities (Figure 4B). Neural activity is simultaneously recorded in primary motor cortex. The maximum spike count of individual units is six spikes in time windows of 64 ms (Figure 4C; binning consistent with Makin et al.[44] to capture behaviorally relevant timescales). We built a sequential VAE for this multi-modal dataset, which consists of both continuous (behavioral) and discrete data (spike counts). Our reconstruction loss, therefore, is composed of a Poisson- (for spike counts) and Gaussian- (for behavior) negative log-likelihood (GNLL) loss (see STAR Methods[34,35]). We specified masks for encoding and decoding distributions during training; i.e., we masked either neural activity (replaced by zeros) or behavior (replaced by the mean cursor position). Masked mean reconstructions (red) of behavioral traces are more accurate than naive (blue) predictions across many model seeds (Figure 4D). Surprisingly, the naive approach is performing relatively well, and, while we see some sections where the masked approach is performing better, the errors are quite consistent across the two approaches. This indicates that some sections of the traces are less correlated with neural activity than others, and both models are capable of exploiting some correlations required for conditional modeling. Sampling from individual masked and naive models (Figure 4E) and calibration checks (Figure 4F) again demonstrate that the masked but not naive approach targets the conditional—the decoding distribution. Note that a discrepancy between x (bottom rows) and y (top) decoding performance (Figures 4D–4F) has been reported previously for this dataset.[40,44,45] While the masked VAE is not perfectly calibrated on this dataset, it clearly outperforms the naive VAE.

In conclusion, our masked-training approach can be readily applied to multi-modality datasets and makes it possible to sample from a conditional distribution over a continuous time series given high-dimensional time series of discrete count data.

Encoding of continuous reaches in neural population activity

Next, we assessed the performance of the same trained model on the reversed and more challenging task: modeling the high-dimensional conditional distribution over the activities of 213 neurons in primary motor cortex given only the two-dimensional behavioral trajectories (Figure 5A).

Both masked and naive approaches generally capture the histogram of observed spike counts given the reach trajectories (Figure 5B)—despite a slight over-prediction of spike counts—but the naive approach has worse population-average estimates across model instantiations (Figure 5C). The log-likelihood per neuron across model instantiations reveals the superior performance of the masked versus naive encoding (Figures 5D and 5E).

Samples from the trained models suggest that both masked and naive approaches correctly predict time-varying firing rates that clearly reflect the reach movements (Figure 5F). Notably, the masked approach reveals higher variability in the rate predictions reflecting higher posterior uncertainty.

Spike counts are discrete rather than continuous variables, so we adapted the method to assess the uncertainty calibration: we compare the cumulative distribution function (CDF) of the ground-truth spike train against a Poisson spike train with a

CellPress
OPEN ACCESS

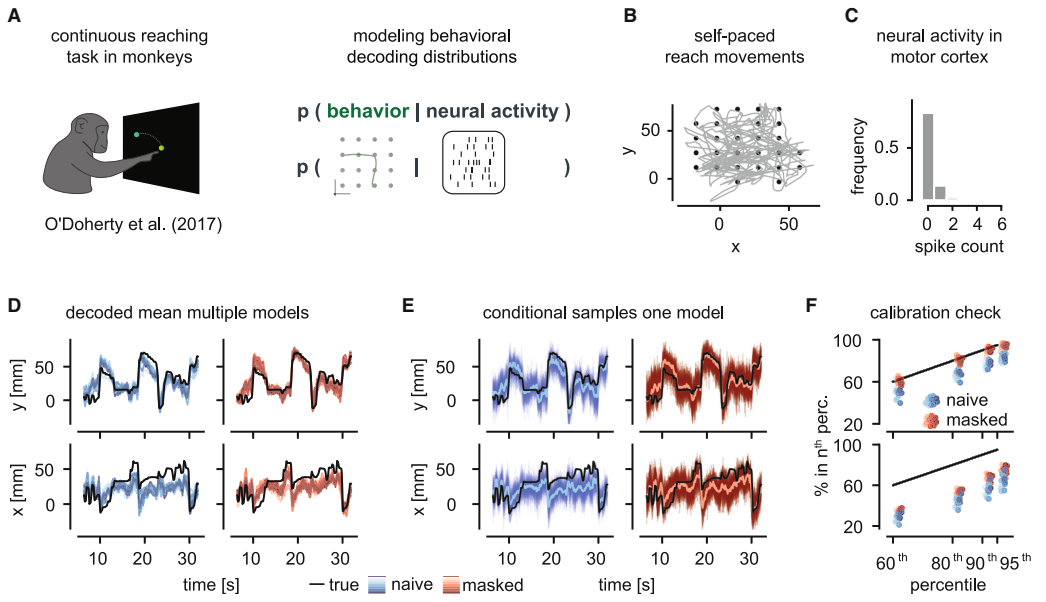


Figure 4. Behavioral decoding of continuous reach movements from monkey primary motor cortex

(A) A monkey performs self-paced, continuous reaches on an 8 × 8 grid with simultaneous cortical electrophysiology recordings. Schematic of the target distribution: the conditional distribution over behavior, in this case, cursor trajectories given neural population activity recorded in monkey primary motor cortex.

(B) Behavioral trace of continuous reach movements (gray) and targets (black).

(C) Frequency of observed spike counts across primary motor cortex during the reach movements shown in (B), binned at 64 ms.

(D) Example traces of cursor positions and the mean predictions for the naive (blue) and masked (red) modeling approaches for multiple model seeds (different hue per seed).

(E) Same cursor traces as in (D) but with conditional samples from a single naive (blue) and masked (red) model, respectively.

(F) Calibration checks of predicted conditional distributions $p(\text{behavior}|\text{neural activity})$ respectively for x-y directions, across multiple model seeds. Optimal calibration in black.

(D–F) The y direction is in the top and the x direction in the bottom row.

rate sampled from either the masked or the naive models (Figure 5G). We find that the sampling distributions for both masked and naive capture the ground-truth distribution reasonably well (Figure 5G; Figures S5-S8 for other units).

In conclusion, our masked VAE approach allows us to model and produce samples from both decoding and encoding distributions in one single model without requiring any retraining.

Decoding from the latent space of partial neural recordings

Finally, we explored the relationship between latent uncertainty and downstream task performance, specifically decoding movements from latents extracted from partial neural observations (Figure 6A). We posit that, when performing downstream decoding from such latents, an important advantage of uncertainty estimates is that they can indicate when not to trust a decoded movement, namely when the corresponding uncertainty is too high.

In order to test this idea, we trained a VAE on neural activity alone. First, we specified different masking levels, ranging from five masked neurons out of 213 to 200 masked neurons. As before, we sampled the masks during masked training and passed masked spikes as zeros (Figure 6B). While masked VAEs increase the latent uncertainty of their most informative latents (highest mean variability across time; see STAR Methods) with increasing mask size, naive VAEs fail to do so (Figure 6C). This holds across different masking levels and throughout the test set (Figure 6D).

Second, we used linear ridge-regression models to decode movement velocity from either the latent means of the masked or naive VAEs, or directly from spikes (Figure 6E). We found that decoding performance (measured as the correlation between the true and predicted velocity) from raw spikes was lower than from inferred latents. This result is in agreement with previous literature,[44,46] further highlighting the advantage of VAE-based latent variable models.

Finally, we assessed how decoding performance is correlated with latent uncertainty. In order to accomplish that, we averaged the mean latent uncertainty over time for different masking levels for the masked and naive VAEs. We then fitted per VAE instantiation (seed) a linear regression from decoding performance to latent uncertainty across masks. The linear fits reveal a negative

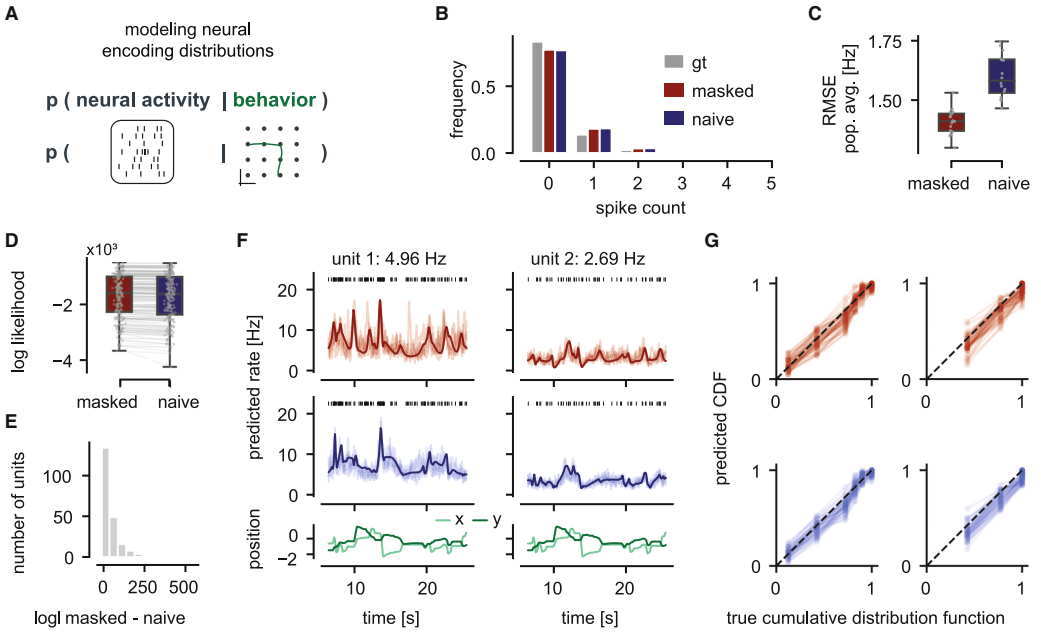


Figure 5. Neural encoding distributions given continuous movements in the monkey-reach task

(A) Schematic of the target distribution: the conditional distribution over neural population activity recorded in monkey primary motor cortex given behavior, i.e., cursor trajectories.

(B) Frequency of observed spike counts across primary motor cortex during reach movements binned at 64 ms and the predicted spike-count distribution of the masked (red) and naive (blue) approach.

(C) Root-mean-squared error (RMSE) of the population average and predicted population average for different model seeds. Data are represented as mean across time points. Boxplots show the median and lower and upper quartiles.

(D) Log-likelihood per neuron for the masked and naive approaches. Data are represented as mean across time points and model seeds. Higher is better. Boxplots as in (C).

(E) Distribution of differences in log-likelihood (logl) of the masked minus the naive approach. Positive values indicate a better model fit of the masked approach.

(F) Sampled rate predictions (10 each) and mean rate prediction from both models (masked red top, naive blue middle) for two example neurons with different activity levels given the standardized behavioral trajectory (bottom).

(G) CDF of the observed spikes (true) vs. predicted spike distributions sampled from the masked (red) and naive (blue) VAE for the rate predictions shown in (F). To calculate the CDF, spike counts are aggregated across five bins due to low spike counts. Optimal predictions would lie on the diagonal (black dotted line). See also Figures S5–S8.

correlation for masked, but not naive, VAEs, between uncertainty prediction and decoding performance (Figure 6E): for the masked case, 10 out of 10 model instantiations reveal a significant ($p < 0.005$) negative slope, with values ranging from -2.8 to -1.5 (R-squared between 0.84 and 0.98). For the naive case, we obtain slopes ranging from 0.04 to 0.22 (R-squared between 0.11 and 0.87), where many (six out of 10) are not significantly different from 0 ($p > 0.05$).

In conclusion, we find that masked but not naive VAEs have the desired property of increasing uncertainty when the predictions are likely wrong (low decoding performance).

DISCUSSION

We introduce a training methodology for modeling conditional distributions with masked variational autoencoders, bridging dimensionality reduction, generative modeling, and encoding and decoding analyses in neuroscience. Our experiments show that modifying the training scheme and loss through structured masking enables VAEs to model specified conditional distributions. Thus, our approach allows for joint dimensionality reduction of high-dimensional multi-modal data and conditioning on specified modalities. It is not restricted to specific architectural or modeling choices and can be easily applied to a variety of variational autoencoders deployed in neuroscience. We validated our approach on a tractable example in which we correctly learned the ground-truth posterior and conditional distributions. We applied our approach to two neuroscientific time-series datasets: a continuous reach task in monkeys,[40] in which we probabilistically encoded behavior in—and decoded behavior from—high-dimensional neural activity, as well as a behavioral dataset of walking flies, for which we successfully modeled the

CellPress
OPEN ACCESS

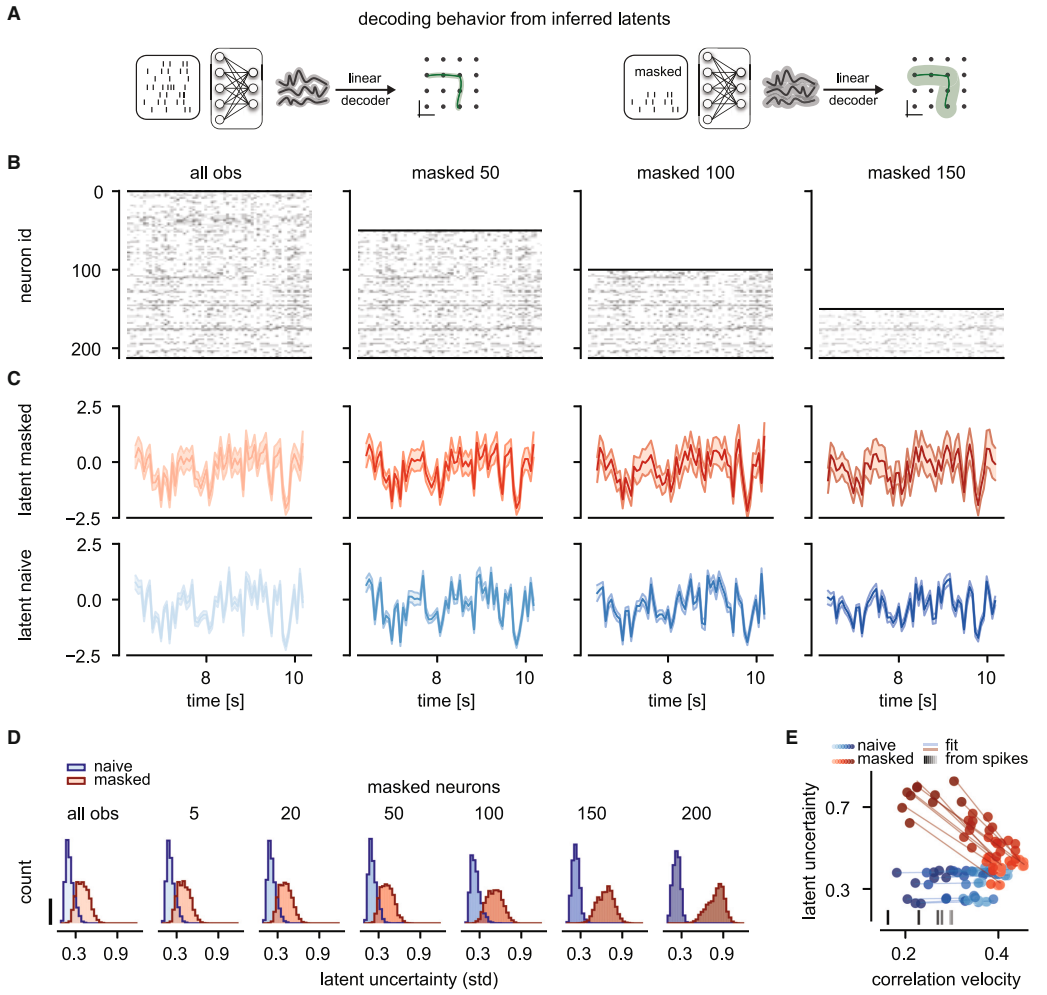


Figure 6. Impact of partial neural observations on latent uncertainty and downstream decoding

(A) Decoding from uncertainty-aware latents: latent uncertainty should increase for partial (right) vs. full (left) observations, which would translate to more variable decoded behavior.

(B) Spiking data for fully observed and masked conditions with three masking levels. Masked spikes are passed as zero values.

(C) Latent state over time represented as mean ± standard deviation (shaded region) returned by the VAEs for masked (red) and naive (blue) VAEs.

(D) Distribution of latent uncertainty per time step at varying masking levels (all observed, and masking of 5, 20, 50, 100, 150, or 200 neurons). The scale bar represents 300 counts.

(E) Linear decoding performance (ridge regression, correlation of y velocities) from either latents of masked VAEs (red) or naive VAEs (blue) plotted against the mean uncertainty (i.e., returned standard deviation) of the most informative latents. Shown for different masking levels (light, all observed, to dark, 200 neurons masked), depicted in (D). Linear fits per VAE-model instantiation (seed) reveal a negative correlation between uncertainty prediction and decoding performance for the masked (slopes: -2.8 to -1.5; R-squared: 0.84 to 0.98; 10/10 seeds $p<0.005$), but not the naive (slopes: 0.04 to 0.22; R-squared: 0.11 to 0.87; 6/10 seeds $p>0.05$) , approach. In gray, decoding performance directly from spikes. Note that decoding from spikes has no associated latent uncertainty.

conditional distributions of masked body parts. Across different levels of missingness of neural data during the reaching task, masked VAEs revealed higher downstream decoding performance than naive VAEs. In addition, both masked and naive VAEs showed higher decoding performance from inferred latents than a linear decoder from neural population activity directly, consistent with previous studies.[44,46] Furthermore, we showed how to assess the models' uncertainty estimates, a crucial but often neglected aspect in deep-learning-based dimensionality reduction, and verified that our models learn calibrated distributions. In particular, we demonstrated on the monkey-reach dataset that our approach has the desired property of increasing latent uncertainty when predictions are likely wrong. In order to validate this method, we restricted the analyses to examples, where, in principle, we would have access to all modalities both during training and testing. This, however, is only a requirement for validating the method, not for future applications.

Generality of modeling conditional distributions in neuroscience and beyond

A key contribution of this work lies in linking conditional distributions of neural and behavioral encoding and decoding to probabilistic approaches for dealing with missing data.[34,39] The generality of this approach opens up various possibilities beyond encoding and decoding, as generating conditional samples and performing inference from partial observations have many applications. In experiments such as the fly-walking example where occlusions or tracking issues occur, our approach enables sampling from distributions over the obscured body key points. In addition, the inherent denoising property of VAEs can correct noisy markers, especially when the observation noise is learned explicitly. More generally, data imputation can be framed as modeling conditional distributions over missing variables given observed ones. This is a relevant pre-processing step for many downstream analyses that require complete datasets in neuroscientific and clinical applications, as well as in other domains.[47,48] For example, if an electrode breaks during a neural recording, a masked VAE approach can salvage the dataset by computing conditionals for the failed electrode using complete data from other sessions.

Using and assessing uncertainties in deep-learning-based models

It is well established that both modern deep-learning models[49] and traditional Bayesian decoders[50] can make overconfident predictions. In contrast, trustworthy, well-calibrated models should exhibit high uncertainty when predictions are likely to be inaccurate and low uncertainty when they are likely to be accurate. In neuroscientific applications, the ability to assess uncertainty can be particularly important, for example in tasks where wrong predictions can have serious consequences, such as brain-computer interfaces and real-time decoding for an actuator. Classical VAEs provide access to the predicted uncertainty over the inferred latent states, often in the form of the variance of a Gaussian approximate posterior distribution.[32,33] However, this aspect is sometimes treated as a convenience for robust training rather than as a meaningful quantity with respect to the system under investigation. In this work, we have demonstrated the effect of increased latent uncertainty when dealing with partially observed inputs, highlighting the need for modeling of latent uncertainty. Note, however, that correctly capturing uncertainty still poses a challenge for VAEs on many real-world datasets.[51] Further, the observation noise process can be modeled explicitly in VAEs allowing the combination of different data types[34,35]: e.g., Poisson noise for spike counts and Gaussian noise for behavioral trajectories. Using a Gaussian negative log-likelihood loss allows estimation of observation noise for each data channel separately, a desirable property in scientific measurements, yet challenging to accomplish.[52–54] To assess overall calibration in VAEs, which reflects both the posterior uncertainty and observation noise, we introduced a version of simulation-based calibration,[41,42] which allows for sample-based uncertainty evaluation in the absence of tractable ground-truth distributions. Assessing calibration for discrete data poses additional challenges—in particular in a low count regime where it is not possible to obtain reliable confidence intervals—and remains an avenue for future investigation.[55]

Limitations of the study

Our masked VAE approach allows for calibrated predictions on a variety of conditioning tasks, but it has some limitations. First, our approach relies on a small number of specified and structured conditioning masks, rather than considering all possible combinatorial (2^D) masking conditions, where D is the data dimension. For many problems in neuroscience, this suffices since the conditional distributions of interest are usually few and well specified, such as behavior given neural data. To tackle the problem of capturing *all* conditionals, it would be an empirical question of how big models and datasets would need to be to effectively generalize to this combinatorial space. Furthermore, we have only evaluated our approach on data with simultaneous recordings of neurons (and behavior) and have yet to consider the case where, for example, different sets of neurons are recorded at different times. Previous work has demonstrated that it is possible to "stitch" neural population dynamics across multiple populations using non-VAE-based approaches.[56] Extending our method with such approaches is an exciting avenue for future work. Second, similar to other deep-learning-based methods, our approach struggles with capturing interactions on long and varying timescales. Here, we only investigated cross-modality interactions occurring on similar timescales (reach movements) and fixed the length of the time segments during training (maximum of 150 time steps). Thus, behavior or neural activity preceding this segment cannot influence subsequent predictions. Integrating transformer-based approaches[57,58] might be useful for capturing such interactions over varying timescales.[59–62] Third, our approach inherits common issues from VAEs, for example, the lack of a principled way to choose the dimensionality of the VAE latent space, rendering hyperparameter tuning potentially costly. If the dimensionality of the latent space is too large, the VAE might fail to exploit correlations within the data and use separate latent variables for the specified conditional distributions, potentially degrading the quality of conditional generation. On the other hand, if the dimensionality is too small, the VAE might not be able to accurately model the data. Thus,

for the monkey-reach task, we introduced a sparsity-inducing prior[63] that mitigates this issue by automatically reducing the latent dimensionality if latents are not used by the decoder network. Furthermore, deep-learning-based methods such as VAEs can sometimes require large amounts of data. While it is generally difficult to assess *a priori* how much data is required to reach maximum performance, we here show how to get an intuition through varying the training set size (Figure S3). In the Gaussian tractable example, masked VAEs achieve good performance with as few as 100 samples. Fourth, in this paper, we have not focused on the disentanglement of latents. For example, in the monkey encoding and decoding experiments, latents may contain information about both neural activity and behavior. Ideally, we would like to separate neural-only, behavior-only, and shared latents through disentanglement, as proposed in Higgins et al.[64,65] However, theoretical work has shown that, without external supervision, this is challenging.[66] Semi-supervised approaches, as proposed and applied in the neuroscientific context in Whiteway et al. and Yi et al.,[67,68] may be promising to explore in future work. Finally, while disentangled latent spaces is desirable for their interpretability, they do not necessarily lead to models with better generalization, for example, when reconstructing previously unseen combinations of different modalities (e.g., shape and color[69]). Lastly, while samples from our masked VAE are well-calibrated in most cases and often close to the ground-truth neural and behavioral trajectories, the sampling quality of VAEs is known to be limited even for simpler and fully observed datasets. Recent generative models such as Denoising Diffusion Probabilistic Models,[70,71] Normalizing Flows,[72] and Generative Adversarial Networks[73] can produce samples of higher quality, but they lack the main feature of VAEs that makes them especially relevant in neuroscience: inference of low-dimensional latent states. Combining our approach with other such generative models (e.g., Bashiri et al. and Vetter et al.[26,48]) could be an interesting future avenue to improve sampling quality while preserving the possibility of performing latent inference.

Conclusions

We present a method that addresses two common goals in neuroscience: inferring low-dimensional representations and unveiling dependencies in simultaneously recorded modalities by modeling their conditional distributions. Our approach will allow for scaling encoding and decoding analyses in neuroscience to today's high-dimensional multi-modal datasets. Furthermore, this work highlights a crucial aspect of analyzing neural and behavioral data: the importance of uncertainty estimates.

RESOURCE AVAILABILITY

Lead contact

Requests for further information and resources should be directed to and will be fulfilled by the lead contact, Auguste Schulz (auguste.schulz@uni-tuebingen.de).

Materials availability

This study did not generate any new materials.

Data and code availability

- The fly-walking dataset has been deposited at https://zenodo.org/records/11002776 and is publicly available as of the date of publication. The DOI is listed in the key resources table.
- The monkey-reach dataset by O'Doherty et al.[40] has been deposited at https://zenodo.org/records/583331 and is publicly available as of the date of publication. The DOI is listed in the key resources table.
- Code for this study and for generating the simulation dataset has been deposited at https://github.com/mackelab/neuro-behavior-conditioning and is publicly available as of the date of publication. The DOI is listed in the key resources table. The code is written in Python; important packages we used include PyTorch,[74] Sklearn,[75] DeepLabCut,[1] and Tracktor.[76]

ACKNOWLEDGMENTS

We thank Artur Speiser and Paul Fischer for data management and technical support, Lisa Haxel and Michael Deistler for feedback on the manuscript, and all Mackelab members for discussions. This work was supported by the German Research Foundation (DFG) through Germany's Excellence Strategy (EXC-Number 2064/1, PN 390727645) and SFB1233 (PN 276693517), SFB 1089 (PN 227953431), the German Federal Ministry of Education and Research (Tübingen AI Center, FKZ: 01IS18039), and the Human Frontier Science Program (HFSP), and the European Union (ERC, DeepCoMechTome, 101089288). A.S. and J.V. are members of the International Max Planck Research School for Intelligent Systems (IMPRS-IS). D.M. acknowledges a Marie Curie EuroTech postdoctoral fellowship, a Swiss Government Excellence Postdoctoral Scholarship (2018.0483), and funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement no. 754462. V.L.-R. acknowledges support from the Mexican National Council for Science and Technology, CONACYT, under the grant number 709993. P.R. acknowledges support from an SNSF Project grant (no. 175667) and an SNSF Eccellenza grant (no. 181239).

AUTHOR CONTRIBUTIONS

Conceptualization, A.S., P.R., P.J.G., and J.H.M.; methodology, A.S., D.M., V.L.-R., and P.R.; software, A.S. and V.L.-R.; validation, A.S.; formal analysis, A.S.; investigation, A.S., D.M., P.J.G., and J.H.M.; resources, A.S.; data curation, A.S., D.M., and V.L.-R.; writing – original draft, A.S.; writing – review & editing, A.S., J.V., R.G., D.M., V.L.-R., P.R., P.J.G., and J.H.M.; visualization, A.S.; supervision, P.J.G. and J.H.M.; project administration, A.S. and J.H.M.; funding acquisition, P.R. and J.H.M.

DECLARATION OF INTERESTS

The authors declare no competing interests.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES

During the preparation of this work, the authors used GitHub Copilot and ChatGPT in order to format LaTeX tables and edit code. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS
  - Drosophila melanogaster
  - Rhesus macaque monkeys
- METHOD DETAILS
  - Background on variational autoencoders

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.celrep.2025.115338.

Received: May 14, 2024
Revised: November 5, 2024
Accepted: January 30, 2025

REFERENCES

1. Mathis, A., Mamidanna, P., Cury, K.M., Abe, T., Murthy, V.N., Mathis, M.W., and Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. Nat. Neurosci. *21*, 1281–1289. https://doi.org/10.1038/s41593-018-0209-y.
2. Günel, S., Rhodin, H., Morales, D., Campagnolo, J., Ramdya, P., and Fua, P. (2019). DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult Drosophila. Elife *8*, e48571. https://doi.org/10.7554/eLife.48571.
3. Pereira, T.D., Aldarondo, D.E., Willmore, L., Kislin, M., Wang, S.S.H., Murthy, M., and Shaevitz, J.W. (2019). Fast animal pose estimation using deep neural networks. Nat. Methods *16*, 117–125. https://doi.org/10.1038/s41592-018-0234-5.
4. Ahrens, M.B., Orger, M.B., Robson, D.N., Li, J.M., and Keller, P.J. (2013). Whole-brain functional imaging at cellular resolution using light-sheet microscopy. Nat. Methods *10*, 413–420. https://doi.org/10.1038/nmeth.2434.
5. Sofroniew, N.J., Flickinger, D., King, J., and Svoboda, K. (2016). A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging. Elife *5*, e14472. https://doi.org/10.7554/eLife.14472.
6. Jun, J.J., Steinmetz, N.A., Siegle, J.H., Denman, D.J., Bauza, M., Barbarits, B., Lee, A.K., Anastassiou, C.A., Andrei, A., Aydın, Ç., et al. (2017). Fully integrated silicon probes for high-density recording of neural activity. Nature *551*, 232–236. https://doi.org/10.1038/nature24636.
7. de Vries, S.E.J., Lecoq, J.A., Buice, M.A., Groblewski, P.A., Ocker, G.K., Oliver, M., Feng, D., Cain, N., Ledochowitsch, P., Millman, D., et al. (2020). A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex. Nat. Neurosci. *23*, 138–151. https://doi.org/10.1038/s41593-019-0550-9.
8. Siegle, J.H., Jia, X., Durand, S., Gale, S., Bennett, C., Graddis, N., Heller, G., Ramirez, T.K., Choi, H., Luviano, J.A., et al. (2021). Survey of spiking in the mouse visual system reveals functional hierarchy. Nature *592*, 86–92. https://doi.org/10.1038/s41586-020-03171-x.
9. Mimica, B., Tombaz, T., Battistin, C., Fuglstad, J.G., Dunn, B.A., and Whitlock, J.R. (2023). Behavioral decomposition reveals rich encoding structure employed across neocortex in rats. Nat. Commun. *14*, 3947. https://doi.org/10.1038/s41467-023-39520-3.
10. Sani, O.G., Pesaran, B., and Shanechi, M.M. (2021a). Where is all the nonlinearity: flexible nonlinear modeling of behaviorally relevant neural dynamics using recurrent neural networks. preprint at bioRxiv. https://doi.org/10.1101/2021.09.03.458628.
11. Kriegeskorte, N., and Douglas, P.K. (2019). Interpreting encoding and decoding models. Curr. Opin. Neurobiol. *55*, 167–179. https://doi.org/10.1016/j.conb.2019.04.002.
12. Pillow, J.W., Shlens, J., Paninski, L., Sher, A., Litke, A.M., Chichilnisky, E.J., and Simoncelli, E.P. (2008). Spatio-temporal correlations and visual signalling in a complete neuronal population. Nature *454*, 995–999. https://doi.org/10.1038/nature07140.
13. Paninski, L., and Cunningham, J.P. (2018). Neural data science: accelerating the experiment-analysis-theory cycle in large-scale neuroscience. Curr. Opin. Neurobiol. *50*, 232–241. https://doi.org/10.1016/j.conb.2018.04.007.
14. Chen, Z.S., and Pesaran, B. (2021). Improving scalability in systems neuroscience. Neuron *109*, 1776–1790. https://doi.org/10.1016/j.neuron.2021.03.025.
15. Yu, B.M., Cunningham, J.P., Santhanam, G., Ryu, S.I., Shenoy, K.V., and Sahani, M. (2009). Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity. J. Neurophysiol. *102*, 614–635. https://doi.org/10.1152/jn.90941.2008.
16. Sussillo, D., Jozefowicz, R., Abbott, L.F., and Pandarinath, C. (2016). LFADS - latent factor analysis via dynamical systems. preprint at arXiv. https://doi.org/10.48550/arXiv.1608.06315.
17. Batty, E., Whiteway, M., Saxena, S., Biderman, D., Abe, T., Musall, S., Gillis, W., Markowitz, J., Churchland, A., Cunningham, J.P., et al. (2019). Behavenet: nonlinear embedding and bayesian neural decoding of behavioral videos. In Advances in Neural Information Processing Systems, *32* (Curran Associates, Inc.), pp. 15706–15717.
18. Keeley, S.L., Zoltowski, D.M., Aoi, M.C., and Pillow, J.W. (2020). Modeling statistical dependencies in multi-region spike train data. Curr. Opin. Neurobiol. *65*, 194–202. https://doi.org/10.1016/j.conb.2020.11.005.
19. Sani, O.G., Abbaspourazad, H., Wong, Y.T., Pesaran, B., and Shanechi, M.M. (2021b). Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. Nat. Neurosci. *24*, 140–149. https://doi.org/10.1038/s41593-020-00733-0.
20. Luxem, K., Mocellin, P., Fuhrmann, F., Kürsch, J., Miller, S.R., Palop, J.J., Remy, S., and Bauer, P. (2022). Identifying behavioral structure from deep variational embeddings of animal motion. Commun. Biol. *5*, 1267. https://doi.org/10.1038/s42003-022-04080-7.
21. Schneider, S., Lee, J.H., and Mathis, M.W. (2023). Learnable latent embeddings for joint behavioural and neural analysis. Nature *617*, 360–368. https://doi.org/10.1038/s41586-023-06031-6.
22. Pfau, D., Pnevmatikakis, E.A., and Paninski, L. (2013). Robust learning of low-dimensional dynamics from large neural ensembles. In Advances in Neural Information Processing Systems, *26* (Curran Associates, Inc.), pp. 2391–2399.
23. Schimel, M., Kao, T.C., Jensen, K.T., and Hennequin, G. (2021). iLQR-VAE : control-based learning of input-driven dynamics with applications to neural data. In International Conference on Learning Representations. https://doi.org/10.1101/2021.10.07.463540.
24. Jensen, K., Kao, T.C., Stone, J., and Hennequin, G. (2021). Scalable Bayesian GPFA with automatic relevance determination and discrete noise models. In Advances in Neural Information Processing Systems, *34* (Curran Associates, Inc.), pp. 10613–10626.
25. Hurwitz, C., Srivastava, A., Xu, K., Jude, J., Perich, M., Miller, L., and Hennig, M. (2021). Targeted neural dynamical modeling. In Advances in Neural Information Processing Systems, *34* (Curran Associates, Inc.), pp. 29379–29392.
26. Bashiri, M., Walker, E., Lurz, K.K., Jagadish, A., Muhammad, T., Ding, Z., Ding, Z., Tolias, A., and Sinz, F. (2021). A flow-based latent state generative model of neural population responses to natural images. In Advances in Neural Information Processing Systems, *34* (Curran Associates, Inc.), pp. 15801–15815.
27. Macke, J.H., Buesing, L., Cunningham, J.P., Yu, B.M., Shenoy, K.V., and Sahani, M. (2011). Empirical models of spiking in neural populations. In Advances in Neural Information Processing Systems, *24* (Curran Associates, Inc.), pp. 1350–1358.
28. Petreska, B., Yu, B.M., Cunningham, J.P., Santhanam, G., Ryu, S., Shenoy, K.V., and Sahani, M. (2011). Dynamical segmentation of single trials from population neural data. In Advances in Neural Information Processing Systems, *24* (Curran Associates, Inc.), pp. 756–764.

29. Linderman, S., Johnson, M., Miller, A., Adams, R., Blei, D., and Paninski, L. (2017). Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics Vol. 54 of *Proceedings of Machine Learning Research*, A. Singh and J. Zhu, eds. (PMLR), pp. 914–922.

30. Buesing, L., Macke, J.H., and Sahani, M. (2012). Spectral learning of linear dynamics from generalised-linear observations with application to neural population data. In Advances in Neural Information Processing Systems, *25* (Curran Associates, Inc.), pp. 1691–1699.

31. Girin, L., Leglaive, S., Bie, X., Diard, J., Hueber, T., and Alameda-Pineda, X. (2021). Dynamical Variational Autoencoders: A Comprehensive Review. FNT. in Machine Learning *15*, 1–175. https://doi.org/10.1561/2200000089.

32. Rezende, D.J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In Proceedings of the 31st International Conference on Machine Learning Vol. 32 of *Proceedings of Machine Learning Research*, E.P. Xing and T. Jebara, eds. (Bejing, China: PMLR), pp. 1278–1286.

33. Kingma, D.P., and Welling, M. (2014). Auto-encoding variational Bayes. In International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1312.6114.

34. Nazábal, A., Olmos, P.M., Ghahramani, Z., and Valera, I. (2020). Handling incomplete heterogeneous data using VAEs. Pattern Recogn. *107*, 107501. https://doi.org/10.1016/j.patcog.2020.107501.

35. Brenner, M., Hess, F., Koppe, G., and Durstewitz, D. (2024). Integrating multimodal data for joint generative modeling of complex dynamics. arXiv *235*, 4482–4516.

36. Zhou, D., and Wei, X.X. (2020). Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-VAE. In Advances in Neural Information Processing Systems, *33* (Curran Associates, Inc.), pp. 7234–7247.

37. Williams, C.K.I., Nash, C., and Nazábal, A. (2019). Autoencoders and Probabilistic Inference with Missing Data: An Exact Solution for The Factor Analysis Case. preprint at arXiv. https://doi.org/10.48550/arXiv.1801.03851.

38. Ivanov, O., Figurnov, M., and Vetrov, D. (2019). Variational autoencoder with arbitrary conditioning. In International Conference on Learning Representations. arXiv:1806.02382.

39. Collier, M., Nazábal, A., and Williams, C.K.I. (2020). VAEs in the presence of missing data. In ICML Workshop on the Art of Learning with Missing Values (Artemiss). arXiv:2006.05301.

40. O'Doherty, J.E., Cardoso, M.M.B., Makin, J.G., and Sabes, P.N. (2017). Nonhuman Primate Reaching with Multichannel Sensorimotor Cortex Electrophysiology (Zenodo). https://doi.org/10.5281/zenodo.583331.

41. Cook, S.R., Gelman, A., and Rubin, D.B. (2006). Validation of Software for Bayesian Models Using Posterior Quantiles. J. Comput. Graph Stat. *15*, 675–692. https://doi.org/10.1198/106186006X136976.

42. Talts, S., Betancourt, M., Simpson, D., Vehtari, A., and Gelman, A. (2018). Validating Bayesian Inference Algorithms with Simulation-Based Calibration.. preprint at arXiv. https://doi.org/10.48550/arXiv.1804.06788.

43. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), A. Moschitti, B. Pang, and W. Daelemans, eds. (Doha, Qatar: Association for Computational Linguistics), pp. 1724–1734. https://doi.org/10.3115/v1/D14-1179.

44. Makin, J.G., O'Doherty, J.E., Cardoso, M.M.B., and Sabes, P.N. (2018). Superior arm-movement decoding from cortex with a new, unsupervised-learning algorithm. J. Neural. Eng. *15*, 026010. https://doi.org/10.1088/1741-2552/aa9e95.

45. Pei, F.C., Ye, J., Zoltowski, D.M., Wu, A., Chowdhury, R.H., Sohn, H., O'Doherty, J.E., Shenoy, K.V., Kaufman, M., Churchland, M.M., et al. (2021). Neural latents benchmark '21: Evaluating latent variable models of neural population activity. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2). arXiv:2109.04463.

46. Afshar, A., Santhanam, G., Yu, B.M., Ryu, S.I., Sahani, M., and Shenoy, K.V. (2011). Single-trial neural correlates of arm movement preparation. Neuron *71*, 555–564. https://doi.org/10.1016/j.neuron.2011.05.047.

47. Talukder, S.J., Sun, J.J., Leonard, M.K., Brunton, B.W., and Yue, Y. (2022). Deep neural imputation: A framework for recovering incomplete brain recordings. In NeurIPS 2022 Workshop on Learning from Time Series for Health. arXiv:2206.08094v1.

48. Vetter, J., Macke, J.H., and Gao, R. (2024). Generating realistic neurophysiological time series with denoising diffusion probabilistic models. Patterns *5*, 101047. https://doi.org/10.1016/j.patter.2024.101047.

49. Guo, C., Pleiss, G., Sun, Y., and Weinberger, K.Q. (2017). On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning Vol. 70 of *Proceedings of Machine Learning Research*, D. Precup and Y.W. Teh, eds. (PMLR), pp. 1321–1330.

50. Wei, G., Mansouri, Z.T., Wang, X., and Stevenson, I.H. (2024). Calibrating Bayesian decoders of neural spiking activity. Journal of Neuroscience *44*, e2158232024. https://doi.org/10.1523/JNEUROSCI.2158-23.2024.

51. Wang, Y., Blei, D., and Cunningham, J.P. (2021). Posterior collapse and latent variable non-identifiability. In Advances in Neural Information Processing Systems, *34* (Curran Associates, Inc.), pp. 5443–5455.

52. Rybkin, O., Daniilidis, K., and Levine, S. (2021). Simple and Effective VAE Training with Calibrated Decoders. In Proceedings of the 38th International Conference on Machine Learning Vol. 139 of *Proceedings of Machine Learning Research*, M. Meila and T. Zhang, eds. (PMLR), pp. 9179–9189.

53. Skafte, N., Jørgensen, M., and Hauberg, S. (2019). Reliable training and estimation of variance networks. In Advances in Neural Information Processing Systems, *32* (Curran Associates, Inc).

54. Seitzer, M., Tavakoli, A., Antic, D., and Martius, G. (2022). On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks. In International Conference on Learning Representations. https://doi.org/10.48550/arXiv.2203.09168.

55. Wei, W., and Held, L. (2014). Calibration tests for count data. TEST *23*, 787–805. https://doi.org/10.1007/s11749-014-0380-8.

56. Turaga, S., Buesing, L., Packer, A.M., Dalgleish, H., Pettit, N., Hausser, M., and Macke, J.H. (2013). Inferring neural population dynamics from multiple partial recordings of the same neural circuit. In Advances in Neural Information Processing Systems (Curran Associates, Inc.).

57. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, .u., and Polosukhin, I. (2017). Attention is All you Need. In Advances in Neural Information Processing Systems, *30* (Curran Associates, Inc.). arXiv:1706.03762.

58. Jaegle, A., Borgeaud, S., Alayrac, J.B., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Zoran, D., Brock, A., Shelhamer, E., et al. (2022). Perceiver IO: A general architecture for structured inputs & outputs. In International Conference on Learning Representations. arXiv:2107.14795.

59. Ye, J., and Pandarinath, C. (2021). Representation learning for neural population activity with neural data transformers. Neuron. Behav. Data Anal. Theory *5*, 1–18. https://doi.org/10.51628/001c.27358.

60. Ye, J., Collinger, J., Wehbe, L., and Gaunt, R. (2023). Neural Data Transformer 2: Multi-context Pretraining for Neural Spiking Activity. In Advances in Neural Information Processing Systems (Curran Associates, Inc.), pp. 80352–80374.

61. Azabou, M., Arora, V., Ganesh, V., Mao, X., Nachimuthu, S., Mendelson, M., Richards, B., Perich, M., Lajoie, G., and Dyer, E. (2023). A Unified, Scalable Framework for Neural Population Decoding. In Advances in Neural Information Processing Systems (Curran Associates, Inc.), pp. 44937–44956.

62. Antoniades, A., Yu, Y., Canzano, J.S., Wang, W.Y., and Smith, S. (2024). Neuroformer: Multimodal and Multitask Generative Pretraining for Brain Data. In International Conference on Learning Representations. arXiv:2311.00136.

63. Ainsworth, S.K., Foti, N.J., Lee, A.K.C., and Fox, E.B. (2018). oi-VAE: Output interpretable VAEs for nonlinear group factor analysis. In Proceedings of the 35th International Conference on Machine Learning, J. Dy and A. Krause, eds. (PMLR), pp. 119–128.

64. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In International Conference on Learning Representations.

65. Higgins, I., Chang, L., Langston, V., Hassabis, D., Summerfield, C., Tsao, D., and Botvinick, M. (2021). Unsupervised deep learning identifies semantic disentanglement in single inferotemporal face patch neurons. Nat. Commun. *12*, 6456. https://doi.org/10.1038/s41467-021-26751-5.

66. Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In Proceedings of the 36th International Conference on Machine Learning, K. Chaudhuri and R. Salakhutdinov, eds. (PMLR), pp. 4114–4124.

67. Whiteway, M.R., Biderman, D., Friedman, Y., Dipoppa, M., Buchanan, E.K., Wu, A., Zhou, J., Bonacchi, N., Miska, N.J., Noel, J.P., et al. (2021). Partitioning variability in animal behavioral videos using semi-supervised variational autoencoders. PLoS Comput. Biol. *17*, e1009439. https://doi.org/10.1371/journal.pcbi.1009439.

68. Yi, D., Musall, S., Churchland, A., Padilla-Coreano, N., and Saxena, S. (2023). Disentangled multi-subject and social behavioral representations through a constrained subspace variational autoencoder (cs-vae). Elife *12*, e88602. https://doi.org/10.7554/eLife.88602.1.

69. Montero, M.L., Ludwig, C.J., Costa, R.P., Malhotra, G., and Bowers, J. (2020). The role of disentanglement in generalisation. In International Conference on Learning Representations.

70. Ho, J., Jain, A., and Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. In Advances in Neural Information Processing Systems, *33* (Curran Associates, Inc.), pp. 6840–6851.

71. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10684–10695.

72. Rezende, D.J., and Mohamed, S. (2015). In Variational inference with normalizing flows, *Vol.* (Lille, France: 37 of Proceedings of Machine Learning Research), pp. 1530–1538.

73. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Advances in Neural Information Processing Systems, *27* (Curran Associates, Inc.). arXiv:1406.2661.

74. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems (Curran Associates, Inc.).

75. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. J. Mach. Learn. Res. *12*, 2825–2830.

76. Sridhar, V.H., Roche, D.G., and Gingins, S. (2019). Tracktor: Image-based automated tracking of animal movement and behaviour. Methods Ecol. Evol. *10*, 815–820. https://doi.org/10.1111/2041-210X.13166.

77. Sen, R., Wu, M., Branson, K., Robie, A., Rubin, G.M., and Dickson, B.J. (2017). Moonwalker descending neurons mediate visually evoked retreat in drosophila. Curr. Biol. *27*, 766–771. https://doi.org/10.1016/j.cub.2017.02.008.

78. Kingma, D.P., and Welling, M. (2019). An introduction to variational autoencoders. FNT. in Machine Learning *12*, 307–392. https://doi.org/10.1561/2200000056.

79. Gregor, K., Danihelka, I., Graves, A., Rezende, D., and Wierstra, D. (2015). Draw: A recurrent neural network for image generation. In Proceedings of the 32nd International Conference on Machine Learning Vol. 37 of *Proceedings of Machine Learning Research*, F. Bach and D. Blei, eds. (Lille, France: PMLR), pp. 1462–1471.

80. Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., and Bengio, Y. (2016). A Recurrent Latent Variable Model for Sequential Data. In Proceedings of the 28th International Conference on Neural Information Processing Systems -, *2*, pp. 2980–2988.

81. Pandarinath, C., O'Shea, D.J., Collins, J., Jozefowicz, R., Stavisky, S.D., Kao, J.C., Trautmann, E.M., Kaufman, M.T., Ryu, S.I., Hochberg, L.R., et al. (2018). Inferring single-trial neural population dynamics using sequential auto-encoders. Nat. Methods *15*, 805–815.

82. Savitzky, A., and Golay, M.J.E. (1964). Smoothing and differentiation of data by simplified least squares procedures. Anal. Chem. *36*, 1627–1639. https://doi.org/10.1021/ac60214a047.

83. Kingma, D.P., and Ba, J. (2015). Adam: A method for stochastic optimization. In International Conference on Learning Representations. arXiv:1412.6980.

84. Loshchilov, I., and Hutter, F. (2019). Decoupled Weight Decay Regularization. In International Conference on Learning Representations. arXiv:1711.05101.

85. Javaloy, A., Meghdadi, M., and Valera, I. (2022). Mitigating modality collapse in multimodal VAEs via impartial optimization. In Proceedings of the 39th International Conference on Machine Learning Vol. 162 of *Proceedings of Machine Learning Research*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds. (PMLR), pp. 9938–9964.

86. Biewald, L. (2020). Experiment tracking with weights and biases. URL: https://www.wandb.com/softwareavailablefromwandb.com.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Walking behavior of Drosophila melanogaster	this paper	https://doi.org/10.5281/zenodo.11002775
Experimental models: Organisms/strains		
D. melanogaster: Moonwalker fly: 20XUAS-CsChrimson-mVenus (attP18)/+; 050660-p65ADZp (attP40)/+; 044845-ZpGAL4DBD (attP2)/+	Sen et al.[77]	MDN-3
Public Monkey Reach Dataset	O'Doherty et al.[40]	https://doi.org/10.5281/zenodo.788569
Software and algorithms		
Original code	this paper	https://github.com/mackelab/neuro-behavior-conditioning https://doi.org/10.5281/zenodo.14766113
Pytorch	Paszke et al.[74]	https://github.com/pytorch/pytorch
Sklearn	Pedregosa et al.[75]	https://github.com/scikit-learn/scikit-learn
Python	Python Software Foundation	https://www.python.org/
DeepLabCut	Mathis et al.[1]	http://www.mackenziemathislab.org/deeplabcut
Tracktor	Sridhar et al.[76]	https://github.com/vivekhsridhar/tracktor

EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS

Drosophila melanogaster

Female flies (Drosophila melanogaster) were placed in an acrylic arena that constrained them to move in a 2D plane. The flies were part of a genetic screen (20XUAS-CsChrimson-mVenus (attP18)/+; 050660-p65ADZp (attP40)/+; 044845-ZpGAL4DBD (attP2)/+,[77] i.e., not wild-type), but were largely morphologically and behaviorally indistinguishable from wild-type flies. All experiments were performed with adult flies, i.e., at least 7 days after emerging from their pupa. Flies were maintained at 25°C and 50% humidity. All experiments were performed in compliance with relevant national (Switzerland) and institutional (EPFL) ethical regulations.

Rhesus macaque monkeys

Neural and behavioral data from a male Rhesus macaque monkey (Macaca mulatta) who performed self-paced reaches were recorded and made publicly available by O'Doherty et al.[40] At the time of data collection, the monkey was 9 years old and weighed 14.5 kg.[44] As stated by the lab performing the experiments, all animal procedures were performed in accordance with the U.S. National Research Council's Guide for the Care and Use of Laboratory Animals and were approved by the UCSF Institutional Animal Care and Use Committee.[40,44]

Given the methodological nature, we do not expect any sex related influence on our results.

METHOD DETAILS

Here, we adapt variational autoencoders[32,33] to address two goals simultaneously: First, to infer low-dimensional representations underlying multi-modal neural and behavioral time-series data and, second, to model their conditional distributions. Modeling conditional distributions is ubiquitous in neuroscience, and since neuroscientific data are typically variable even in controlled experiments, relations between modalities may also be variable. Therefore, we focus on probabilistic rather than deterministic approaches to characterize such conditional distributions. We reformulate the estimation of conditional distributions in VAEs in a more general way: modeling the distribution of an unobserved subset of the data given an observed subset $p(\text{unobserved}|\text{observed})$ similar to Nazábal et al. and Collier et al.[34,39] To target such distributions with a VAE, we modify the training scheme and loss of classical VAEs. We validate our approach on a tractable example and two neuroscientific time-series datasets: walking behavior of the fly and a continuous reach task in monkeys. We introduce calibration metrics to evaluate the models' uncertainty estimates in the context of scientific data, i.e., without access to ground-truth uncertainties.

Background on variational autoencoders

Variational Autoencoders (VAEs) are probabilistic models capable of capturing complex multi-modal data distributions $p(\mathbf{x})$. The assumption underlying VAEs is that all variations in the data distributions can be captured (up to observation/measurement noise) by the variations of corresponding unobserved latent variables z. VAEs learn stochastic mappings between the observed data space and the unobserved or latent (z-space). Both mappings from the data to latent distributions and vice versa are typically parameterized through flexible neural networks. The generative model is described by the joint distribution of data and latent variables, which factorizes into

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z}) p_\theta(\mathbf{x}|\mathbf{z}) \tag{1}$$

parameterized by θ where $p(\mathbf{z})$ is the prior over the latent space. The prior is usually chosen to be a simple distribution such as a standard Gaussian, and $p_\theta(\mathbf{x}|\mathbf{z})$ is the probabilistic decoder. The inference or encoder model $q_\phi(\mathbf{z}|\mathbf{x})$, parameterized by ϕ, which infers the latent distribution from data, is an approximation of the true, intractable posterior $p(\mathbf{z}|\mathbf{x})$[33,32,78]. VAEs are trained by maximizing a lower bound of the data log-likelihood. This so-called Evidence Lower Bound (ELBO) can be written as:

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{D}_{\mathrm{KL}}[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})]. \tag{2}$$

The first term assesses how well the predicted distribution matches the original data and is often referred to as the reconstruction loss. The second term is the Kullback-Leibler divergence D_{KL} between the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ and the latent prior $p(\mathbf{z})$, which regularizes the learned latent space. Maximizing the ELBO $\mathcal{L}_{\theta,\phi}(\mathbf{x})$ with respect to the parameters θ and ϕ leads to a better generative model and increases the similarity between the approximate and the true (intractable) posterior. All parameters θ and ϕ can be optimized jointly using stochastic gradient descent. Once trained successfully, one can sample from the prior and consecutively from the stochastic decoder output to obtain a new sample $\tilde{x}_{\mathrm{pred}}$ from the learned data distribution. Alternatively, one can sample from the approximate posterior of a previously unseen test datum x_{test} to obtain reconstructions that closely resemble x_{test}. VAEs have been successfully applied to various types of potentially heterogeneous data (continuous, discrete, ordinal, etc.)[34] and have been extended to time series,[31,79,80] paving the way for applications to neuroscientific time series.[16,20,35,81]

Capturing arbitrary conditional distributions with VAEs

To model flexible conditional distributions with VAEs, we modified the training scheme of classical VAEs similar to Nazábal et al. and Collier et al.[34,39] While training the VAE, we randomly mask out subsets of the data corresponding to the desired conditional distributions and compute the loss on the remaining data (see Method S1). Selecting a subset of the data that is held out for specific analyses may seem related to cross-validation. It is important to distinguish that while cross-validation separates independent subsets of the data to assess the generalization properties of a model, here, the data selection happens in terms of individual features, i.e., dependent data dimensions. Prior to training, for each conditional distribution of interest, we specify a conditioning mask m together with a mask probability p_m (Figure 1C, left). During training, the conditioning masks are sampled independently for each data point according to the mask probabilities. Concretely, the masking is performed by replacing the data with their respective mean values. Other replacement values, such as zeros for spiking (count) data, are also possible. We calculate the reconstruction loss $\mathcal{L}_{recon}$ solely on observed, that is, non-masked data. Sometimes, to facilitate learning that some data has been masked out, we additionally provide the encoder network with a binary mask consisting of 0s for unobserved and 1s for observed data points (see Methods, networks). Through this training procedure, the masked VAE simultaneously optimizes the ELBO over all different conditional distributions, that is

$$\mathcal{L}^{\mathrm{masked}}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{m\sim p(m)}\left[\mathcal{L}^{m}_{\theta,\phi}(\mathbf{x})\right], \tag{3}$$

where $p(m)$ is the previously specified probability distribution over all conditioning masks, including the fully observed case, where no data is masked out. As noted above, the mask m is applied to both the data and the corresponding part of the reconstruction loss. This training procedure promotes the learning of different encoder networks that share parameters, allowing us to target different approximate posterior distributions given different conditioning masks. From an implementation perspective, the conditioning masks can be passed to the encoder network in various ways. They can, for example, be concatenated or added directly to the input, but also at later stages in the network, possibly after transformations with a (non-linear) embedding. In contrast to Collier et al.,[39] we explicitly do not pass the conditioning masks to the decoder since all uncertainty and mean shifts induced by masking should be reflected in the latent representation.

Modeling observation noise with VAEs

It is important to ensure that VAEs correctly capture the uncertainties in the (conditional) data distributions. In a VAE, there are two sources of uncertainty - the inferred posterior uncertainty and the observation or measurement noise. The latter source of uncertainty is often ignored, which is reflected in the common choice of the mean squared error (MSE) as the reconstruction loss. The MSE only evaluates the quality of the mean prediction and ignores the stochastic nature of the VAE decoder. If we instead want to correctly capture the observation noise, it is necessary to learn it explicitly. Assuming that the observation noise follows a

Gaussian distribution, we use the Gaussian negative log-likelihood (GNLL) as our reconstruction loss. The GNLL for an observation x given a model prediction of the Gaussian mean μ and standard deviation σ is given by

$$\mathcal{L}_{\text{GNLL}}(x;\mu,\sigma) = -\log P(X = x;\mu,\sigma) = \frac{1}{2}\log\left(2\pi\sigma^2\right) + \frac{(x-\mu)^2}{2\sigma^2}. \quad (4)$$

Note that the MSE is a special case of the GNLL where the standard deviation is set to 1. As noted above, this usually leads to samples from the model that are not calibrated in the statistical sense. Optimization, however, is more challenging when using the GNLL and might require additional adjustments.[52,54]

Datasets and data preprocessing

Linear Gaussian Latent Variable Model

We simulated a dataset based on a Gaussian Latent Variable Model (GLVM) with one latent variable z, where $z \backsim \mathcal{N}(0,1)$, and 20 data dimensions $\mathbf{x}$, where $\mathbf{x} \backsim \mathcal{N}(\mathbf{C}z + \mathbf{d}, \mathbf{\Lambda})$ (Figure 2A). To demonstrate the difference between noisy and more precise, less noisy, variables in a setup that accounts for uncertainty, noise levels for all data dimensions differ. For each data dimension i, σ_i is drawn from a log-normal distribution with $\mu_{LN} = \log(0.7)$ and $\sigma_{LN} = 0.5$. C_is are drawn from a normal distribution with $\mu_N = 1.1$ and $\sigma_N^2 = 0.1$. Additionally, the sign for a given C_i is flipped with probability 0.5. All offsets $\mathbf{d}$ are set to 1. We use 9000 samples from this model for training and 1000 each for validation and testing. This fully Gaussian setup allows for the analytical computation of both conditional $p(\mathbf{x}^{\text{unobs}}|\mathbf{x}^{\text{obs}})$ and posterior $p(z|\mathbf{x}^{\text{obs}})$ distributions (see Method S2), which we compare to the distributions learned by our model.

Fly walking behavior

To collect the data on fly walking behavior, we placed flies, *Drosophila melanogaster*, in an acrylic arena that constrained them to move in a 2D plane. Thus, flying is not part of the otherwise rich repertoire of observed behaviors, which includes both forward and backward walking, grooming, resting, etc. The flies were part of a genetic screen (not wild-type) but were examined during behavior capture and were morphologically and behaviorally indistinguishable from wild-type flies. We placed three female flies in one arena simultaneously and filmed them from below, three times for 3 s (frame rate of 80*Hz*). This procedure was repeated about 10000 times, resulting in 28059 time series with 234 time points each. To extract each fly from the video separately, we tracked the centroid of each fly using Tracktor,[76] cropped out the flies in each frame, and aligned them to point in the upward direction. We then tracked 32 body parts (four joints per leg, as well as head features, thorax, abdomen, and wings), each with x- and y-directions using DeepLabCut,[1] resulting in time series with 64 feature dimensions. We then smoothed the extracted time series using a Savitzky-Golay-Filter[82] with a polynomial order of two and a window length of seven. The smoothed trajectories were then cut into sequences of length 48 with buffers of length 9 between each sequence to avoid information leakage. 95% of the data was used for training, while the remaining data was used for validation and testing (2806 sequences each). Potential information leakage due to autocorrelation between training and test/validation sets is further reduced by choosing the last sequences for testing/validation instead of an interleaved approach, which can often cause information leakage in time-series models. Prior to passing the time series to the network, we standardize each feature dimension across the 48 time steps.

Continuous reach task in monkeys

The neural and behavioral dataset, made publicly available by O'Doherty et al.,[40] was recorded from two monkeys (rhesus macaque) performing self-paced continuous reaches, i.e., without gaps or pre-movement delay intervals. Targets were arranged in an 8 by 8 grid, and a new target was presented when the previous target was reached. Neural recordings were taken from the cortical hemisphere contralateral to the arm performing the reach movements. O'Doherty et al.[40] provide the neural data after spike sorting in the shape channels vs. spike times. We focus only on one session, 'loco_20170213_02', which contains neural activity from both primary motor (M1) and (S1) activity, as well as the cursor, target, and finger positions. Here, we take only the neural activity from M1 and the cursor positions (x,y direction) as the behavioral correlate. We filter out channels with firing rates below 0.5 Hz analogous to Makin et al.,[44] resulting in 213 remaining M1 units. We convert spike times into spike counts in bins of 64 ms (15.625 Hz). We down-sample the cursor position by querying it at fewer time points consistent with the reduced sampling rate used for binning the spikes (15.625 Hz instead of 250 Hz). We do not introduce a delay between neural activity and behavior as done, e.g., in Schimel et al. and Jensen et al.,[23,24] we rather let the model identify which aspects of the respective other time series to consider for its predictions. We use the first 70% of the data for training (approx. 28 min recording time), the following 10% for testing (approx. 4 min), and the remaining 20% for validation (approx. 8 min). We standardize the behavioral train, test, and validation time series with respect to the overall mean and standard deviation of the training set for both reach directions. During training, we introduce 'pseudo-trials' with 150 time steps each, that start at randomly sampled time points.

Network architectures and optimization

Model details: GLVM

Training scheme and masks: Prior to training, we select three randomly sampled masks (10 out of 20 dimensions are masked) to test if the masked approach can capture the true posterior and conditional distributions. Since we chose different loading factors C_i and noise levels σ_i for each dimension, the corresponding posterior mean and variance, and thus also the conditional distributions,

differ between conditions. During training, we uniformly sampled the four conditioning masks (all observed and mask 1–3) and used the Adam optimizer[83] to train our model.

Architecture: The encoder network consists of a simple linear embedding for the mask, which is passed through a multilayer perceptron together with the 20-dimensional data vector to parameterize the one-dimensional posterior mean and log variance. To focus on posterior inference under different masking conditions, we set the decoder to be the true generative model. Note, however, that this GLVM example is identifiable, i.e., all parameters of the generative model (C_i, d_i, σ_i) can be learned using a VAE, which we confirmed even in our masked training scheme. Nevertheless, fixing the decoder is beneficial in this case since the posterior is only identifiable up to a rotation in the latent space (μ_z, σ_z), which in the one-dimensional setting corresponds to a flipped sign. See Table S1 in the supplement for hyperparameters.

Loss: For this well-specified, identifiable Gaussian example, the regular masked GNLL was used together with a standard Gaussian prior in the latent space.

Model details: Fly walking behavior

Training scheme and masks: To investigate low-dimensional representations of fly walking behavior, we built a sequential VAE and specified masks for the body keypoints most relevant to walking. Analogous to the GLVM case, we adapted the masked training scheme for the time-series case to allow for modeling the conditional distribution over a subset of the fly body keypoints, given the remaining keypoints. Specifically, we mask the hind claw, hind tibia-tarsal joint, mid tibia-tarsal joint, and mid claw of the left side. The entire time segment of masked keypoints is replaced with the mean value across this segment. We assign a probability of 50% to the all-observed and the leg-masking condition. We again use the Adam optimizer[83] with a learning rate of 0.0005 for training our model.

Architecture: The VAE for fly walking behavior consists of an encoder and a decoder network that are both trainable neural networks. The encoder network consists of two sets of 1D convolutional layers, each followed by batch normalization and ELU activation. We then apply temporal convolutions that compress the data in the temporal dimension before passing it to a bidirectional RNN[43] for temporal context. Thus, the encoder network is non-causal in time. The RNN output is then passed through a multi-layer perceptron to parameterize the posterior mean and log variance. This results in a latent space with spatial (N_z) and temporal (T_z) dimensions smaller than the 64 features and 48 time-steps of the data (for our choice of parameters $N_z = 18$ and $T_z = 13$, i.e., the size of the latent space is less than 8% of the original data). Unlike in the GVLM case, we do not pass the mask to the encoder network, since it does not improve the conditional modeling. After sampling from the approximate posterior, the decoder network expands the time dimension of $\mathbf{z}$ using transposed convolutions, followed by dimensionality expansion to parameterize the Gaussian mean and observation noise variance. The latter is constrained to be positive by a softplus function to ensure well-defined variances. See Table S2 in the supplement for hyperparameters.

Loss: For the continuous behavioral data, we again use GNLL (Equation 4), which is computed per feature and timepoint. The prior distribution in the latent space is standard Gaussian.

Model details: Neural and behavioral data from a monkey reach task

Training scheme and masks: The sequential VAE for the monkey reach task jointly models time series of high-dimensional neural spike-count data and continuous cursor positions. We specify the masks required for neuroscientific encoding and decoding: either all neural activity is masked out (spike counts set to zero), or all behavioral traces are masked and set to their respective mean values. Following Ainsworth et al.,[63] we introduced a sparsity-inducing prior that sets latent contributions to zero if they are not used by the model. We used the AdamW optimizer[84] with a learning rate of 0.001 and weight decay of 0.2) to train our encoder and decoder networks. For parameters related to sparsity-induction, we follow Ainsworth et al.[63] and use Stochastic Gradient Descent (SGD) with zero momentum. Here, we show results that are trained on only one session, but the architecture allows training on data from multiple sessions using session-specific input and output mappings.

Architecture: First, we expand the data using a session-specific linear mapping. Similar to the sequential VAE for the fly data, the encoder network then performs a non-linear dimensionality reduction followed by a bidirectional RNN to parameterize the latent posterior mean and log-variance. Here, we do not consider compression in time, and each latent time-point corresponds to a time-point in dataspace. The decoder also has an RNN and uses further multi-layer-perceptrons to map the latent samples back into data space. For the continuous behavioral data, the decoder again predicts the Gaussian mean and observation noise variance. For the discrete spike data, however, the decoder only models the underlying firing rates. See Table S3 in the supplement for hyperparameters.

Loss: This discrepancy arises from the different distributions used to model the respective data modalities. While behavior is continuous and thus appropriately modeled with a Gaussian, discrete spike counts are best modeled with a Poisson distribution. Consequently, the GNLL is replaced by the negative Poisson log-likelihood, which, for an observed spike count x and a rate parameter λ, is defined as:

$$\mathcal{L}_{\text{Poisson}}(x;\lambda) = -\log P(X = x;\lambda) = -x\log(\lambda) + \lambda + \log(x!). \tag{5}$$

For this task, we also refined the behavioral GNLL loss by incorporating the concept of β-NLL introduced by Seitzer et al.[54] by weighting each data point's contribution to the loss based on the β-NLL-exponentiated variance estimate $\sigma_i^{2\beta_{NLL}}$. Effectively, this small loss modification prevents a potential issue when learning the observation noise, namely that poorly fitted variables are assigned high variance, which, since it appears in the denominator in Equation 4, leads to smaller gradients and hence less incentives for the network to improve its fit. In this application, an NLL-beta of 0.3 worked well. Before calculating the overall gradients, we sum up

the behavioral and neural contributions to the reconstruction loss. Note that when using heterogeneous noise models the scales of the loss contributions can vastly differ. Hence, depending on the application and downstream tasks, it can be beneficial or even necessary to introduce weighting factors to balance out the losses and corresponding gradients.[85] To improve stability and prevent over-fitting, we additionally regularize the session input and output weight matrices. For the sparsity-inducing prior in the latent space, we apply a version of Lasso regularization that encourages sparsity in the weight matrix that transforms the z-samples before they are passed through the decoder (see[63] for details).

Linear decoding from neural latents: For the experiment of partial neural observations, we set up identical VAEs with the main difference that the VAE receives no behavioral input and output. We vary between passing in all neural activity and masked activity with 5, 20, 50, 100, 150, or 200 masked neurons of 213. Note that the (Poisson-)loss is then only computed on the remaining observed neurons. Since we are only dealing with a single discrete modality, we did not incorporate the GNLL loss nor any NLL-beta scaling. All other training configurations and parameter settings remained identical. At test time, we apply the masks to the training data and assess the inferred latent means and associated uncertainty estimates (standard deviation returned by the VAE). For each method and model instantiation, we identify the most informative latents as those with the highest variability in the inferred latent mean over time. Unused or less relevant latents converge to the prior, which in these experiments was set to mean 0 and standard deviation 1. Unused latents hardly vary over time and can thus be easily identified post hoc. Setting the correct latent dimension *a priori* poses a challenge (see Limitations). After training the VAE, we trained a linear ridge regression model ($\alpha = 0.01$) to predict behavior (velocity) given latents (means) of the fully observed condition. When training the decoder, we shifted the velocity by 128 ms following Jensen et al.[24] and Schimel et al.[23] We scaled both neural activity and behavior using min-max scaling, with scaling factors obtained on the training set and applied to the validation and test sets. We quantify decoding performance as the correlation between the predicted velocity (y-direction) and the true velocity. We apply the same decoding strategy to raw spiking data, training on all observed spiking activity and predicting velocity from only partial test recordings.

Computational cost

We used Weights and Biases[86] to track VAE training times. Since the architectures of masked and naive VAEs are almost identical and masked VAEs have only a few extra parameters to optimize, the compute times are comparable: For the GLVM task, training naive VAEs took 4.58 ± 0.08 min (mean ± standard deviation) and masked VAEs 4.57 ± 0.10 min; for the fly task, 145.18 ± 24.21 min (naive) and 155.05 ± 15.46 min (masked); for the monkey encoding and decoding task, 7.61 ± 0.12 min (naive) and 7.36 ± 0.06 min (masked); for the monkey neural latents task, 6.22 ± 0.25 min (naive) and 6.21 ± 0.29 min (masked). We did not perform early stopping, so the reported times do not reflect the time until convergence. Furthermore, for naive VAEs, smaller architectures may have been sufficient to accurately model the fully observed case. We trained and evaluated masked and naive VAEs on the Gaussian and Monkey reach datasets on an NVIDIA RTX 3090 (24GB RAM). The Fly dataset, which is significantly larger, was trained on a GeForce RTX 2080 Ti (11GB RAM), which is slower than the 3090. Therefore, the numbers between the different datasets are not directly comparable.

QUANTIFICATION AND STATISTICAL ANALYSIS

Calibration metrics: Evaluating uncertainties in variational autoencoders

To investigate the statistical calibration in VAEs, we perform a version of simulation-based calibration,[41,42] which is associated to frequentist coverage tests.[50] Here, we focus on the calibration of the predictive distribution in data space. For each test datum x_{test}, we sample n_z-times from the approximate posterior, pass the sampled z through the decoder and sample from the observation noise model n_{obs}-times. This sampling procedure results in a sampling distribution in data space that reflects both posterior uncertainty and observation noise.

For continuous data, we then compute confidence intervals corresponding to the nth percentile. For a statistically well-calibrated model, n% of the ground truth data should lie in this interval. When plotting the different percentiles against the proportion of data points falling in the corresponding interval, well-calibrated predictions lie on the diagonal. Here, we evaluate this for the 60th, 80th, 90th, and 95th percentiles, which roughly correspond to one to three standard deviations. However, other evaluations, including percentile bins from 0 to 100, are also common (see e.g., Wei et al.[50]). If a model is overconfident, the corresponding values fall in the lower triangle below the diagonal. In the underconfident regime, i.e., if the average predictions are accurate, but the estimated uncertainty is too high, the values fall in the region above the diagonal. In all three datasets, we evaluate the calibration of masked continuous variables (GLVM, fly walking behavior, and behavior in the monkey reach task). The test set sizes, as well as the computational cost of estimating confidence intervals, differ between models. Therefore, we chose different n_z and n_{obs} to compute the confidence intervals but kept both values the same when evaluating the naive model to ensure a fair comparison.

For count data, we compute cumulative distribution functions (CDFs) of the spike counts to assess the calibration of the predicted firing rate. This is because most bin counts are either 0 or 1, making it impractical to construct confidence intervals.[55] More specifically, to obtain informative CDFs, we aggregate five neighboring bin counts of time-series with 200 time bins. Then, for each of the $n_z \cdot n_{obs}$ predicted rate time series, we sample spikes and compute the CDF over all 40 aggregated bins. Finally, we plot the obtained CDFs against the analogously aggregated CDF of the ground truth spike train. If the rate predictions are well calibrated, their resulting CDFs closely match the ground truth CDF and lie on the diagonal.

Decoding performance vs. latent uncertainty We evaluate the decoding performance of VAE latents obtained from partial neural recordings together with the VAE latent uncertainty in the most informative latents. For each of seven masking patterns (all observed, 5, 20, 50, 100, 150, or 200 masked neurons out of 213), we plot the mean latent standard deviation of the most informative latent against the corresponding decoding performance, fitting the relationship using linear regression. We can then evaluate the relationship for each model using the R^2 values, slopes, and associated p-values.

Cell Reports, Volume *44*

Supplemental information

Modeling conditional distributions of neural and behavioral data with masked variational autoencoders

Auguste Schulz, Julius Vetter, Richard Gao, Daniel Morales, Victor Lobato-Rios, Pavan Ramdya, Pedro J. Gonçalves, and Jakob H. Macke

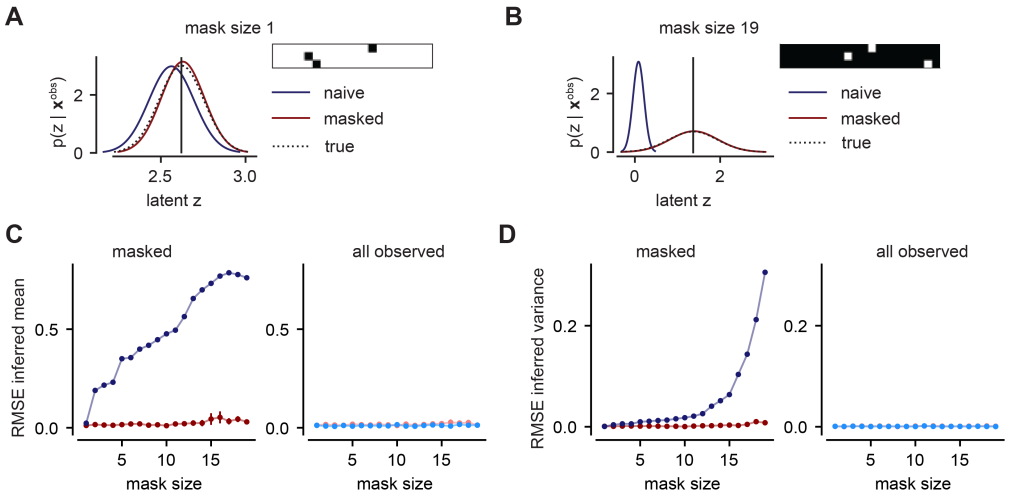


Figure S1. Effect of the ratio of masked / observed data on inference quality, related to Figure 2.

(A) Posterior or latent inference for the Gaussian dataset with 1 out of 20 data dimensions masked and (B) 19 out of 20 dimensions masked. Masked VAEs accurately increase the posterior uncertainty: masked VAEs estimate a broad distribution and naive VAEs a narrow and biased distribution. (C) RMSE between the true and the inferred posterior mean of the test set for different numbers of masked dimensions. For the masked condition, the RMSE of the masked approach is lower than that of the naive approach, for which the RMSE increases dramatically with increasing masked values. For all masking conditions, the RMSE remains low for masked VAEs, demonstrating that it has learned to both accurately infer the mean when all data points are observed and when most are missing. Data are represented as mean ± standard deviation over several seeds. (D) RMSE for the posterior variance. Same analysis as in C.

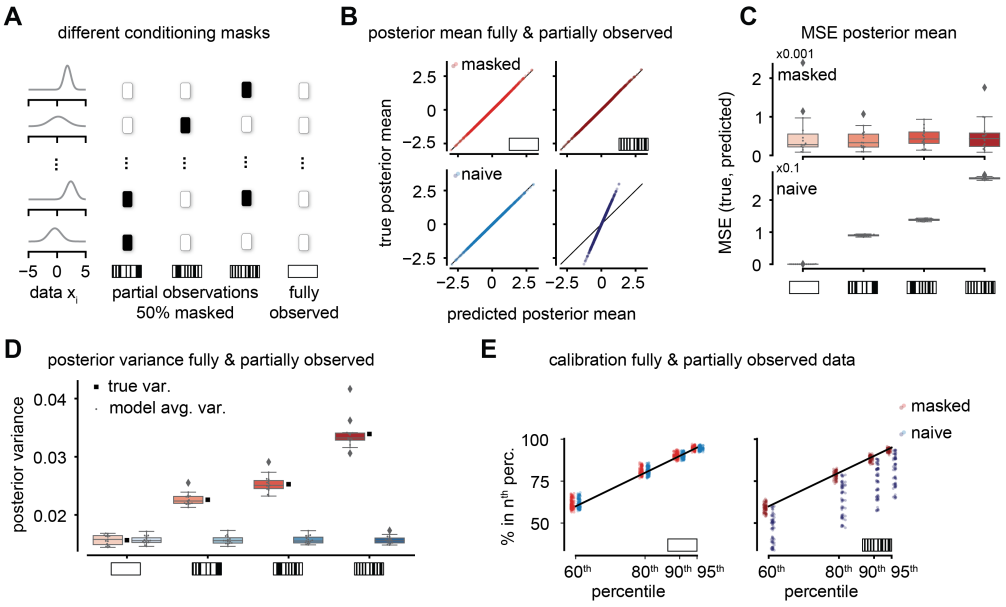


Figure S2. Posterior inference for different masking conditions, related to Figure 2.

(A) We test three different masking conditions where half of the input data dimensions are masked in addition to the fully observed case. (B) Masked VAEs correctly capture the posterior mean for partially and fully observed conditions, while the naive VAE can only perform correct inference in the fully observed condition. (C) The mean squared error (MSE) between the true and predicted posterior mean is low for all conditions for masked but not naive VAEs. Note the different orders of magnitude. Data are represented as mean over test samples. Box plots show the median and lower and upper quartiles. (D) Masked VAEs appropriately adjust the posterior variance for all conditions, while naive VAEs always wrongly predict the variance level of the fully observed condition (white rectangle). Data are represented as in C. (E) Both naive and masked VAEs are well calibrated in the fully observed condition (left). Yet, only masked VAEs are calibrated when modeling conditional distributions (right).

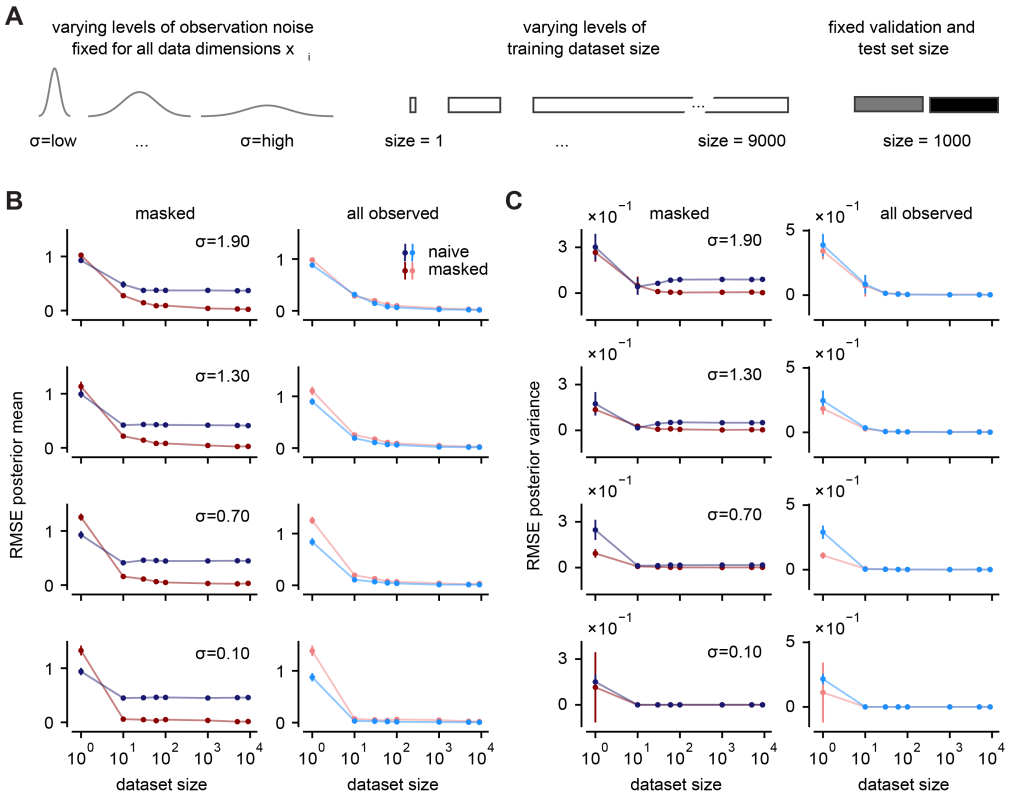


Figure S3. Assessing the effect of the training set size on inference quality in different noise regimes, related to Figure 2.

(A) We vary the overall levels of observation noise, (i.e., $\Lambda = \mathrm{diag}(\sigma^2)$), as well as the amount of available training data. The validation and test set sizes are fixed at 1000 samples to ensure comparability. (B) Root Mean Squared Error (RMSE) between the true (analytical) and the inferred posterior mean (on the test set) after training on different amounts of training samples (dataset size, x-axis). The RMSE is lower for the masked approach than for the naive one, and both decrease with increasing dataset size. The lower the noise level, the faster the performance converges. As expected, for the "all observed" case, the naive approach converges slightly faster, but both converge to similar values. For each dataset size, data are represented as the mean ± standard deviation over several seeds. (C) RMSE between the true (analytical) and the inferred posterior variance (same layout and data representation as in B). The RMSE in the masked condition for masked VAEs steadily decreases and reaches lower levels than for naive VAEs, while for the all-observed case, the performance is comparable across dataset sizes and noise levels.

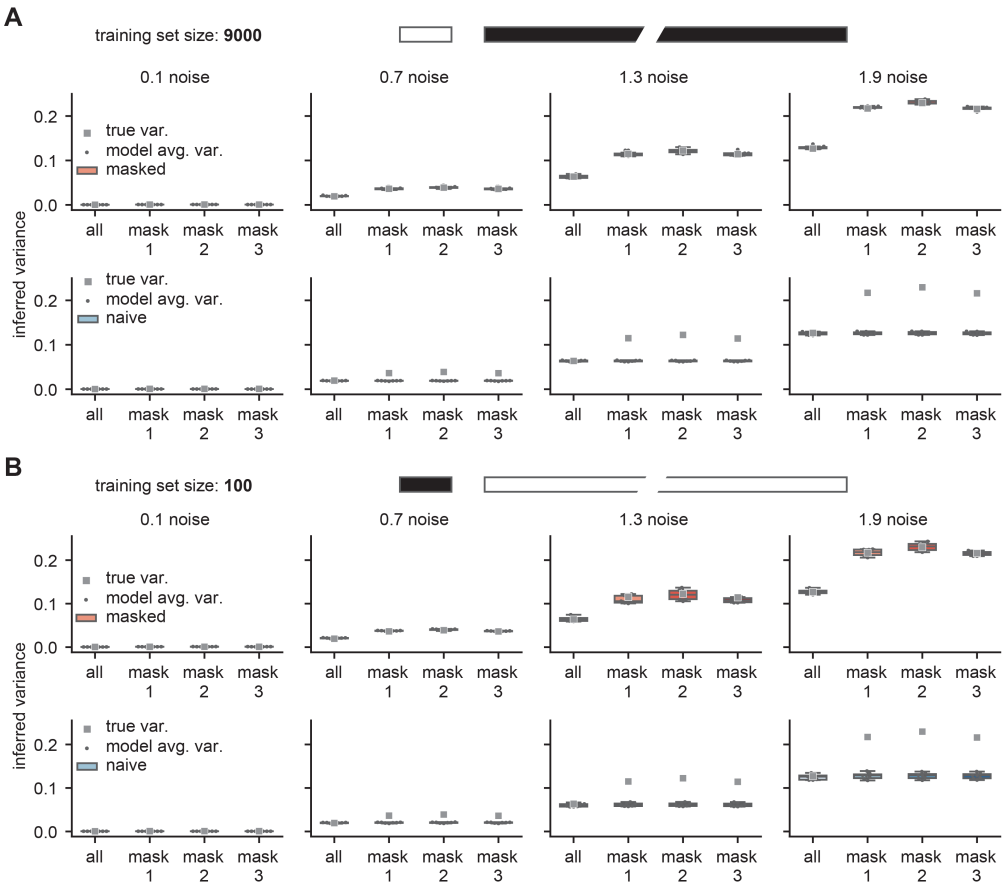


Figure S4. Inferred variance for different training set sizes and observation noise regimes, related to Figure 2.

(A) Average inferred variance of test set samples after training on 9000 training data points. Masked VAEs successfully infer the correct posterior variance across different observation noise levels. The true posterior variance (gt) is increasing with increasing observation noise. Data are represented as mean over test samples. Box plots show the median and lower and upper quartiles. (B) 100 training data points already suffice to achieve good performance in this example. Data representation as in A.

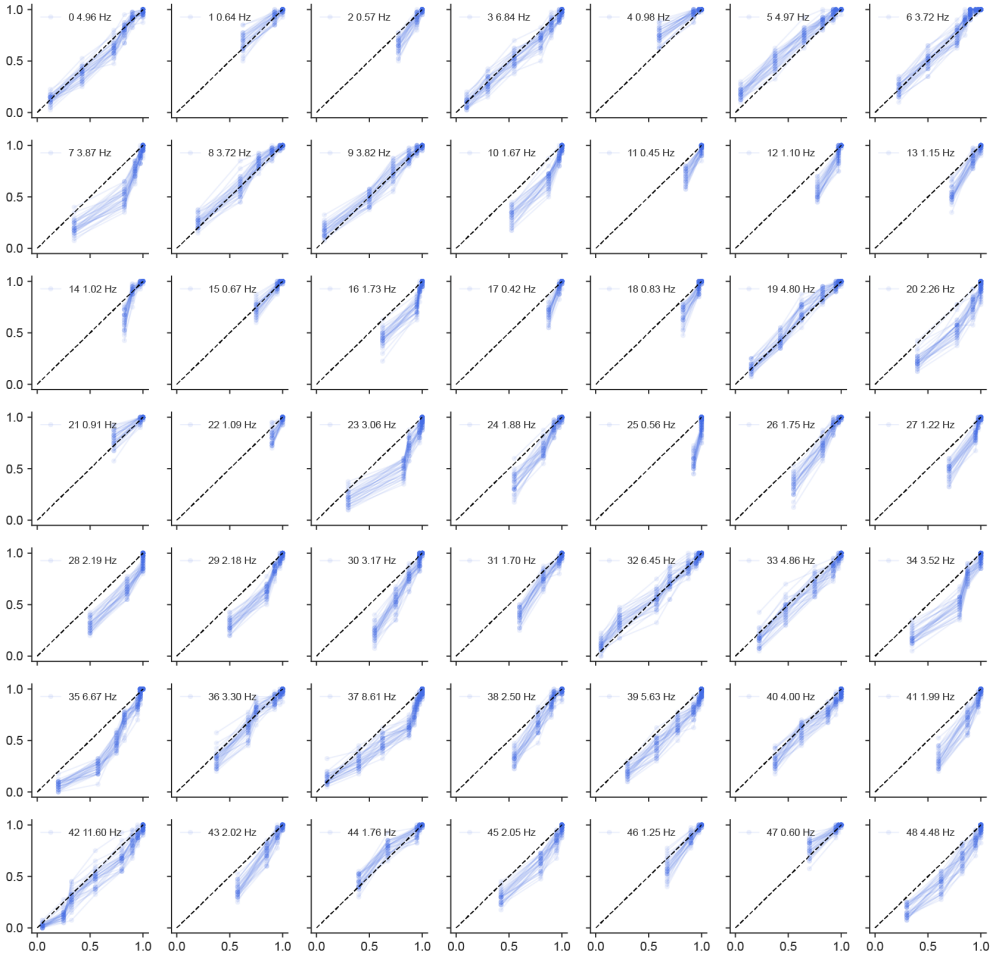


Figure S5. Additional naive cumulative distribution functions, related to Figure 5.

Subset of 49 units from monkey primary motor cortex. Sample CDFs of the encoding predictions of the naive VAE.

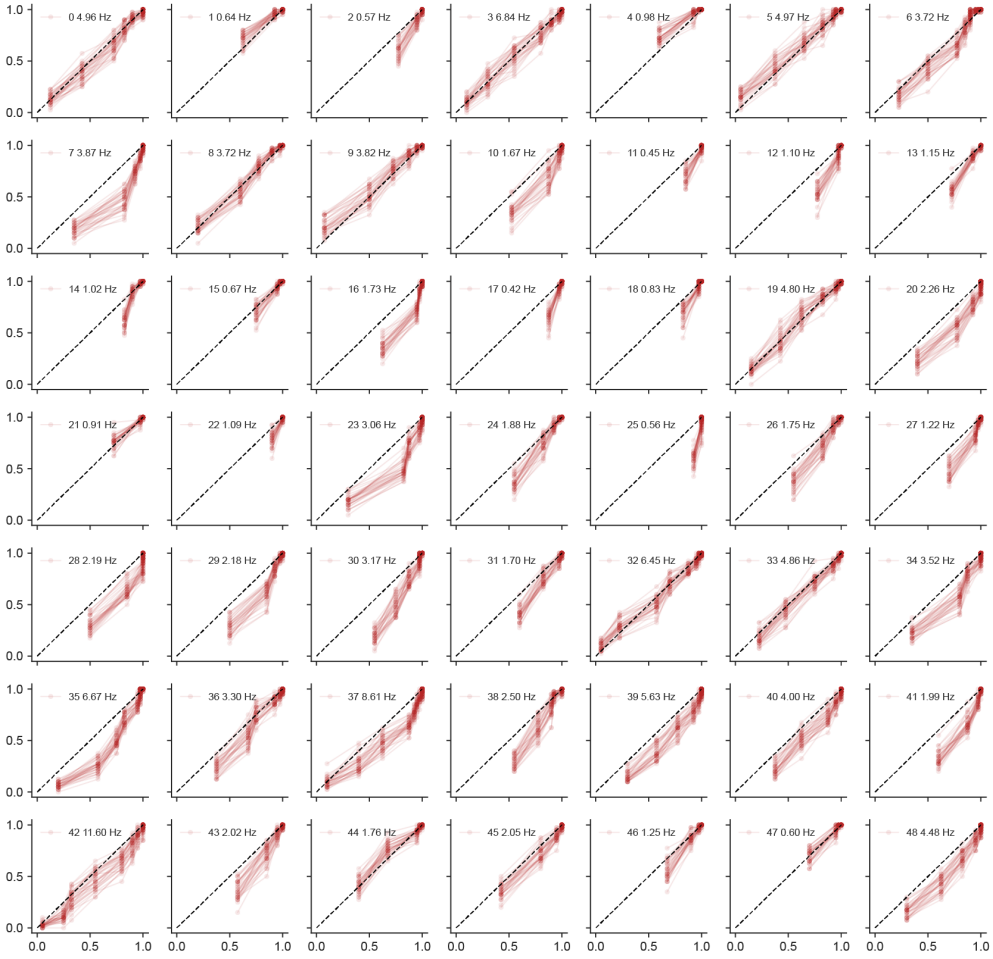


Figure S6. Additional masked cumulative distribution functions, related to Figure 5.

Subset of 49 units from monkey primary motor cortex. Sample CDFs of the encoding predictions of the masked VAE.

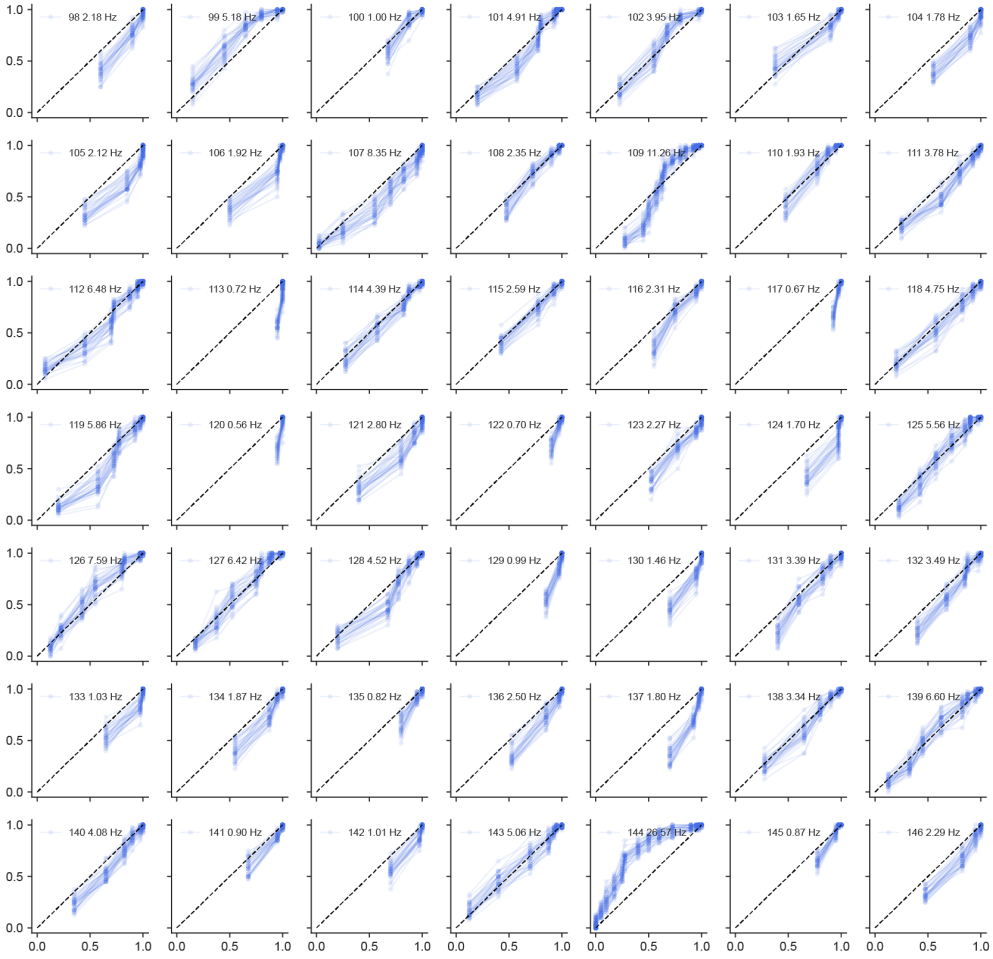


Figure S7. Additional naive cumulative distribution functions, related to Figure 5.

Subset of 49 units from monkey primary motor cortex. Sample CDFs of the encoding predictions of the naive VAE.

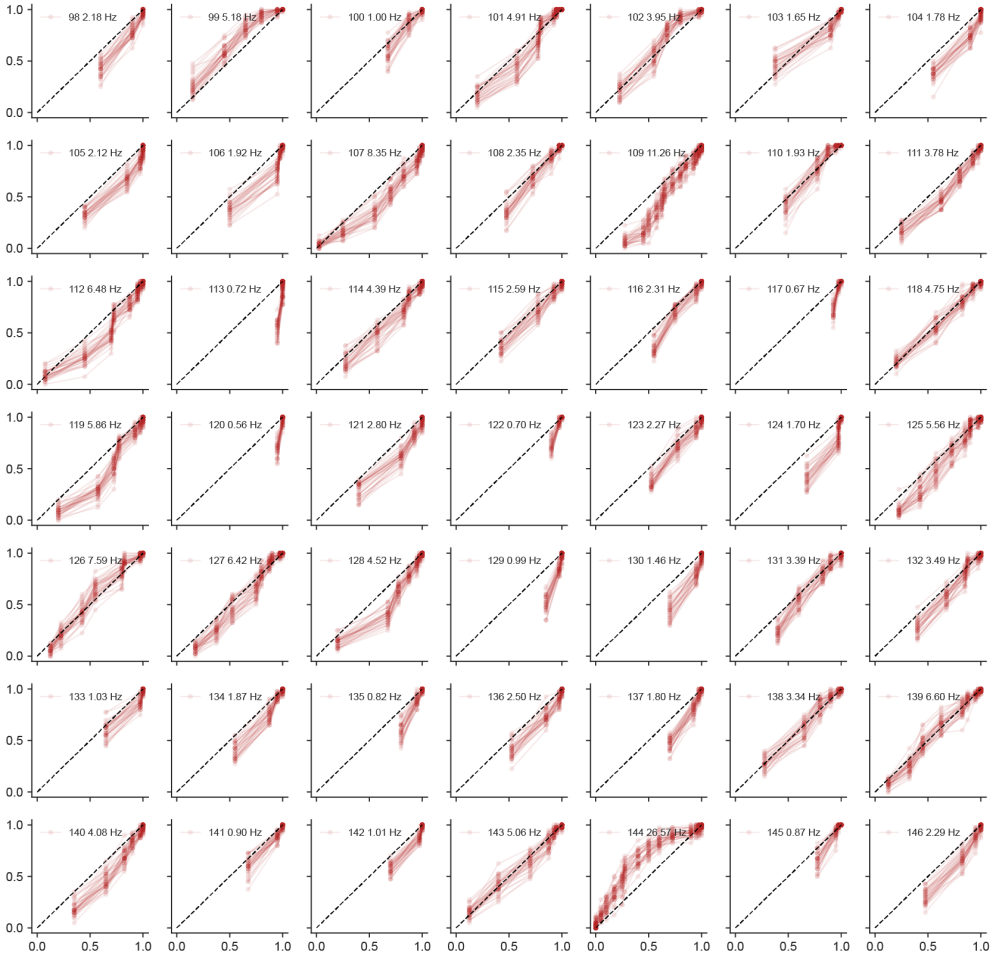


Figure S8. Additional masked cumulative distribution functions, related to Figure 5.

Subset of 49 units from monkey primary motor cortex. Sample CDFs of the encoding predictions of the masked VAE.

Table S1. Summary of relevant hyperparameters and dataset information for the GLVM VAE training, related to STAR Methods.

Hyperparameter	Value
Learning Rate	0.001
Beta	1.0
Warmup Range	30 epochs
Beta Step	1/30
Epochs	1500
Fraction fully observed	0.25
Latent Size	1
Training Batch Size	1000
Validation/Test Batch Size	1000
Dataset dimensions (samples, dimensions)	(11000, 20)
training, validation, test samples	9000, 1000, 1000
Trainable Parameters	7442

Table S2. Summary of relevant hyperparameters and dataset information for the fly VAE training, related to STAR Methods.

Hyperparameter	Value
Learning Rate	0.0005
Beta	1.0
Warmup Range	epochs 3-6, beta=0 before
Beta Step	0.25
Epochs	700
Epoch when masking starts	250
Fraction fully observed	0.5
Latent Size	18
Training Batch Size	256
Validation/Test Batch Size	512
Dataset dimensions (samples, sequence length, key points)	(28059, 234, 64)
training, validation, test split	95%, 2.5%, 2.5%
Sequence length cut for training	48
Trainable Parameters	39426

Table S3. Summary of relevant hyperparameters and dataset information for the monkey reach VAE training, related to STAR Methods.

Hyperparameter	Value
Learning Rate	0.001
Learning Rate sparsity	0.005
Beta	1.0
Warmup Range	50 iterations
Beta Step	1/50
NLL Beta	0.3
Iterations	25000
Iteration when masking starts	5000
Fraction fully observed	0.5
Max. Latent Size	40
Training Batch Size	32
Dataset dimensions	total time: 26439, units: 213, behavior: 2
training, validation, test split	70%, 20%, 10%
Sequence length cut for training	150
Trainable Parameters	379177

Method S1. Related work on VAEs for partial observations, related to STAR Methods.

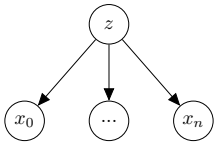
Our presented approach inherits most ideas and methodology from Nazábal et al.[S1] and Collier et al.[S2].

Methodological similarities and differences Nazábal et al.[S1] have demonstrated how to handle heterogeneous data with VAEs, i.e., how to combine different reconstruction losses. Inspired by this, we combined the Poisson loss for neural data with the Gaussian loss for continuous behavioral data. Furthermore, they showed how to compute the evidence lower bound on incomplete datasets by computing it only on the observed data. As such, this paper laid the technical foundation for our approach. Conceptually, we built on their findings on how to handle heterogeneous and incomplete data and treat the modality over which we want to learn the conditional distribution as missing. Collier et al.[S2] have expanded the work of Nazábal et al.[S1], most notably by exploring the effect of passing the missingness mask to the encoder and decoder networks. Here, while we make it optional to pass the mask to the encoder, we never pass the mask to the decoder, as all information about missingness should be mediated through the latent space. Passing the mask to the decoder alters the likelihood function that is approximated by the decoder, making it challenging to compare the learned latent distributions across masking patterns.

Domain differences The major difference lies in the application domain: Nazábal et al.[S1] focus on benchmark classification tasks and tabular data with missingness (Adult, Breast, Default Credit, Letter, Spam and Wine - Markelle Kelly, Rachel Longjohn, Kolby Nottingham, The UCI Machine Learning Repository, `https://archive.ics.uci.edu`). Collier et al.[S2] focus on image imputation tasks using typical machine learning benchmark datasets such as MNIST (Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, 29(6), 141–142.) and The Street View House Numbers (Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, Andrew Y. Ng Reading Digits in Natural Images with Unsupervised Feature Learning NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011). In contrast, we focus on neuroscientific time series and sequential VAEs and have shown how to recast calculating conditional distributions arising in neuroscience in such a general missingness framework. As such, we bridge the machine learning literature on missingness with neuroscience, allowing the community to benefit from the methodological advances in another field.

Method S2. Supplementary Information Gaussian Latent Variable Model, related to STAR Methods.

Here, we consider a linear Gaussian latent variable model [S3] with a one-dimensional latent space and n-dimensional observations x.



$$\begin{aligned} p(z) &= \mathcal{N}(z; \mu_z, \sigma_z^2) \\ p(x_i \mid z) &= \mathcal{N}(x_i; C_i z + d_i, \sigma_i^2) \\ p(\mathbf{x} \mid z) &= \mathcal{N}(\mathbf{x}; \mathbf{C}z + \mathbf{d}, \mathbf{\Lambda}), \end{aligned} \tag{1}$$

where $\mathbf{\Lambda}) = \mathrm{diag}(\sigma_i^2)$.

Joint distribution of latent and observed variables

For a GLVM as outlined above the joint distribution of latent and observed variables can be written as

$$p(z, x_0, ..., x_n) = \mathcal{N}\left(\begin{bmatrix} z \\ x_0 \\ ... \\ x_n \end{bmatrix}; \begin{bmatrix} \mu_z \\ C_0\mu_z + d_0 \\ ... \\ C_n\mu_z + d_n \end{bmatrix}, \begin{bmatrix} \sigma_z^2 & C_0\sigma_z^2 & ... & C_{n-1}\sigma_z^2 & C_n\sigma_z^2 \\ & C_0^2\sigma_z^2 + \sigma_0^2 & ... & C_0C_{n-1}\sigma_z^2 & C_0C_n\sigma_z^2 \\ & & C_1^2\sigma_z^2 + \sigma_1^2 & & ... \\ & & & ... & C_{n-1}C_n\sigma_z^2 \\ & & & & C_n^2\sigma_z^2 + \sigma_n^2 \end{bmatrix} \right), \tag{2}$$

$$= \mathcal{N}\left(\begin{bmatrix} z \\ \mathbf{x} \end{bmatrix}; \begin{bmatrix} \mu_z \\ \mathbf{C}\mu_z + \mathbf{d} \end{bmatrix}, \begin{bmatrix} Q & R \\ R^\top & S \end{bmatrix} \right), \tag{3}$$

$$\text{where } \mathbf{C}^\top = \begin{bmatrix} C_0 & ... & C_n \end{bmatrix}, \ \mathbf{d}^\top = \begin{bmatrix} d_0 & ... & d_n \end{bmatrix}, \tag{4}$$

$$R = \begin{bmatrix} C_0\sigma_z^2 & ... & C_n\sigma_z^2 \end{bmatrix}, \tag{5}$$

$$Q = \begin{bmatrix} \sigma_z^2 \end{bmatrix}, \text{and} \tag{6}$$

$$S = \begin{bmatrix} C_0^2\sigma_z^2 + \sigma_0^2 & ... & C_0C_{n-1}\sigma_z^2 & C_0C_n\sigma_z^2 \\ & C_1^2\sigma_z^2 + \sigma_1^2 & & ... \\ & & ... & C_{n-1}C_n\sigma_z^2 \\ & & & C_n^2\sigma_z^2 + \sigma_n^2 \end{bmatrix}. \tag{7}$$

Marginalization in the fully Gaussian case corresponds to ignoring the rows and columns of the mean and covariance matrices of the joint distribution that correspond to the variable we aim to marginalize out. This means that the covariance of e.g. the joint of only the data dimensions $p(x_0, ..., x_n)$ is simply the submatrix that leaves out the first row and column of (2). Corresponding adjustments to the matrices when marginalizing out certain variables are denoted as $\hat{Q}, \hat{R}, \hat{S}, \hat{C}$ and $\hat{d}$.

Posterior distribution - inferring the latent distribution given observations

In order to compute the posterior of $p(z|\mathbf{x})$, we can apply the conditioning rule for Gaussians [S3] and obtain

$$p\left(z \mid \mathbf{x}\right) = \mathcal{N}(z; \mu_z + RS^{-1}(\mathbf{x} - \mathbf{C}\mu_z - \mathbf{d}), Q - RS^{-1}R^\top). \tag{8}$$

If some variables are unobserved, we can still perform exact inference of the posterior distribution $p\left(z \mid \mathbf{x}^{\text{obs}}\right)$ [S3,S4]. We obtain the analytical result by first marginalizing out the respective unobserved variables in the joint distribution in Equation 2 before performing the conditioning step as in Equation 8.

$$p\left(z \mid \mathbf{x}^{\text{obs}}\right) = \mathcal{N}(z; \mu_z + \hat{R}\hat{S}^{-1}(\hat{\mathbf{x}} - \hat{\mathbf{C}}\mu_z - \hat{\mathbf{d}}), Q - \hat{R}\hat{S}^{-1}\hat{R}^\top) \tag{9}$$

where the hat indicates altered matrices where respective entries corresponding to unobserved values have been dropped.

Hence, for each masking pattern, a different matrix inversion has to be carried out. Also, see the discussion of exact inference in the factor analysis model in Williams et al. [S4]. It is worth noting here that the posterior variance does not depend on the exact x values. Intuitively, the posterior variance (uncertainty) will increase if some of the inputs are unobserved.

Accurate posterior inference and conditional sampling in masked VAEs with fixed decoders

As pointed out in the Methods section, the masked training procedure promotes the learning of different encoder networks that share parameters. This allows for targeting different approximate posterior distributions given different conditioning masks: $q_\phi(\mathbf{z}|\mathbf{x}^{\text{all obs}})$ will most likely not be equal to $q_\phi(\mathbf{z}|\mathbf{x}^{\text{partial obs}})$, where partial obs indicates that some but not all x-dimensions where observed in contrast to all obs. The generative model of the VAE $p_\theta(\mathbf{x}|\mathbf{z})$, however, should not depend on the masking condition. Once this relationship from latent to data space is learned and once we can correctly infer the approximate posterior given only observed variables $q_\phi(\mathbf{z}|\mathbf{x}^{\text{obs}})$ we can correctly capture conditional distributions of unobserved data $\mathbf{x}^{\text{unobs}}$ given observed data $\mathbf{x}^{\text{obs}}$:

$$\begin{aligned} p(\mathbf{x}^{\text{unobs}}|\mathbf{x}^{\text{obs}}) &= \int p(\mathbf{x}^{\text{unobs}}, \mathbf{z}|\mathbf{x}^{\text{obs}})\, d\mathbf{z} \\ &= \int p(\mathbf{x}^{\text{unobs}}|\mathbf{z}, \mathbf{x}^{\text{obs}}) p(\mathbf{z}|\mathbf{x}^{\text{obs}})\, d\mathbf{z} \\ &= \int p(\mathbf{x}^{\text{unobs}}|\mathbf{z}) p(\mathbf{z}|\mathbf{x}^{\text{obs}})\, d\mathbf{z} \\ &= \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}|\mathbf{x}^{\text{obs}})}[p(\mathbf{x}^{\text{unobs}}|\mathbf{z})]. \end{aligned} \tag{10}$$

Note that we used that $p_\theta(\mathbf{x}|\mathbf{z})$ can be written as $p_\theta(\mathbf{x}^{\text{obs}}, \mathbf{x}^{\text{unobs}}, |\mathbf{z})$ and since $\mathbf{x}^{\text{obs}}$ and $\mathbf{x}^{\text{unobs}}$ are conditionally independent given $\mathbf{z}$, $p(\mathbf{x}^{\text{unobs}}|\mathbf{z}, \mathbf{x}^{\text{obs}})$ simplifies to $p(\mathbf{x}^{\text{unobs}}|\mathbf{z})$. Hence, assuming we have correctly learned to infer $p(\mathbf{z}|\mathbf{x}^{\text{obs}})$ and $p(\mathbf{x}^{\text{unobs}}|\mathbf{z})$, we can sample from all sorts of conditional distributions we specify prior to training with structured masks.

Supplemental References

[S1] Nazábal, A., Olmos, P.M., Ghahramani, Z., and Valera, I. (2020). Handling incomplete heterogeneous data using VAEs. Pattern Recognition 107, 107501. doi: 10.1016/j.patcog. 2020.107501.

[S2] Collier, M., Nazábal, A., and Williams, C.K.I. (2020). VAEs in the presence of missing data. In ICML Workshop on the Art of Learning with Missing Values (Artemiss). doi: arXiv:2006. 05301.

[S3] Bishop, C.M. (2006). Pattern Recognition and Machine Learning. Springer.

[S4] Williams, C.K.I., Nash, C., and Nazábal, A. (2019). Autoencoders and Probabilistic Inference with Missing Data: An Exact Solution for The Factor Analysis Case. arXiv. doi: 10.48550/ arXiv.1801.03851.

Latent Diffusion for Neural Spiking Data

Jaivardhan Kapoor[1] * **Auguste Schulz**[1] * **Julius Vetter**[1]

Felix Pei[1] **Richard Gao**[1] † **Jakob H. Macke**[1,2] †

[1]Machine Learning in Science, University of Tübingen & Tübingen AI Center, Tübingen, Germany
[2]Department Empirical Inference, Max Planck Institute for Intelligent Systems, Tübingen, Germany
*Equal contribution, order determined by a coin toss.
†Equal supervision.
{firstname.lastname@uni-tuebingen.de}

Abstract

Modern datasets in neuroscience enable unprecedented inquiries into the relationship between complex behaviors and the activity of many simultaneously recorded neurons. While latent variable models can successfully extract low-dimensional embeddings from such recordings, using them to generate realistic spiking data, especially in a behavior-dependent manner, still poses a challenge. Here, we present Latent Diffusion for Neural Spiking data (LDNS), a diffusion-based generative model with a low-dimensional latent space: LDNS employs an autoencoder with structured state-space (S4) layers to project discrete high-dimensional spiking data into continuous time-aligned latents. On these inferred latents, we train expressive (conditional) diffusion models, enabling us to sample neural activity with realistic single-neuron and population spiking statistics. We validate LDNS on synthetic data, accurately recovering latent structure, firing rates, and spiking statistics. Next, we demonstrate its flexibility by generating variable-length data that mimics human cortical activity during attempted speech. We show how to equip LDNS with an expressive observation model that accounts for single-neuron dynamics not mediated by the latent state, further increasing the realism of generated samples. Finally, conditional LDNS trained on motor cortical activity during diverse reaching behaviors can generate realistic spiking data given reach direction or unseen reach trajectories. In summary, LDNS simultaneously enables inference of low-dimensional latents and realistic conditional generation of neural spiking datasets, opening up further possibilities for simulating experimentally testable hypotheses.

1 Introduction

Modern datasets in neuroscience are becoming increasingly high-dimensional with fast-paced innovations in measurement technology [1, 48, 23], granting access to hundreds to thousands of simultaneously recorded neurons. At the same time, the types of animal behaviors and sensory stimuli under investigation have become more naturalistic and complex, resulting in experimental setups with heterogeneous trials of varying length, or lacking trial structure altogether [35, 33, 57]. Therefore, a key target in systems neuroscience has shifted towards understanding the relationship between high-dimensional neural activity and complex behaviors.

For high-dimensional neural recordings, analyses that infer low-dimensional structures have been very useful for making sense of such data [11]. For example, latent variable models (LVMs) are often used to identify neural population dynamics not apparent at the level of single neurons [61, 32, 40]. More recently, deep learning-based approaches based on variational autoencoders (VAEs) [26, 44,

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

51, 36, 63, 22, 6, 16] have become particularly popular for inferring latent neural representations due to their expressiveness and ability to scale to large, heterogeneous neural recordings with behavioral covariates [36, 63, 16].

However, in addition to learning latent representations, another important consideration is the ability to act as faithful generative models of the data. In other words, models should be able to produce diverse, realistic samples of the neural activity they were trained on, ideally in a behavior- or stimulus-dependent manner. Models with such capabilities not only afford better interpretability analyses and diagnoses for whether structures underlying the data are accurately learned, but have a variety of downstream applications surrounding the design of closed-loop *in silico* experiments. For example, with faithful generative models, one can simulate population responses to hypothetical sensory, electrical, or optogenetic stimuli, as well as possible neural activity underlying hypothetical movement patterns. Most VAE-based approaches focus on the interpretability of the inferred latents, but not the ability to generate realistic and diverse samples when conditioning on external covariates, while sample-realistic models (e.g., based on generative adversarial networks (GANs) [17]) do not provide access to underlying low-dimensional representations. As such, there is a need for models of neural population spiking activity that both provide low-dimensional latent representations *and* can (conditionally) generate realistic neural activity.

Here, we propose **Latent Diffusion for Neural Spiking data (LDNS)**, which combines the ability of autoencoders to extract low-dimensional representations of discrete neural population activity, with the ability of (conditional) denoising diffusion probabilistic models (or, diffusion models) to generate realistic neural spiking data by modeling the inferred low-dimensional continuous representations.

Diffusion models [49, 20, 50] have been highly successful for conditional and unconditional data generation in several domains, including images [20], molecules [59], and audio spectrograms [27] and have demonstrated sampling-fidelity that outperforms that of VAEs and GANs [20]. A key strength of diffusion models that makes them particularly attractive in the context of modeling neural datasets is the ability to flexibly condition the generation on various (potentially complex) covariates, such as to simulate neural activity given certain behaviors. Recently, diffusion models have been extended to continuous neural time series such as local field potentials (LFPs) and electroencephalography (EEG) recordings [53]. However, due to the discrete nature of spiking data, standard diffusion models cannot be easily applied, thus excluding their use on many datasets in systems neuroscience.

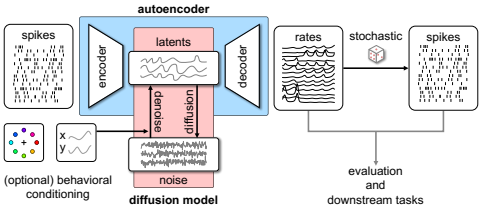


Figure 1: **Latent Diffusion for Neural Spiking data.** LDNS allows for (un)conditional generation of neural spiking data through combining a regularized autoencoder with diffusion models that act on the low-dimensional latent time series underlying neural population activity.

To bypass these limitations, LDNS employs a regularized autoencoder using structured state-space (S4) layers [18] to project the high-dimensional discrete spiking data into smooth, low-dimensional latents without making assumptions about the trial structure. We then train a diffusion model with S4 layers as a generative model of the inferred latents—akin to latent diffusion for images [45], where generation can be flexibly conditioned on behavioral covariates or task conditions.

A fundamental assumption of most low-dimensional latent variable models is that all statistical dependencies between observations are mediated by the latent space. However, in neural spiking data, there are prominent statistical dependencies that ought to persist *conditional* on the latent state, e.g., single-neuron dynamics such as refractory periods, burstiness, firing rate adaptation, or potential direct synaptic interactions. We show how such additional structure can be accounted for in LDNS, by equipping it with an expressive observation model [41, 32, 54, 62]: We use a Poisson model for spike generation with autoregressive couplings which are optimized post hoc to capture the temporal structure of single-neuron activity. This allows LDNS to capture a wide range of biological neural dynamics [55], with only a small additional computational cost.

Main contributions In summary, LDNS is a flexible method that allows for both high-fidelity diffusion-based sampling of neural population activity and access to time-aligned low-dimensional representations, which we validate on a synthetic dataset. Next, we show the utility and flexibility of this approach on complex real datasets: First, LDNS can handle variable-length spiking recordings from the human cortex. Second, LDNS can unconditionally generate faithful neural spiking activity recorded from monkeys performing a reach task. We demonstrate how LDNS can be equipped with an expressive autoregressive observation model that accounts for additional dependencies between data points (e.g., single neuron dynamics), increasing the realism of generated samples. Third, LDNS can generate realistic neural activity while conditioning on either reach direction or full reach trajectories (time series), including unseen behaviors that are then accurately decoded from the simulated neural data. Overall, LDNS enables simultaneous inference of low-dimensional latent representations for single-trial data interpretation and high-fidelity diffusion-based (conditional) generation of diverse neural spiking datasets, which will allow for closed-loop *in silico* experiments and hypothesis testing.

2 Methods

2.1 Latent Diffusion for Neural Spiking Data (LDNS)

We consider a dataset recorded from a population of n neurons, consisting of trials with spiking data $\mathbf{s} \in \mathbb{N}_0^{n \times T}$ (sorted into bins of fixed length resulting in spike counts over time), and optional simultaneously recorded behavioral covariates $\mathbf{y} \in \mathbb{R}^n$ (that can also be time-varying $\mathbf{y} \in \mathbb{R}^{n \times T}$). A dataset of M such trials $\mathcal{D}$ can be written as $\mathcal{D} = \{\mathbf{s}^{(i)}, \mathbf{y}^{(i)}\}$, possibly with varying trial lengths $T_1 \dots T_M$. We make the assumption that a large fraction of the variability in this dataset can be captured with a few underlying latent variables $\mathbf{z} \in \mathbb{R}^{d \times T}$, where $d < n$.

Our goal is to generate realistic spiking data $\mathbf{s}^*$ that faithfully capture both population-level and single-neuron dynamics of $\mathbf{s}_{1\dots T}$ with the ability to optionally condition the generation on behavior $\mathbf{y}_{\text{cond}}$. To this end, we propose a new method, LDNS, that combines the strength of neural dimensionality reduction approaches with that of diffusion-based generation.

LDNS uses a two-stage training framework, adopted from the highly successful family of latent diffusion models (LDMs) [45, 8, 59]. To train LDNS, we first train a regularized autoencoder [14] to compress the spiking data into a low-dimensional continuous latent space (Fig. 1). Concretely, we focus on two objects of interest for the LDNS autoencoder: **(1)** inferring a time-aligned, low-dimensional smooth representation $\mathbf{z} \in \mathbb{R}^{d \times T}$ that preserves the shared variability of the spiking data, and **(2)** predicting smooth firing rates λ that are most likely to give rise to the observed spiking data.

In the second stage, we train a diffusion model in latent space, possibly employing *conditioning* to make generation contingent on external (e.g., behavioral) covariates (Fig. 1). For the diffusion model, our main objective is the generation of $\mathbf{z}^* \in \mathbb{R}^{d \times T}$ that captures the distribution of inferred autoencoder latents. We also want the ability to sample latent trajectories of varying length.

In both stages, we use structured state-space (S4) [18] layers for modeling temporal dependencies. S4 layers consist of state-space transition matrices that can be unrolled into arbitrary-length convolution kernels, allowing sequence modeling of varying lengths. For details on network architectures and S4 layers, see appendix A1.

2.2 Regularized autoencoder for neural spiking data

For the spiking data, we choose a Poisson observation model, and train autoencoders by minimizing the Poisson negative log-likelihood of the input spikes $\mathbf{s}$ given the predicted rates $\lambda = \texttt{decoder}(\mathbf{z})$. To enforce smoothness in the latent space, where $\mathbf{z} = \texttt{encoder}(\mathbf{s})$, we add an L_2 regularization along with a temporal smoothness regularizer over $\mathbf{z}$, resulting in the combined loss

$$\mathcal{L}_{\text{AE}} = \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}\left[\sum_{\substack{i=1\\t=1}}^{n,T} \underbrace{(\lambda_i(t) - s_i(t)\ln\lambda_i(t))}_{\text{Poisson NLL}} + \beta_1 \underbrace{\|\mathbf{z}\|^2}_{L_2\text{ reg.}} + \beta_2 \sum_{\substack{k=1\\t=k+1}}^{K,T} \underbrace{\frac{\|\mathbf{z}(t) - \mathbf{z}(t-k)\|^2}{(1+k)}}_{\text{temporal smoothness}}\right]. \quad (1)$$

Code available at https://github.com/mackelab/LDNS.

To prevent the autoencoder from predicting highly localized Poisson rates, which have sharp peaks at input spike locations, we further regularize training using coordinated dropout [24], i.e., we randomly mask input spikes and compute the loss on the predicted rates at the masked locations (details in appendix A1.2).

Accounting for single-neuron dynamics with an expressive observation model So far, LDNS (like most latent variable models for neural data) uses a Poisson observation model, which assumes that all statistical dependencies are mediated by the latent state. To address this limitation and to capture dynamics and variability, which are "private" to individual neurons (such as refractory periods or burstiness), we propose to learn an autoregressive observation model. We make the predicted Poisson rates for each neuron i dependent also on recent spiking history, by including additional spike history couplings h_i [41, 32], resulting in the observation model

$$s_i(t) \sim \exp\left(\log \lambda_i(t) + h_{i,0} + \sum_{\tau=1}^{T'} h_{i,\tau} s_i(t-\tau)\right), \tag{2}$$

where T' corresponds to the time-lagged window length. This modification is learned post hoc, and the parameters h_i are fit with a maximum-likelihood objective (details in appendix A1.3). This approach does not alter the latent dynamics, while augmenting the model with single-neuron autoregressive dynamics. We observe that including spike history increases the realism of generated data and enables us to accurately capture single-neuron autocorrelation structures (Sec. 3.4).

2.3 Denoising Diffusion Probabilistic Models

In the second stage of training, we train diffusion models [20] to generate (conditional) samples from the distribution of inferred latents. The training dataset therefore contains autoencoder-derived latents of each trial, and optionally, additional conditioning information such as the corresponding behavior, i.e., $\mathcal{D}_z = \{\mathbf{z}^{(\mathbf{i})} = \texttt{encoder}(\mathbf{s}^{(i)}), \mathbf{y}^{(i)}\}$.

Diffusion models aim to approximate the data distribution $q(\mathbf{z})$ through an iterative denoising process starting from standard Gaussian noise. For latent $\mathbf{z}$ (denoted as $\mathbf{z}_0$ for diffusion timestep 0), we first produce a noised version at step t by adding Gaussian noise of the form $q(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\mathbf{z}_0, (1-\bar{\alpha}_t)I\right)$. Here, $\bar{\alpha}_t = \prod_{k=1}^{t} \alpha_k$, where the noise scaling factors $\alpha_1 \ldots \alpha_T$ follow a fixed linear schedule. We then train a neural network to approximate the reverse process $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t)$ for each diffusion timestep. The true (denoising) reverse transition $q(\mathbf{z}_{t-1}|\mathbf{z}_t)$ is intractable—however, we can apply variational inference to learn the *conditional* reverse transition $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0)$, which has a closed form written as

$$q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0) = \mathcal{N}\left(\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{z}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\mathbf{z}_0, \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}I\right). \tag{3}$$

We train the neural network $\mu_\theta(\mathbf{z}_t, t)$ to approximate the mean of this distribution by optimizing the loss $\mathbb{E}_{\mathbf{z}_0 \sim \mathcal{D}_z, \epsilon_0, t}\|\epsilon_\theta(\mathbf{z}_t, t) - \epsilon_0\|^2$, where ϵ_0 is the noise used to generate $\mathbf{z}_t$ from $\mathbf{z}_0$, and $\epsilon_\theta(\mathbf{z}_t, t)$ is the equivalent reparameterization for $\mu_\theta(\mathbf{z}_t, t)$. At test time, we sequentially sample $\mathbf{z}_{t-1}$ given $\mathbf{z}_t$ using the learned transition $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t)$, starting from standard Gaussian noise. Using S4 layers in the denoising network allows us to generate latents with varying lengths. This is achieved by unrolling the state transition matrix in the S4 layers to the desired length for each denoising step.

Diffusion models may be conditioned on fixed-length and time-varying covariates $\mathbf{y}$, in which case we learn the approximate reverse transition $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{y})$. Details on the conditioning mechanisms in appendix A1.4.

3 Experiments and Results

3.1 Datasets and tasks

We first evaluate the performance of LDNS on a synthetic spiking dataset where we have access to the ground-truth firing rates and latents. We choose the Lorenz attractor [29] as a low-dimensional, non-linear dynamical system commonly used in neuroscience [7, 36]. We simulate rates as an affine mapping from the 3-dimensional system to a 128-dimensional neural space, and sample from a

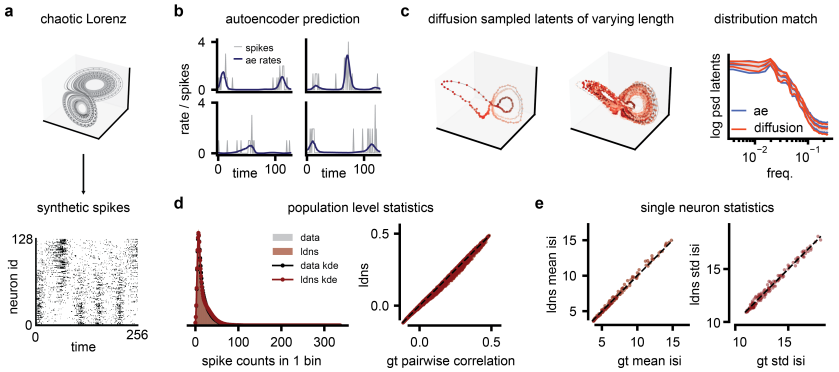


Figure 2: **Realistic generation of spiking data with underlying chaotic dynamics. a)** Synthetic spiking data from an underlying Lorenz system with a Poisson observation model. **b)** Accurate, smooth rate predictions of the autoencoder for held-out spiking data. **c)** Plotted trace of sampled latents (256 bins training length, left) and $16\times$ the original training length (middle). The sampled latent distribution matches the PSD of the autoencoder latents (right; median, 10%, and 90% percentiles). **d)** LDNS population spike count histogram (kde: kernel density estimate) and pairwise cross-correlations match the training distribution. **e)** LDNS single neuron statistics, i.e., mean inter-spike interval (isi) and std isi, match the training distribution.

Poisson distribution to generate spiking data. Next, we showcase the applicability of LDNS on two neural datasets: We apply our method on a highly complex dataset of human neural activity (128 units) recorded during attempted speech [56]. This dataset poses a challenge to many modeling approaches due to the different imagined sentences, resulting in variable lengths of the neural time series (between 2-10 seconds with a sampling rate of 50 Hz). Finally, we apply LDNS to model premotor cortical activity (182 units) recorded from monkeys performing a delayed center-out reach task with barriers [9, 38]. The multi-modal nature of the dataset allows us to assess both unconditional as well as conditional generation of neural spiking activity given monkey reach directions and entire velocity profiles of the performed reaches. See appendix A2,A3 for data and training details.

For the unconditional generation of monkey reach recordings (Sec. 3.4), we train both a Poisson observation model as well as a spike history-dependent autoregressive observation model. For all other experiments, we only train a Poisson observation model.

Baselines We compare LDNS to the most commonly known VAE-based latent variable model: Latent Factor Analysis via Dynamical Systems (LFADS [51, 36, 47]), which has been shown to outperform various classical latent variable models on a variety of tasks ([38], details in appendix A4). To ensure that we use optimal hyperparameters for LFADS, we follow the auto-ML pipeline proposed by Keshtkaran et al. [25]. This approach, termed AutoLFADS, has been shown to perform better than the original LFADS on benchmark tasks [38]. For the unconditional generation of monkey reach recordings, we further compared to additional VAE baselines [21, 62] (appendix A5).

Metrics For all experiments, we assess how well LDNS-generated samples match the spiking data in the training distribution. Concretely, we compare population-level statistics by computing 1) the distribution over the population spike count, which sums up all spikes co-occurring in the population in a single time bin (i.e., spike count histogram), and 2) pairwise correlations of LDNS samples and the spiking data for each pair of neurons. For single-neuron statistics, we compare 3) the mean and 4) standard deviation of the inter-spike-interval distribution for each neuron (mean isi and std isi). When multiple spikes occur in a single time bin, the spike times are distributed equally in this bin [12]. To further evaluate population dynamics, we compare the principal components of smoothed spikes.

3.2 LDNS captures the true spiking data distribution with an underlying Lorenz system

We simulate trials of length 256 timesteps from the three-dimensional (chaotic) Lorenz system (Fig. 2a). The regularized autoencoder extracts smooth latent time series (eight latent dimensions)

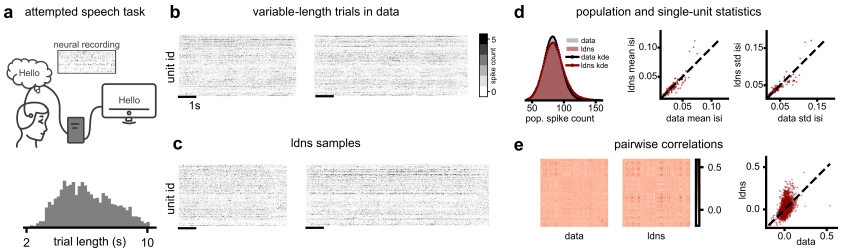


Figure 3: **Unconditional generation of variable-length trials of human spiking data during attempted speech. a)** Multi-unit activity is recorded from speech production-related regions of the brain (top) during attempted vocalization of variable-length sentences (bottom). **b)** Neural activity during sentences of different lengths. **c)** LDNS unconditionally sampled trials with different lengths, using the Poisson observation model. **d)** LDNS population spike count histogram, and mean and std of the isi match those of the data. **e)** Correlation matrices of the data (left) and LDNS samples (middle), and scatterplot of the pairwise correlations of data vs. LDNS samples (right).

from the 128-dimensional spiking data, resulting in smooth firing rate predictions that closely match the ground-truth rates (Fig. 2b, Supp. Fig. A2,A3). We then train a diffusion model on the extracted autoencoder latents. Latents sampled from the diffusion model (red) preserve the attractor geometry of the Lorenz system (Fig. 2c, left, three of the eight latent dimensions), indicating that LDNS preserves a meaningful latent space. The architectural choice of S4 layers allows for length generalization: although we train on time segments of 256-time steps, we can sample and successfully generate latent trajectories that are much longer, but still accurately reflect the Lorenz dynamics (Fig. 2c, middle, $16\times$ longer generation). In comparison, LFADS exhibits instabilities when generating such longer sequences (appendix A6.1). Overall, the latent time series distribution is captured well by the diffusion model, with matching power spectral densities (PSD) per latent dimension (Fig. 2c, right, other dimensions in Supp. Fig. A3).

To assess the sampling fidelity of the generated synthetic neural activity, we compute a variety of spike statistics frequently used in neuroscience. LDNS captures both population-level statistics, such as the population spike count histogram and pairwise correlations between neurons (Fig. 2d), as well as single-neuron statistics, quantified by the mean and standard deviation of inter-spike-intervals (Fig. 2e). LDNS also captures the temporal correlation structure of the data (Supp. Fig. A4). These results demonstrate that LDNS can both perform inference of low-dimensional latents and provide high-fidelity diffusion-based generation that perfectly captures the statistics of the ground-truth synthetic data.

3.3 Modeling variable-length trials of neural activity recorded in human cortex

Next, we assess whether LDNS is capable of capturing real electrophysiological data, applying it to neural recordings from human cortex during attempted speech (Fig. 3a, top, Willett et al. [56]). A participant with a degenerative disease who is unable to produce intelligible speech attempts to vocalize sentences prompted on a screen, while neural population activity is recorded from the ventral premotor cortex. Since there is a large variation in the length of prompted sentences (Fig. 3a, bottom), this dataset allows us to evaluate the performance of LDNS on real data in naturalistic settings with variable-length and highly heterogeneous dynamics.

To account for varying trial length during autoencoder training, we pad all trials to a maximum length of 512 bins and compute the reconstruction loss only on the observed time bins. For the diffusion model, we indicate the target trial length with a binary mask as a conditioning variable.

This approach allows us to infer time-aligned latents underlying the cortical activity of the participants, compressing the population activity by a factor of four before training an unconditional diffusion model on these latents. Resulting samples of LDNS, mimicking human cortical activity, are visually indistinguishable from the real data (Fig. 3b,c, additional samples in Supp. Fig. A7). This is reflected in closely matched population spike count histograms (Fig. 3d, left), and single neuron statistics such as mean and standard deviation of the inter-spike interval (Fig. 3d, right). Additionally, real

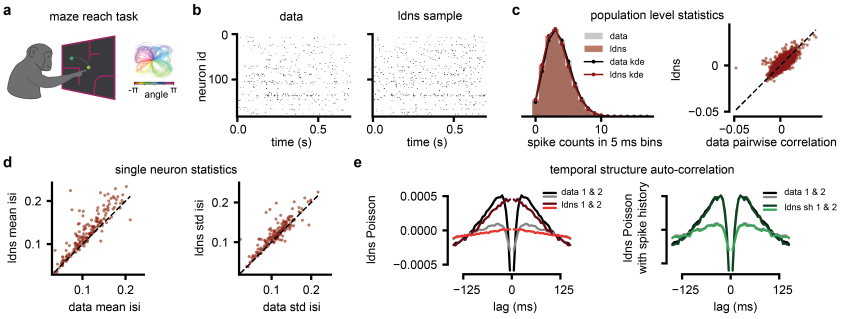


Figure 4: **Realistic generation of spiking data in a monkey performing reach tasks. a)** A monkey performs diverse reach movements in different mazes. **b)** Neural activity during a reach trial and a sampled trial from LDNS with a Poisson observation model. **c)** The LDNS population spike count histogram, and pairwise correlations match those of the data. **d)** LDNS mean- and std isi match the monkey data distribution. **e)** Auto-correlation of data, LDNS samples with Poisson observations (left), and LDNS samples with spike history, grouped according to correlation strength.

and LDNS-sampled spikes display similar population dynamics, as reflected in the top principal components (Supp. Fig. A8). While LDNS tends to overestimate some pairwise correlations, it captures prominent features of the correlation structure in the data (Fig. 3e, Pearson correlation coefficient $r = 0.47$), and our analysis indicates that this slight mismatch already arises at the autoencoder stage (Supp. Fig. A9).

LDNS allows for both inferring latent representations and generating variable-length trial data, making it applicable to complex real neural datasets without a fixed trial structure.

3.4 Realistic generation of spiking data from a monkey performing reach tasks

We further evaluate LDNS in a different setting by applying it to model sparse spiking data recorded from a monkey performing a reaching task constrained by barriers that form a maze (Fig. 4a, left). The variety of different maze architectures leads to diverse reach movements of both curved and straight reaches (Fig. 4a, right). We again infer low-dimensional latent trajectories that capture the shared variability of the neural population and then train an unconditional diffusion model on these latents. Sampled spikes from LDNS closely resemble the true, sparse population data (Fig. 4b, additional samples in Supp. Fig. A11), and closely match population-level spike statistics (Fig. 4c). Single neuron statistics in this low spike count regime (a maximum of three spikes per neuron in 5 ms bins) are also captured well (Fig. 4d), and are on par with or better than LFADS [36, 25] (see Table 1 for summary of main comparisons, and appendix A5 for additional baselines [21, 62]). Beyond spiking statistics, we observe that LDNS also preserves the temporal structure of population dynamics, as reflected in the top principal components of smoothed spikes (Supp. Fig. A15). Thus, LDNS can generate spiking data that is faithful at the level of both single-neuron and population dynamics.

Table 1: **Model metrics comparison.** D_{KL} for the population spike count histogram and RMSE comparisons. Mean and standard deviation across 5 folds sampled with replacement. **sh** represents observation models with spike history. **Bolded** entries represent best-performing values for Poisson and spike-history observation models.

Method	D_{KL} psch	RMSE pairwise corr	RMSE mean isi	RMSE std isi
AutoLFADS	$0.0040 \pm 2.2\text{e-}4$	$0.0026 \pm 1.25\text{e-}5$	0.039 ± 0.003	0.029 ± 0.001
LDNS	$\mathbf{0.0039 \pm 3.9\text{e-}4}$	$\mathbf{0.0025 \pm 1.1\text{e-}5}$	$\mathbf{0.037 \pm 0.001}$	$\mathbf{0.023 \pm 0.001}$
AutoLFADSsh	$0.0036 \pm 2.1\text{e-}4$	$0.0026 \pm 1.8\text{e-}5$	0.034 ± 0.002	$\mathbf{0.023 \pm 0.0001}$
LDNSsh	$\mathbf{0.0016 \pm 6.2\text{e-}4}$	$\mathbf{0.0025 \pm 1.07\text{e-}5}$	$\mathbf{0.024 \pm 0.002}$	$\mathbf{0.023 \pm 0.001}$

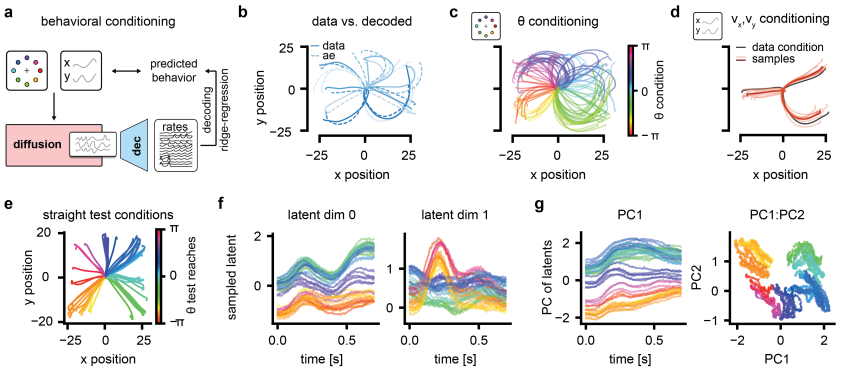


Figure 5: **Generation conditioned on monkey reach directions and velocity traces. a)** Closed loop assessment: do conditionally generated latents translate to neural activity consistent with the desired direction or reach movement? **b)** Unseen reach movements (data) and corresponding movements decoded from the rates predicted by the autoencoder (ae). **c)** Decoded reach directions of LDNS samples conditioned on initial reach angles θ. **d)** Decoded reach directions of LDNS samples conditioned on 3 unseen reach movements (velocities v_x, v_y). **e)** Straight reaches from the test set used for velocity conditioning. **f)** LDNS sampled latents conditioned on trajectories shown in e) vary smoothly over time and reflect information about reach angles. **g)** PCs of sampled LDNS latents shown in f) reveal meaningful and separable information about behavior.

Both LFADS and the LDNS autoencoder are optimized by maximizing the Poisson log-likelihood, and thus cannot capture single-neuron dynamics such as refractoriness [41, 55], which can have a strong influence on the observed autocorrelation structure. Given the overall sparsity of the spiking data and resulting low correlations (Supp. Fig. A10), we focus on the temporal structure of auto-correlations averaged within groups of neurons (≈ 45 neurons per group) split by their instantaneous correlation strength [32]: darker colors correspond to the highest correlated group of four, lighter colors correspond to the group with the second highest correlations (Fig. 4e). We then compare these auto-correlations to those of grouped LDNS samples with a Poisson observation model (red). As expected, LDNS with Poisson observations is unable to capture the dip in the data auto-correlation at 5 ms lags (one time bin) (Fig. 4e, left).

To overcome this mismatch, we train an additional spike history-dependent autoregressive observation model on top of the inferred rates (LDNSsh, for spike history). In contrast to the Poisson samples, autoregressive samples can capture this aspect of neural spiking data very accurately while also improving the overall fit to the empirical auto-correlation (Fig. 4e, right). Moreover, the post hoc optimization of these filters also improves modeling of other single-neuron, as well as population-level statistics, such as the population spike count histogram or the mean of the isi (Table 1, Supp. Fig. A13).

We view this post-hoc augmentation as a key modular contribution, which can be flexibly applied to other generative models. To this end, we extend AutoLFADS with spike history dependence (LFADSsh), improving its performance across metrics. The augmented LFADSsh also captures the dip in autocorrelation at 5 ms lags (Supp. Fig. A12). Still, in both observation model variants, LDNS maintains superior or comparable performance (Table 1).

Thus, Poisson LDNS allows for the generation of spiking data that is on par or better in terms of sampling fidelity than previous approaches. Incorporating spike-history dependence and sampling spikes autoregressively allows us to further increase the realism of generated spike trains, leading to a large improvement on several of the considered metrics.

3.5 Conditional generation of neural activity given reach directions or velocity profiles

Lastly, we assess the ability of conditional LDNS to generate realistic neural activity conditioned on behavioral covariates of varying complexity: the reach angle or entire velocity time series (Fig. 5a). We first validate that the autoencoder predicts firing rates that allow us to linearly decode the behavior

following the ridge-regression approach proposed in [38]. Decoded behavior from autoencoder reconstructed rates matches the true trajectories of unseen test trials (Fig. 5b).

Given that the autoencoder performs adequately, we then test the ability to generate neural time series conditioned on the initial reach angle performed by the monkey θ_{reach}. Indeed, from the generated samples of neural activity, we can decode—using the same linear decoder—realistic reach kinematics that are consistent with the conditioning angle θ_{reach} and overall reach kinematics (Fig. 5c). This indicates that LDNS can generate realistic neural activity consistent with a queried reach direction.

An even more challenging task that is intriguing for hypothesis generation is the ability to mimic an entire experiment and ask what the neural activity *would have looked like* if the monkey had performed a particular hypothetical movement. To this end, we train a diffusion model on the same autoencoder-inferred latents but now condition on entire velocity traces (Fig. 5d). Velocity-conditioned LDNS is able to produce different samples of neural activity that are consistent with, but not exact copies of, the reach trajectories of the held-out trials given as the conditioning covariate. Such closed-loop conditioning experiments open the possibility of making predictions about neural activity during desired unseen behaviors, and thus make experimentally testable predictions.

Finally, to understand how LDNS incorporates behavioral information, we analyzed latent trajectories that were conditionally sampled based on straight reach movements in different directions (Fig. 5e). Individual samples of latent trajectories vary smoothly within a trial (Fig. 5f), while reach direction varies smoothly across samples in the first principal component (PC1) of the latents (Fig. 5g, left). Projection onto the first two PCs of latent trajectories shows clear clustering by reach direction (Fig. 5g, right), and we show that such clustering arises already at the autoencoder stage (Supp. Fig. A17).

In summary, LDNS not only produces faithful spiking samples but also allows for flexible conditioning. Furthermore, LDNS learns an interpretable latent space with behaviorally-relevant structure.

4 Related Work

Latent variable models of neural population dynamics LDNS builds on previous LVMs in neuroscience, which have been extensively applied to infer low-dimensional latent representations of neural spiking data [61, 32, 39, 58, 62, 13, 28, 63] (see [38] for a comprehensive list.) In addition to capturing shared population-level dynamics and dependence on external stimuli [4], LVMs have been extended to allow autoregressive neuron-level (non-Poisson) dynamics [32, 13, 62] or even direct neural interactions [54]. While these methods often have useful inductive biases (e.g., linear dynamical systems [32, 28] or Gaussian process priors [61]), these models are typically not expressive enough to yield realistic neural samples across a range of conditions.

Deep LVMs and other deep learning-based approaches Variational autoencoders (VAEs) [26] are particularly popular in neuroscience as they allow us to infer low-dimensional dynamics underlying high-dimensional discrete data [63, 16, 46], especially when combined with nonlinear recurrent neural networks [36, 21]. VAEs have been used to infer identifiable low-dimensional latent representations conditioned on behavior [63, 21] and have incorporated smoothness priors using Gaussian Processes to regularize the latent space [16]. However, the generation performance of VAEs is rarely explored in neuroscience. Besides VAEs, generative adversarial networks (GANs [17]) have been proposed to synthesize spiking neural population activity [34, 42]. While GANs produce high-fidelity samples, they are challenging to train reliably and lack a low-dimensional latent space. More recently, transformer-based architectures have also been adapted to model neural activity [5, 60], though often with the focus of accurate decoding of behavior instead of generation of realistic spiking samples, while also lacking an explicit latent space [3]. Lastly, deterministic approaches utilizing RNNs for dynamical systems reconstruction also target low-dimensional latent dynamics underlying neural data [19], but they do not act as probabilistic generative models.

Diffusion models LDNS leverages recent advances in diffusion models, which have become state-of-the-art for high-fidelity generation in several domains [20, 27], including continuous-valued neural signals such as EEG [53], as well as in time series forecasting and imputation tasks [2, 52, 43]. Similar to the LDNS architecture, Alcaraz and Strodthoff [2] also use an S4-based denoiser for imputation. More specifically, LDNS is inspired by latent diffusion models [45, 27, 59, 15], which benefit from operating on the latent space of an autoencoder and flexible conditioning mechanisms to generate samples based on a given covariate, as is done with text-to-image [45] and other cross-

modality scenarios. Conveniently, this allows LDNS to bypass the challenges of directly modeling discrete-valued spiking data, by instead transforming spikes into the continuous latent space.

5 Summary and discussion

We here proposed LDNS, a flexible generative model of neural spiking recordings that simultaneously infers low-dimensional latent representations *and* generates realistic neural activity conditioned on behavioral covariates. We apply LDNS to model three different datasets: synthetic data simulated from chaotic Lorenz dynamics, human cortical recordings with heterogeneous and variable-length trials, and finally, neural recordings in monkeys performing reach actions in a maze. Through our experiments, we demonstrate how several features of LDNS are beneficial for modeling complex datasets in neuroscience:

First, following other LDMs in the literature, LDNS decouples latent inference and probabilistic modeling of the data, offering flexibility in reusing the trained autoencoder and diffusion model. For the monkey recordings, all diffusion models (unconditional, conditioned on reach angle, and conditioned on hand velocities) operate in the latent space of the same autoencoder, in contrast to existing approaches that require end-to-end retraining for each type of conditioning variable. LDNS is also faster to train than AutoLFADS, which requires population-based training to optimize hyperparameters (appendix A3.1). Second, we show that LDNS autoencoders can be augmented with per-neuron autoregressive dynamics to capture single-neuron temporal dynamics (e.g., refractoriness), which otherwise cannot be captured with population-level shared dynamics. Third, as a result of the length-generalizable autoencoders and diffusion models using S4 layers, LDNS can generate variable-length trials in both the Lorenz example and human cortical recordings—a feature that will be particularly useful in modeling datasets recorded during naturalistic stimuli or behavior.

Altogether, these features enable LDNS to generate realistic neural activity, especially when conditioned on behavioral covariates. In our experiments, we demonstrate that unseen movement trajectories can be used to conditionally generate samples of neural activity, from which we can decode these hypothetical behaviors. These generated latent trajectories reflect behavioral information in an interpretable way. Our methodology is general and can be applied to recordings from any brain region, beyond the motor and speech cortex examples shown here. Thus, LDNS opens up further possibilities for hypothesis generation and testing *in silico*, potentially enabling stronger links between experimental and computational works.

Limitations In real neural data, the latent dimensionality of the system is not known, and as with all LVMs (which often assume that population dynamics are intrinsically low-dimensional), choosing an appropriate latent dimension can be challenging. Furthermore, any modeling errors at the encoding and decoding stage of the autoencoder will affect the overall performance of the latent diffusion approach. Nevertheless, in our experiments, we found that autoencoder training is fast, stable, and reasonably robust to hyperparameter configurations. While LDNS was still able to model the data well under relatively severe compression (e.g., 182-to-16 for the monkey recordings), optimizing latent dimensionality to balance expressiveness and interpretability remains a goal for future research.

Broader impact Realistic spike generation capabilities increase the risk of research manipulation by generating synthetic data that may be difficult to detect. On the other hand, LDNS could be useful for the dissemination of privatized clinical data, though we acknowledge the critical importance of protecting data privacy when working with sensitive human participant data. Finally, synthetically generated data (conditioned on unseen behavioral conditions) could be useful for augmenting the training of brain-computer interface decoding models.

Acknowledgements

This work was supported by the German Research Foundation (DFG) through Germany's Excellence Strategy (EXC-Number 2064/1, PN 390727645) and SFB1233 (PN 276693517), SFB 1089 (PN 227953431), SPP2041 (PN 34721065), the German Federal Ministry of Education and Research (Tübingen AI Center, FKZ: 01IS18039), the Human Frontier Science Program (HFSP), and the European Union (ERC, DeepCoMechTome, 101089288). We utilized the Tübingen Machine Learning Cloud, supported by DFG FKZ INST 37/1057-1 FUGG. JK, AS, and JV are members of the International Max Planck Research School for Intelligent Systems (IMPRS-IS) and JV is supported

by the AI4Med-BW graduate program. We thank Chethan Pandarinath for providing access to their compute cluster to train AutoLFADS. We thank Christian F. Baumgartner and all Mackelab members for feedback and discussions. We would like to also thank our reviewers for their insightful comments which improved our paper.

References

[1] Misha B. Ahrens, Michael B. Orger, Drew N. Robson, Jennifer M. Li, and Philipp J. Keller. Whole-brain functional imaging at cellular resolution using light-sheet microscopy. *Nature Methods*, May 2013.

[2] Juan Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Transactions on Machine Learning Research*, 2022.

[3] Antonis Antoniades, Yiyi Yu, Joe S. Canzano, William Yang Wang, and Spencer Smith. Neuroformer: Multimodal and Multitask Generative Pretraining for Brain Data. In *International Conference on Learning Representations*, October 2024.

[4] Evan W Archer, Urs Koster, Jonathan W Pillow, and Jakob H Macke. Low-dimensional models of neural population activity in sensory cortical circuits. *Advances in neural information processing systems*, 27, 2014.

[5] Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael Mendelson, Blake Richards, Matthew Perich, Guillaume Lajoie, and Eva Dyer. A Unified, Scalable Framework for Neural Population Decoding. *Advances in Neural Information Processing Systems*, 2023.

[6] Mohammad Bashiri, Edgar Walker, Konstantin-Klemens Lurz, Akshay Jagadish, Taliah Muhammad, Zhiwei Ding, Zhuokun Ding, Andreas Tolias, and Fabian Sinz. A flow-based latent state generative model of neural population responses to natural images. In *Advances in Neural Information Processing Systems*, 2021.

[7] Manuel Brenner, Florian Hess, Georgia Koppe, and Daniel Durstewitz. Integrating multimodal data for joint generative modeling of complex dynamics, 2024.

[8] Zijiao Chen, Jiaxin Qing, Tiange Xiang, Wan Lin Yue, and Juan Helen Zhou. Seeing beyond the brain: Conditional diffusion model with sparse masked modeling for vision decoding. *Computer Vision and Pattern Recognition*, 2023.

[9] Mark Churchland and Matthew Kaufman. Mcmaze: macaque primary motor and dorsal premotor cortex spiking activity during delayed reaching (version 0.220113.0400) [data set], 2022. Version 0.220113.0400.

[10] Mark M. Churchland, John P. Cunningham, Matthew T. Kaufman, Justin D. Foster, Paul Nuyujukian, Stephen I. Ryu, and Krishna V. Shenoy. Neural population dynamics during reaching. *Nature*, 2012.

[11] John P Cunningham and Byron M Yu. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 2014.

[12] Kevin Doran, Marvin Seifert, Carola A. M. Yovanovich, and Tom Baden. Distance function for spike prediction, 2023.

[13] Lea Duncker and Maneesh Sahani. Temporal alignment and latent gaussian process factor inference in population spike trains. In *Advances in Neural Information Processing Systems*, 2018.

[14] Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Scholkopf. From variational to deterministic autoencoders. In *International Conference on Learning Representations*, 2020.

[15] Karan Goel, Albert Gu, Chris Donahue, and Christopher R'e. It's raw! audio generation with state-space models. *International Conference on Machine Learning*, 2022.

[16] Rabia Gondur, Usama Bin Sikandar, Evan Schaffer, Mikio Christian Aoi, and Stephen L. Keeley. Multi-modal Gaussian Process Variational Autoencoders for Neural and Behavioral Data. In *International Conference on Learning Representations*, 2024.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

[18] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *International Conference on Learning Representations*, 2022.

[19] Florian Hess, Zahra Monfared, Manuel Brenner, and Daniel Durstewitz. Generalized teacher forcing for learning chaotic dynamics. *International Conference for Machine Learning*, 2023.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, 2020.

[21] Cole Hurwitz, Akash Srivastava, Kai Xu, Justin Jude, Matthew Perich, Lee Miller, and Matthias Hennig. Targeted neural dynamical modeling. *Advances in Neural Information Processing Systems*, 2021.

[22] Kristopher Jensen, Ta-Chu Kao, Jasmine Stone, and Guillaume Hennequin. Scalable Bayesian GPFA with automatic relevance determination and discrete noise models. In *Advances in Neural Information Processing Systems*, 2021.

[23] James J. Jun, Nicholas A. Steinmetz, Joshua H. Siegle, et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 2017.

[24] Mohammad Reza Keshtkaran and Chethan Pandarinath. Enabling hyperparameter optimization in sequential autoencoders for spiking neural data. *Advances in neural information processing systems*, 32, 2019.

[25] Mohammad Reza Keshtkaran, Andrew R. Sedler, Raeed H. Chowdhury, Raghav Tandon, Diya Basrai, Sarah L. Nguyen, Hansem Sohn, Mehrdad Jazayeri, Lee E. Miller, and Chethan Pandarinath. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *Nature Methods*, 2022.

[26] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014. doi: 10.48550/arXiv.1312.6114.

[27] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *International Conference on Learning Representations*, 2021.

[28] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

[29] Edward N. Lorenz. Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, March 1963.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations*, 2019.

[31] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. doi: arXiv:1711.05101.

[32] Jakob H Macke, Lars Buesing, John P Cunningham, Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. In *Advances in Neural Information Processing Systems*, 2011.

[33] Bartul Mimica, Tuçe Tombaz, Claudia Battistin, Jingyi Guo Fuglstad, Benjamin A. Dunn, and Jonathan R. Whitlock. Behavioral decomposition reveals rich encoding structure employed across neocortex in rats. *Nature Communications*, 2023.

[34] Manuel Molano-Mazon, Arno Onken, Eugenio Piasini, and Stefano Panzeri. Synthesizing realistic neural population activity patterns using generative adversarial networks. *International Conference on Learning Representations*, 2018.

[35] Joseph E. O'Doherty, Mariana M. B. Cardoso, Joseph G. Makin, and Philip N. Sabes. Nonhuman Primate Reaching with Multichannel Sensorimotor Cortex Electrophysiology, 2017.

[36] Chethan Pandarinath, Daniel J. O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D. Stavisky, Jonathan C. Kao, Eric M. Trautmann, Matthew T. Kaufman, Stephen I. Ryu, Leigh R. Hochberg, Jaimie M. Henderson, Krishna V. Shenoy, L. F. Abbott, and David Sussillo. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, 15, 2018.

[37] William S. Peebles and Saining Xie. Scalable diffusion models with transformers. *IEEE International Conference on Computer Vision*, 2022.

[38] Felix C Pei, Joel Ye, David M. Zoltowski, Anqi Wu, Raeed Hasan Chowdhury, Hansem Sohn, Joseph E O'Doherty, Krishna V. Shenoy, Matthew Kaufman, Mark M Churchland, Mehrdad Jazayeri, Lee E. Miller, Jonathan W. Pillow, Il Memming Park, Eva L Dyer, and Chethan Pandarinath. Neural latents benchmark '21: Evaluating latent variable models of neural population activity. In *Neural Information Processing Systems, Datasets and Benchmarks Track (Round 2)*, 2021.

[39] Biljana Petreska, Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Dynamical segmentation of single trials from population neural data. In *Advances in Neural Information Processing Systems*, 2011.

[40] David Pfau, Eftychios A Pnevmatikakis, and Liam Paninski. Robust learning of low-dimensional dynamics from large neural ensembles. In *Advances in Neural Information Processing Systems*, 2013.

[41] Jonathan W. Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M. Litke, E. J. Chichilnisky, and Eero P. Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454:995–999, 2008.

[42] Poornima Ramesh, Mohamad Atayi, and Jakob H Macke. Adversarial training of neural encoding models on population spike trains. *Real Neurons & Hidden Units: Future directions at the intersection of neuroscience and artificial intelligence @ NeurIPS 2019*, 2019.

[43] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. *International Conference on Machine Learning*, 2021.

[44] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[46] Auguste Schulz, Julius Vetter, Richard Gao, Daniel Morales, Victor Lobato-Rios, Pavan Ramdya, Pedro J. Gonçalves, and Jakob H. Macke. Modeling conditional distributions of neural and behavioral data with masked variational autoencoders. *bioRxiv*, 2024.

[47] Andrew R. Sedler and Chethan Pandarinath. lfads-torch: A modular and extensible implementation of latent factor analysis via dynamical systems. *arXiv preprint arXiv:2309.01230*, 2023.

[48] Nicholas James Sofroniew, Daniel Flickinger, Jonathan King, and Karel Svoboda. A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging. *eLife*, 2016.

[49] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.

[50] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, 2021.

[51] David Sussillo, Rafal Jozefowicz, L. F. Abbott, and Chethan Pandarinath. Lfads - latent factor analysis via dynamical systems, 2016.

[52] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 2021.

[53] Julius Vetter, Jakob H Macke, and Richard Gao. Generating realistic neurophysiological time series with denoising diffusion probabilistic models. *bioRxiv*, 2023.

[54] Michael Vidne, Yashar Ahmadian, Jonathon Shlens, Jonathan W. Pillow, Jayant Kulkarni, Alan M. Litke, E. J. Chichilnisky, Eero Simoncelli, and Liam Paninski. Modeling the impact of common noise inputs on the network activity of retinal ganglion cells. *Journal of Computational Neuroscience*, 2012.

[55] Alison I. Weber and Jonathan W. Pillow. Capturing the Dynamical Repertoire of Single Neurons with Generalized Linear Models. *Neural Computation*, 2017.

[56] Francis Willett et al. Data for: A high-performance speech neuroprosthesis [dataset], 2023. URL `https://doi.org/10.5061/dryad.x69p8czpq`.

[57] Francis R. Willett, Erin M. Kunz, Chaofei Fan, Donald T. Avansino, Guy H. Wilson, Eun Young Choi, Foram Kamdar, Matthew F. Glasser, Leigh R. Hochberg, Shaul Druckmann, Krishna V. Shenoy, and Jaimie M. Henderson. A high-performance speech neuroprosthesis. *Nature*, 2023. Publisher: Nature Publishing Group.

[58] Anqi Wu, Nicholas A Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in neural information processing systems*, 30, 2017.

[59] Minkai Xu, Alexander Powers, R. Dror, Stefano Ermon, and J. Leskovec. Geometric latent diffusion models for 3d molecule generation. *International Conference on Machine Learning*, 2023.

[60] Joel Ye, Jennifer Collinger, Leila Wehbe, and Robert Gaunt. Neural Data Transformer 2: Multi-context Pretraining for Neural Spiking Activity. *Advances in Neural Information Processing Systems*, 2023.

[61] Byron M. Yu, John P. Cunningham, Gopal Santhanam, Stephen I. Ryu, Krishna V. Shenoy, and Maneesh Sahani. Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity. *Journal of Neurophysiology*, 2009.

[62] Yuan Zhao and Il Memming Park. Variational latent gaussian process for recovering single-trial dynamics from population spike trains. *Neural Computation*, 2017.

[63] Ding Zhou and Xue-Xin Wei. Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-VAE. In *Advances in Neural Information Processing Systems*, 2020.

Appendix

A1 LDNS architecture

Here we describe the exact network components and architecture for the autoencoder and diffusion model.

A1.1 Structured State-Space Layers (S4)

A central component of our autoencoder architecture is the recently introduced structured state space models (S4)[18]. With an input sequence $\mathbf{x} = [x_1 \dots x_T] \in \mathbb{R}^T$ and corresponding output $\mathbf{y} = [y_1 \dots y_T] \in \mathbb{R}^T$, an S4 layer applies the following operation for each timestep –

$$\begin{aligned} s_t &= \overline{A} s_{t-1} + \overline{B} x_t \\ y_t &= C s_t, \end{aligned} \tag{4}$$

where the discretized state and input matrices $\overline{A}, \overline{B}$ given continuous analogues A, B and step size Δ are computed as

$$\begin{aligned} \overline{A} &= (I - \Delta/2 \cdot A)^{-1}(I + \Delta/2 \cdot A) \\ \overline{B} &= (I - \Delta/2 \cdot A)^{-1}\Delta B. \end{aligned} \tag{5}$$

When the state s_t is not required, this recurrent computation of the output $\mathbf{y}$ given input sequence $\mathbf{y}$ can be unrolled into a parallelizable convolution operation

$$\begin{aligned} \mathbf{y} &= K * \mathbf{x}, \text{ with unrolled kernel} \\ K &= (C\overline{B}, C\overline{AB}, \dots, C\overline{A}^{T-1}\overline{B}). \end{aligned} \tag{6}$$

We used the S4 implementation[1] provided by Gu et al. [18] that stably initializes the state transition matrix A using a diagonal plus low-rank approximation. For a multivariate input-output pair $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{D \times T}$, we apply D separate univariate S4 layers for each dimension and then mix them in the channel-mixing layer using an MLP (see next section). Each univariate input-output mapping consists of H separate S4 "head" that are expanded and contracted from and into a single dimension.

Due to its recurrent nature, S4 is a causal layer, enabling variable-length training and inference. To enable bidirectionality, we flip the input signal $\mathbf{x}$ in time, apply $H/2$ S4 heads each for the flipped and unflipped signal, and then combine these at the end into univariate signal $\mathbf{y}$. This allows bidirectional flow of information from front-to-back and back-to-front of the signal.

A1.2 Autoencoder

We include temporal information only in the encoder and model the decoder as a lightweight pointwise MLP for the autoencoder (Supp. Fig. A1). This allows us to temporally align the latents with the signal, and ensure that no further temporal dynamics are introduced when mapping the latents back into Poisson rates.

We use causal S4 layers, allowing length generalization and handling of variable-length signals. During training, we pad the input spiking data with zeros into a fixed length and only backpropagate through the unpadded output rates.

Furthermore, to infer smooth rates and avoid spiking behavior, we use coordinated dropout [24]. For each time bin independently, we mask the input spikes to zero with random probability p and scale up the remaining spikes by $\frac{1}{1-p}$ (this preserves the firing statistics of the spiking data). We then backpropagate through the Poisson NLL loss only over the masked positions, effectively preventing the network from collapsing to a spiking prediction of the Poisson rates.

[1] Github link: https://github.com/state-spaces/s4/blob/main/models/s4/s4.py

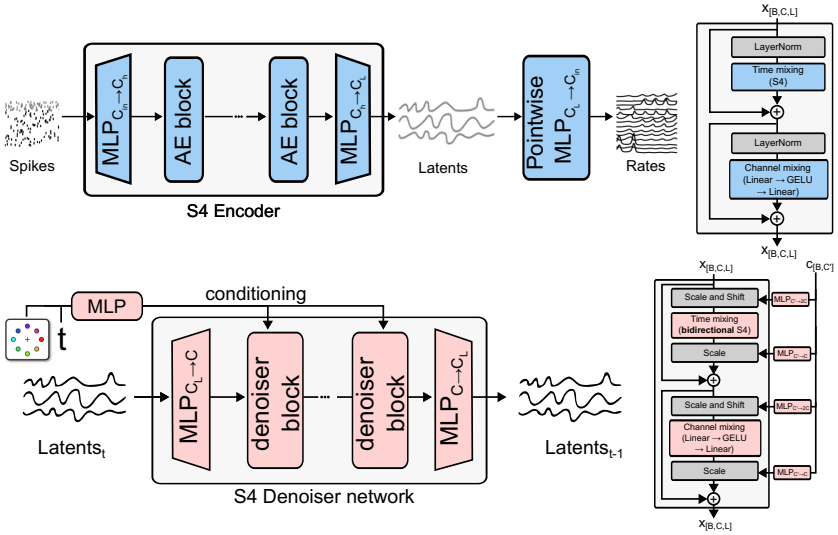


Figure A1: (Top left) The S4 autoencoder architecture. (Top right) Architecture for the autoencoder blocks used in the encoder. (Bottom left) The S4 diffusion model architecture. (Bottom right) Architecture for the diffusion blocks.

A1.3 Spike-history-augmented Poisson model

The parameters of the autoregressive observation model in Eq. (2) are learned by maximizing the Poisson log-likelihood. Training is performed jointly for all neurons with a history length of $T' = 20$, corresponding to 100 ms, using the AdamW optimizer [30] (learning rate 0.1, weight decay 0.01). In our implementation, we use the Softplus function given by $\mathrm{Softplus}(x) = \log(1 + \exp(x))$ as an approximation to the exponential function in Eq. (2), which is accurate for the low-count regime while increasing numerical stability. During autoregressive sampling, we limit the maximum possible spike count to 5 spikes, which corresponds to the biological maximum, limited by the refractory period for 5 ms time bins.

A1.4 Diffusion model

We consider four variants of our proposed diffusion model with time-mixing and channel-mixing layers for four different tasks. In all cases, except for the conditioning mechanism, the internal architecture remains the same (Supp. Fig. A1). For diffusion timestep and fixed-length conditioning vector, we shift and scale the inputs and outputs to the time mixing and channel mixing blocks using adaptive instance normalization, as done in Peebles and Xie [37].

1. Unconditional generation for synthetic spiking data with Lorenz Dynamics and cortical spiking data in monkeys – we use only time conditioning.
2. Angle-conditioned generation for cortical monkey spiking data – we add an embedding $\mathrm{MLP}([\cos\theta, \sin\theta])$ of the reach angle θ to the timestep embedding output.
3. Trajectory conditioned generation for cortical monkey spiking data – we concatenate the hand velocities v_x, v_y of the monkey with the input as two additional channels.
4. Unconditional variable-length generation for cortical human spiking data – we concatenate the desired length (with a maximum sequence length of 512) as a centered binary mask channel in the input. We only backpropagate through the central section of the output corresponding to the binary mask.

We use a DDPM scheduler with 1000 timesteps and ϵ-parameterization. To stabilize and speed up training, we train all diffusion models using a smooth L_1 loss, written as

$$L(x;\delta) = \begin{cases} x^2/(2\delta) & \text{if } |x| < \delta \\ |x| - \delta & \text{otherwise,} \end{cases} \quad \text{with } \delta = 0.05. \tag{7}$$

A2 Dataset access information

All real-world datasets used in this work are publicly available under open-access licenses. Our work does not involve the collection of new experimental data.

The human BCI dataset is available at `https://datadryad.org/stash/downloads/file_stream/2547369` under a CC0 1.0 Universal Public Domain Dedication license. This dataset was originally published in Willett et al. [57]. The data was collected under appropriate ethical oversight, with approval from the Institutional Review Board at Stanford University (protocol #20804).

The monkey reaching dataset (MC_Maze) is available through the DANDI Archive (`https://dandiarchive.org/dandiset/000128`, ID: 000128) under a CC-BY-4.0 license. This dataset contains sorted unit spiking times and behavioral data from primary motor and dorsal premotor cortex during a delayed reaching task.

A3 Hyperparameters and compute resources

Table 2: Training details for autoencoder models on Lorenz, Monkey reach, and Human BCI datasets. We used the AdamW [31] optimizer, whose learning rate was linearly increased over in the initial period and then decayed to 10% of the max value with a cosine schedule. Mean firing rate for Lorenz was 0.3. In all cases, we used $K = 5$ for the temporal smoothness loss in Eq. 1.

Parameter	Lorenz	Monkey Reach	Human BCI
Dataset Details			
Num training trials	3500	2008	8417
Trial length (bins)	256	140	512 (max)
Data channels (neurons)	128	182	128
Model Details			
Hidden layer channels	256	256	256
Latent channels	8	16	32
Num AE blocks	4	4	6
Spike history	No	Used in unconditional	No
Training Details			
Max learning rate	0.001	0.001	0.001
AdamW weight decay	0.01	0.01	0.01
Num epochs	200	140 (early stop.)	400
Num Warmup Epochs	10	10	20
Batch size	512	512	256
L_2 reg. β_1	0.01	0.001	0.001
Temporal smoothness β_2	0.01	0.2	0.1
CD mask prob. p	0.2	0.5	0.2

A3.1 Computational Resources

We performed all training and evaluation of LDNS on the Lorenz and Monkey reach datasets on an NVIDIA RTX 3090 GPU with 24GB RAM. For the Human BCI data, we used an NVIDIA A100 40GB GPU.

The autoencoder for the Lorenz dataset is trained in ≈ 6 minutes, and the diffusion model in ≈ 20 minutes. For the evaluation, all sampling is performed on the GPU in 5 minutes. The *effective GPU wallclock time* (time when the GPU is utilized) for the entire training and evaluation run is within 30 minutes.

Table 3: Training details for diffusion models on Lorenz, Monkey reach, and Human BCI datasets. We used the same learning rate scheduler as for the autoencoder.

Parameter	Lorenz	Monkey Reach	Human BCI
Model Details			
Latent channels	8	16	32
Hidden layer channels	64	256	384
Num diffusion blocks	4	6	8
Num denoising steps	1000	1000	1000
Training Details			
Max learning rate	0.001	0.001	0.001
AdamW weight decay	0.01	0.01	0.01
Num epochs	1000	2000	2000
Num warmup epochs	50	50	100
Batch size	512	512	256

For the Monkey reach dataset, the autoencoder with the given hyperparameters is trained in ≈ 8 minutes, and the unconditional and conditional diffusion models in 40 minutes to 1 hour. With similar sampling times as in Lorenz, the effective GPU wallclock time is approximately within one hour. Optimizing the autoregressive observation model took less than 1 minute.

AutoLFADS, the baseline used for unconditional sampling for the Monkey reach dataset, was trained on a cluster of 8 NVIDIA RTX 2080TI GPUs for one day. As it requires automated hyperparameter tuning to achieve the best accuracy using population-based training (PBT, [24]), AutoLFADS is significantly more compute-expensive to train than LDNS.

For the Human BCI dataset, due to larger trial lengths, more data points, and more heterogeneous temporal dynamics, we trained a slightly larger autoencoder and diffusion model than in Monkey reach. The autoencoder took 50 minutes to train, and the diffusion model took 10 hours to train. Sampling from the trained model took 9 minutes, resulting in a total of under 12 hours of effective GPU wallclock time.

We ran several preliminary experiments for LDNS to optimize the architecture and hyperparameters, as well as for designing appropriate evaluations. We estimate the total effective GPU wallclock time to be $\approx 10\times$ that of the final model runs. The AutoLFADS baseline was only trained once with PBT, as this framework automatically optimizes the model hyperparameters.

We implemented all training and evaluation code using the Pytorch framework[2], and used Weights & Biases[3] to log metrics during training.

A4 Baseline comparison: Latent Factor Analysis via Dynamical Systems - LFADS

Latent Factor Analysis via Dynamical Systems (LFADS) is a sequential variational autoencoder used to infer latent dynamical systems from neural population spiking activity [51, 36]. LFADS consists of an encoder, a generator, and optionally, a controller, all of which are RNNs. The generator RNN implements the learned latent dynamical system, given an initial condition and time-varying inputs. The internal states of the generator are mapped through affine transformations to lower-dimensional latent factors and single-neuron Poisson firing rates. The encoder RNN maps the neural population activity into an approximate posterior over the generator's initial condition.

At each timestep, the controller RNN receives both encoded neural activity and the latent factors from the previous timestep and outputs an approximate posterior over the input to the generator. The entire model is trained end-to-end to maximize the ELBO, as is done in VAEs. To address the difficulty of hyperparameter optimization for LFADS, Population-Based Training (PBT) has been proposed to automate hyperparameter selection, termed AutoLFADS [25].

[2]Paszke et. al. PyTorch: An Imperative Style, High-Performance Deep Learning Library (2019)
[3]Lukas Beiwald. Experiment Tracking with Weights and Biases (2022)

In our experiments with the monkey reach dataset, we use the PyTorch implementation of AutoLFADS [47]. We use the hyperparameters and search ranges from Pei et al. [38], but omit the controller RNN to simplify generation from prior samples. Although this might limit the model's expressiveness, prior research indicates that the monkey reach data can be well-modeled as autonomous, without external inputs from the controller [10]. LFADS has previously performed well on this data without the controller [36].

We generate samples from LFADS by sampling initial conditions from the Gaussian prior, running the generator RNN forward, and Poisson-sampling spikes from the resulting firing rates. For inclusion of spike history in the observation model of LFADS, we used the same training method and hyperparameter settings as in LDNSsh (appendix A1.3).

A5 Supplementary baseline comparisons

For an extended baseline comparison, we implemented two additional methods for the task of unconditional generation on the Monkey dataset (Sec. 3.4) – Targeted Neural Dynamical Modeling (TNDM, [21]) and Poisson-identifiable VAE (pi-VAE, [63]). It is important to note that while both TNDM and pi-VAE have demonstrated success in analyzing neural and behavioral data, neither was specifically designed for realistic spike train generation. The architectural choices in our implementation of these methods reflect their original intended applications in neural data analysis rather than generation of neural spiking data. Nevertheless, our comparisons show that LDNS, especially with spike-history, is superior or on par with all other methods (Table 4).

A5.1 Targeted Neural Dynamical Modeling (TNDM)

TNDM [21] is a VAE-based model designed to jointly model neural activity and behavior. TNDM extends LFADS by using an RNN to generate latent dynamics that are mapped to both neural activity and behavioral variables. TNDM separates the latent space into behavior-specific and behavior-independent subspaces to disentangle task-relevant and intrinsic neural dynamics.

For our comparison on the unconditional monkey reach task, we trained TNDM using the architecture and hyperparameters proposed in the original implementation of the paper[4]. We used 64-dimensional latent dynamics for each of the two sets. These project to a total of 10 latent factors z (5 behavior-specific z_r and 5 behavior-independent z_i), which is the maximum number demonstrated in the original work.

To generate unconditional samples, we sampled initial generator states from a standard normal prior $\mathcal{N}(0, I)$, then generated the latent dynamics and projected into neuron rates the same way as in LFADS. TNDM, performs well in matching real data in spike statistics (Supp. Fig. A13 cyan, Supp. Fig. A16e) and temporal dynamics (Supp. Fig. A15). Overall, we observe that LDNS captures spike statistics better than TNDM, except for the the population spike history count. In all metrics, LDNS augmented with spike history outperforms TNDM on spike statistics.

A5.2 Poisson-identifiable VAE (pi-VAE)

Poisson-identifiable VAE (pi-VAE) [63] is a VAE-based model for count data that ensures identifiability in the latent space. pi-VAE does **not** model temporal dependencies, instead treating each time point as an independent sample.

We trained pi-VAE on the monkey dataset using the original architecture and hyperparameters[5]. We used a General Incompressible-flow Network as a decoder, with 2 behaviorally relevant dimensions and 2 independent dimensions in the latent space. However, our evaluation context differs significantly from the original paper's demonstrations: while pi-VAE was initially evaluated on 50ms time bins and straight reaches only, our comparison uses 5ms bins and conditions on angles across all reaches at both middle and end trajectory points. The "label", or behavior, is presented as a 4-dimensional vector containing the cosine and sine of initial and final reach angles. Since this has a conditional latent space, sampling is performed by sampling angles randomly.

[4] Code adapted from Github respository: https://github.com/HennigLab/tndm.

[5] Code adapted from article by Lyndon Duong (2021) – https://www.lyndonduong.com/pivae.

Importantly, sampling from pi-VAE does not introduce any temporal dependence between spike bins within a trial — pi-VAE was not intended to be a generative model of neural spiking data. The lack of temporal modeling in pi-VAE's is a fundamental limitation for generating realistic spike trains, as evident in our empirical comparisons (Supp. Fig. A13 in yellow, Supp. Fig. A15,A16f). Note that this failure cannot be diagnosed simply from looking at the sampled spiking data (Supp. Fig. A14).

A5.3 Contributions of LDNS in context to LFADS, TNDM and pi-VAE

- LDNS is designed specifically for the purpose of accurately generating neural spiking data (unconditionally or conditionally)—a task often ignored by other LVMs designed for neural data analysis such as LFADS, pi-VAE, and TNDM.
- The S4 autoencoder and diffusion model in LDNS are trained in separate stages, offering modularity, while both components naturally account for temporal dependencies (unlike pi-VAE).
- S4 is autoregressive, similar to other RNN-based models, but empirically we found it to perform better when extending past the training trial length (compared to LFADS, see Sec. A6.1).
- One feature provided by some neural-behavioral analysis models (such as pi-VAE and TNDM) is an explicit disentangling of neural vs. behavior-relevant latents. While we found LDNS latents contain relevant behavioural information (Fig. 5e-g, Supp. Fig. A17), we did not explicitly supervise the latent space to induce this property.
- Finally, the spike history-dependent observation model in LDNS is modular and can be optimized post-hoc using rate predictions of any model to improve spike generation quality. We observed this with LDNS as well as LFADS (Table 1).

Table 4: **Added Baselines metrics comparison.** D_{KL} for the population spike count histogram and RMSE comparisons. Mean and standard deviation across 5 folds sampled with replacement. **Bolded** entries represent best-performing values for Poisson observation and spike-history observation model, respectively.

Method	D_{KL} psch	RMSE pairwise corr	RMSE mean isi	RMSE std isi
pi-VAE	$0.0063 \pm 2.9\text{e-}4$	$0.0031 \pm 1.08\text{e-}5$	0.064 ± 0.002	0.034 ± 0.001
TNDM	$\mathbf{0.0028 \pm 6.6\text{e-}5}$	$0.0027 \pm 1.17\text{e-}5$	0.057 ± 0.004	0.029 ± 0.001
AutoLFADS	$0.0040 \pm 2.2\text{e-}4$	$0.0026 \pm 1.25\text{e-}5$	0.039 ± 0.003	0.029 ± 0.001
LDNS	$0.0039 \pm 3.9\text{e-}4$	$\mathbf{0.0025 \pm 1.1\text{e-}5}$	$\mathbf{0.037 \pm 0.001}$	$\mathbf{0.023 \pm 0.001}$
AutoLFADSsh	$0.0036 \pm 2.1\text{e-}4$	$0.0026 \pm 1.8\text{e-}5$	0.034 ± 0.002	$\mathbf{0.023 \pm 0.0001}$
LDNSsh	$\mathbf{0.0016 \pm 6.2\text{e-}4}$	$\mathbf{0.0025 \pm 1.07\text{e-}5}$	$\mathbf{0.024 \pm 0.002}$	$\mathbf{0.023 \pm 0.001}$

A6 Supplementary Figures Lorenz

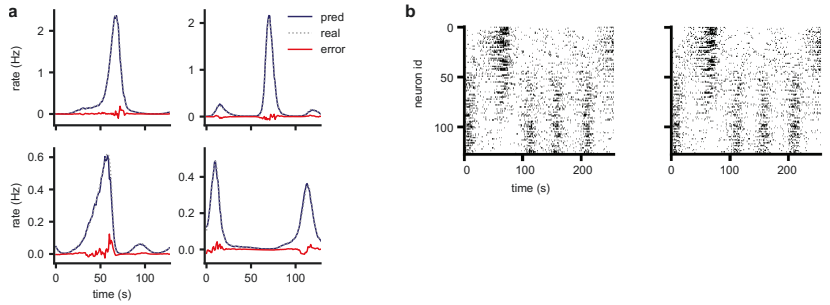


Figure A2: **The autoencoder captures the gt Lorenz synthetic firing rate perfectly a)** Autoencoder predictions (pred) and true rates from the test set, together with their difference (error, in red). **b)** Reconstructions sampled from the Poisson observation model (right) closely resemble the test sample (left). Both spiking activity is binarized for the visualization.

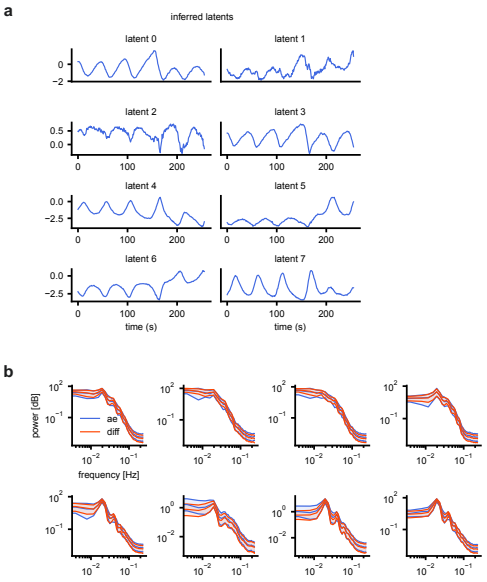


Figure A3: **The S4 autoencoder infers smooth latents from discrete spikes and samples from the diffusion model capture the latent distribution. a)** Inferred autoencoder latents for a test sample. **b)** Power spectral density for all eight latent dimensions for the inferred autoencoder training set and samples from the diffusion model

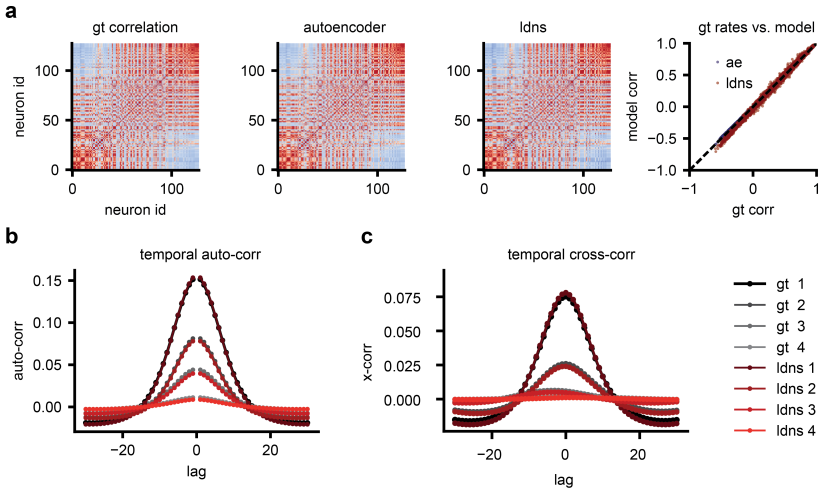


Figure A4: **LDNS captures the correlation structure of the Lorenz dataset a)** Both the autoencoder and LDNS-sampled rates capture the ground truth instantaneous correlation structure of the synthetic rates. **b)** The auto-correlation structure of ground truth and sampled spiking activity matches perfectly in 4 neuron groups, sorted according to correlation strength. Synthetic Lorenz data group x is denoted by gt x, LDNS samples as ldns x. **c)** The time-lagged cross-correlational structure is also perfectly captured by LDNS in all groups.

A6.1 Length generalization of LFADS on Lorenz

To analyze whether LFADS [36] exhibits similar length generalization properties as LDNS, we trained an AutoLFADS model on the Lorenz dataset (256 bins). We used the same architecture as the Monkey dataset, with 40-dimensional latent dynamics. We sampled initial conditions from the LFADS prior, then generated dynamics for both the original 256 steps and for an extended duration of $16\times$ the training length.

Qualitatively, we observed that while LFADS produced trajectories resembling the attractor dynamics of the ground truth Lorenz system (Supp. Fig. A5a, across various dimension combinations), these trajectories often diverged when run for longer intervals (Supp. Fig. A5b). However, the system eventually returned to typical dynamics.

Furthermore, when generating for extended durations, we observed that the mean population firing rates sometimes reached extreme values in some samples (Supp. Fig. A6), though they eventually returned to typical ranges. This behavior was not observed in LDNS samples, suggesting that bidirectional generation in the diffusion model provides more stability in variable-length generation.

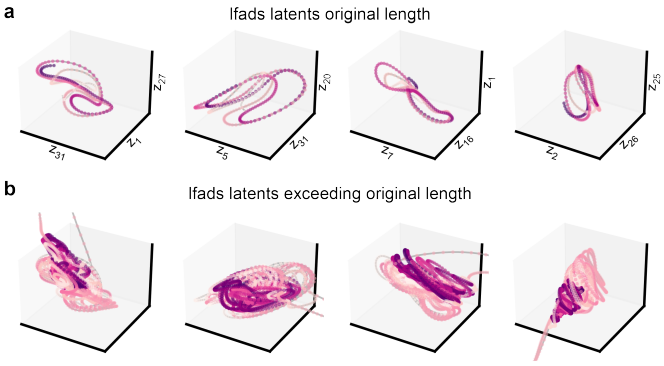


Figure A5: **Length generalization of LFADS on Lorenz** Different projections of a 40-dimensional latent space from LFADS trained on the Lorenz system. Trajectories are compared between **a)** the original length and **b)** 16 times the original length using sampled initial conditions. For comparison with LDNS length generalization, see Fig. 2c

.

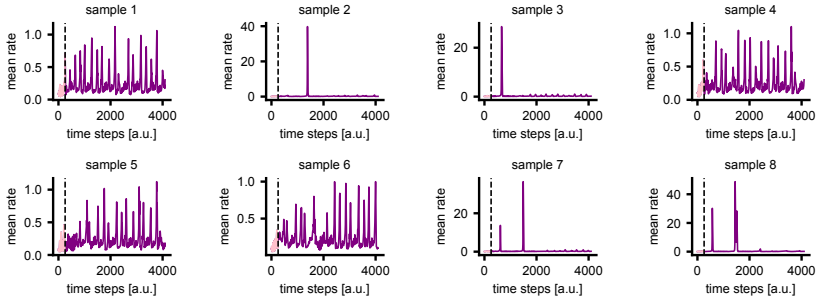


Figure A6: Mean population firing rates for eight different samples, shown for both the original length (pink) and $16\times$ the original length (purple).

A7 Supplementary Figures Human BCI

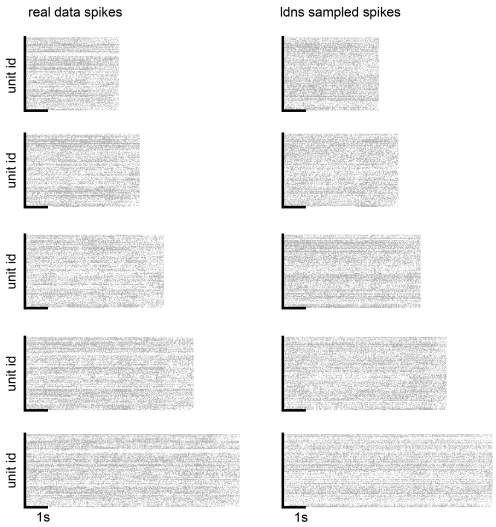


Figure A7: Visual comparison of different sampled spiking data from LDNS, with five samples from the true dataset.

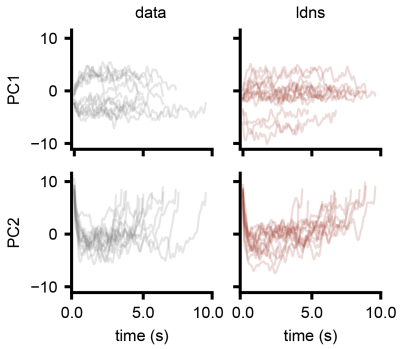


Figure A8: First two principal components (PCs) of smoothed spikes from true data (left) and model samples (right). Each line represents one sampled trial. Spikes were smoothed using a Gaussian filter with a window of 160ms prior to extracting the PCs.

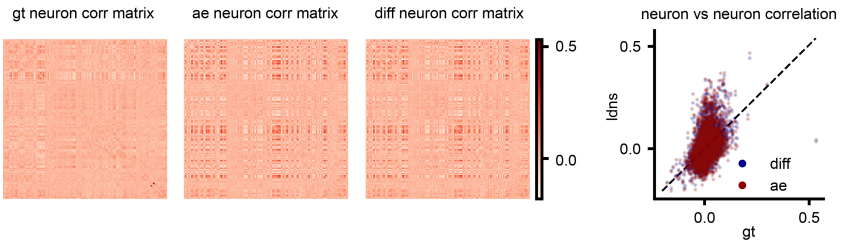


Figure A9: Correlation matrices for real spiking data from human and LDNS, comparing the autoencoder-inferred (ae) correlation (sampled from reconstructed rates) and correlation of sampled spikes (diff). The deviations from the data (gt) already arise at the autoencoder stage.

A8 Supplementary Figures Monkey Reach task

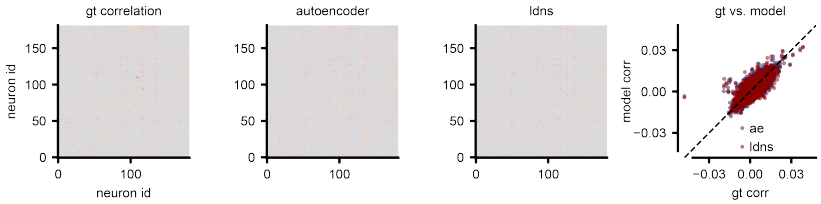


Figure A10: Correlation matrices for real spiking data and samples from Poisson LDNS, concatenated across trials, comparing the autoencoder-inferred (ae) correlation (sampled from reconstructed rates) and correlation of sampled spikes (diffusion + ae). Most deviations from the data (gt) already arise at the autoencoder stage.

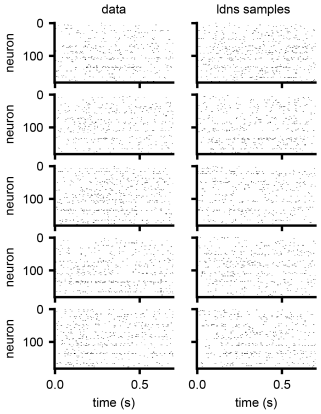


Figure A11: Visual comparison of different sampled spiking data from LDNS with five samples from the real dataset.

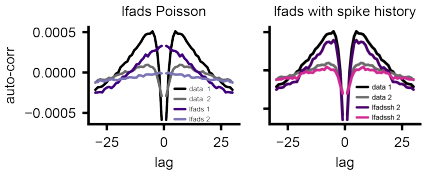


Figure A12: **Equipping LFADS with spike history** Auto-correlation of data, LFADS samples with Poisson observations (left) and LFADSsh samples (with spike history), grouped according to correlation strength.

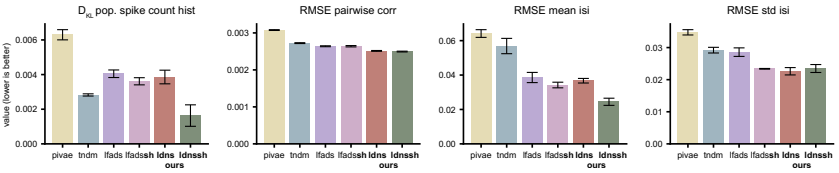


Figure A13: **Performance comparison with additional baselines** pi-VAE [63], TNDM [21], LFADS [36], LFADS with spike history (LFADSsh), LDNS and LDNS with spike history (LDNSsh). Mean and standard deviation across 5 folds sampled with replacement.

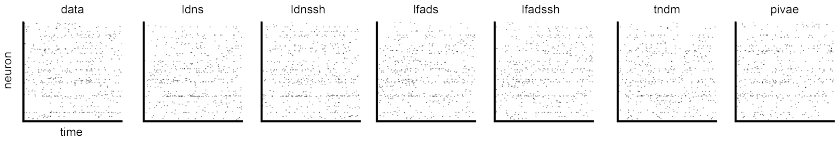


Figure A14: Visual comparison of sampled spiking data from LDNS and all baselines with real data.

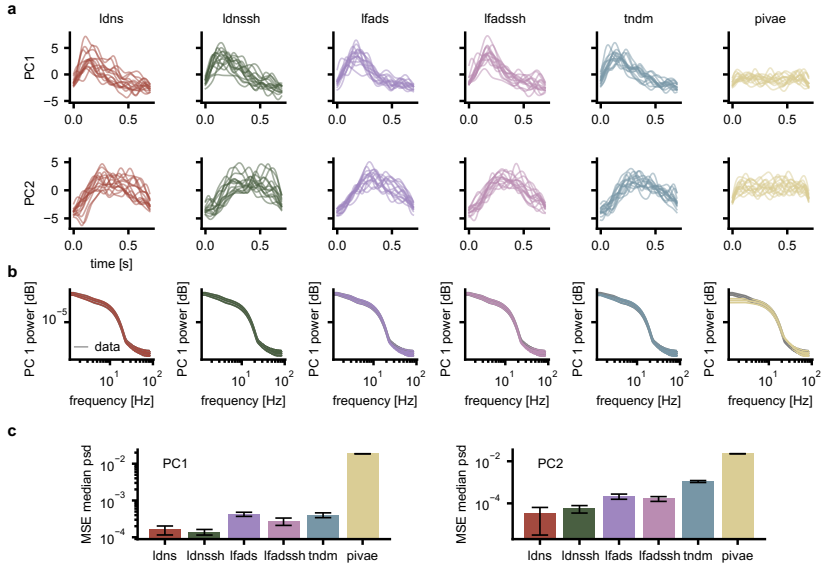


Figure A15: **Comparing principal components of smoothed sampled spikes a)** First two principal components (PCs) of smoothed spikes from model samples. Each line represents one sampled trial. Spikes were smoothed using a Gaussian filter with a window of 40ms prior to extracting the PCs. The PCs were fit using real data. Since pi-VAE does not account for temporal dynamics, it does not show any temporal structure in the PCs. **b)** Power spectral density (PSD) of PC1, plotted for model vs. data (in grey). **c)** Mean squared error of median PSD between model samples and data for PC1 (left) and PC2 (right). LDNS and LDNSsh perform the best here, with pi-VAE showing large errors.

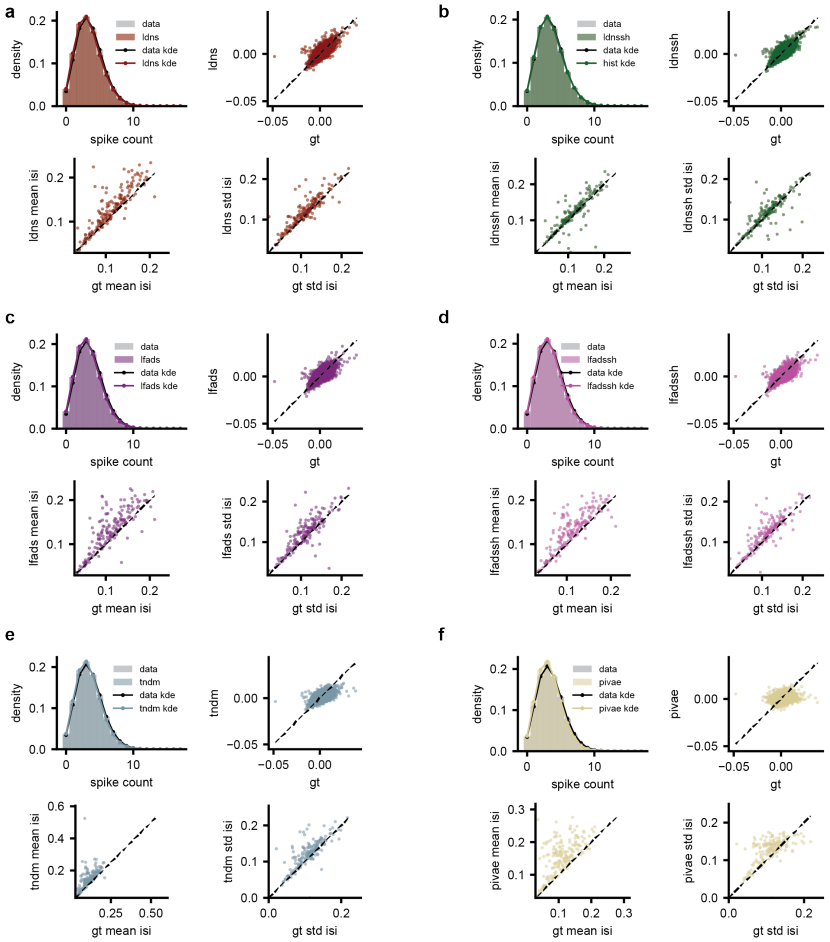


Figure A16: Population-level and single neuron-level statistics of **a)** LDNS, **b)** LDNSsh (with spike history), **c)** LFADS, **d)** LFADSsh (with spike history), **e)** TNDM, and **f)** pi-VAE.

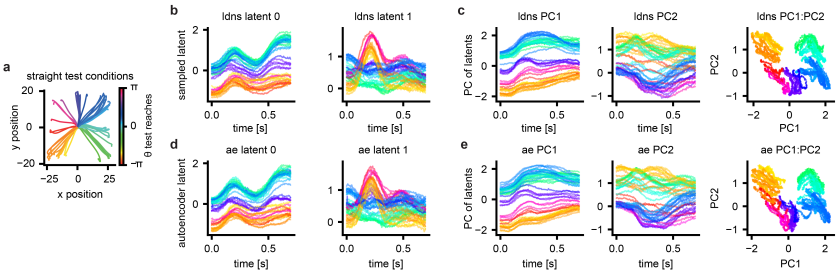


Figure A17: **Comparison of LDNS latent space trajectories of inferred and conditionally sampled latents a)** Straight reaches from the Monkey reach test set. **b)** LDNS sampled latents (velocity-conditioned on reaches shown in a). **c)** PCs of LDNS sampled latents. **d)** Autoencoder-inferred latents of corresponding neural activity for the reaches shown in a). **e)** PCs of autoencoder-inferred latents.