Multi-Channeled Localization Microscopy with Cloud-Enabled Simulation-Based Machine Learning

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Lucas-Raphael Müller

aus Bad Kreuznach

Tübingen

2024

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

Tag der mündlichen Qualifikation:	19.11.2024
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter/-in:	Prof. Dr. Jakob Macke
2. Berichterstatter/-in:	Prof. Dr. Sven Nahnsen

Multi-Channeled Localization Microscopy with Cloud-Enabled Simulation-Based Machine Learning

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Lucas-Raphael Müller

aus Bad Kreuznach

Tübingen

2024

Abstract

Simulation-based Machine Learning (ML) algorithms have proven successful in the realm of microscopy and enabled significant speedups in Single Molecule Localization Microscopy (SMLM) compared to conventional algorithms. These routines, however, did not work for all SMLM modalities and were difficult to use for individuals without dedicated hardware and computational experience. This thesis addresses these challenges.

SMLM, an inverse problem, is a strong candidate for simulation-based ML. In SMLM, fluorophores of one or more kinds are stochastically activated, resulting in sparse events (emitters) imaged over tens to hundreds of thousands of frames. These emitters are localized and rendered to compute the final superresolution image. SMLM is inherently slow due to its need for sparse emitters. Recent ML approaches tackled this problem by enabling *high-density* imaging. However, the field lacks a dedicated algorithm for high-density, multi-channeled SMLM applications.

ML algorithms in microscopy are applied to various tasks including denoising images, improving their resolution, segmenting biological sites, or detecting biological events. Yet, these methods often rely on dedicated hard- and software setups and have not found their way into daily scientific routines for that reason. The field lacks approaches for repetitive large-scale workflows without complicated manual intervention.

We present *DECODE-Plex*, a new framework for high-density multi-channeled localization in SMLM, addressing multi-color and biplane imaging. DECODE-Plex is trained on-the-fly by a simulated training procedure. We demonstrate DECODE-Plex's performance across various densities for simulated and experimental data.

Furthermore, we present *DECODE-OpenCloud*, a cloud-backed solution for ML algorithms in microscopy. DECODE-OpenCloud abstracts away hardware and maintenance concerns providing researchers and developers with a user-friendly yet production-ready API. It encourages the integration of unused local computing power, benefiting from centralized maintenance and computational power. DECODE-OpenCloud's design allows for integration without significant additional effort, and we present three algorithms for SMLM as reference implementations: Localization with *DECODE*, *DECODE-Plex* and drift correction with *COMET*.

Kurzfassung

Simulationsbasierte auf maschinellem Lernen (ML) beruhende Algorithmen haben sich im Bereich der Mikroskopie bewährt und deutlich schnellere Aufnahmen in der Einzelmolekülmikroskopie (SMLM) im Vergleich zu klassischen Algorithmen ermöglicht. Diese Algorithmen unterstützen bisher jedoch nicht alle SMLM Modalitäten und sind schwer ohne dedizierte Hardware- und Softwareerfahrung nutzbar. Die vorliegende Arbeit adressiert diese Probleme.

SMLM ist eine interessante Anwendung für simulationsbasiertes ML, da es um ein inverses Problem handelt. Eine Probe wird hier derart präpariert, dass Fluorophore einer oder mehrerer Farben stochastisch aktiviert werden was zu vereinzelten sogenannten Emittern führt, die auf zehn- bis hunderttausenden Bildern aufgenommen werden. Jene Emitter werden lokalisiert um das superaufgelöste Bild zu berechnen. SMLM ist langsam, da die Aufnahmezeit direkt von der Dichte der Emitter abhängt. Kürzlich entwickelte ML Algorithmen erlauben höhere Dichten, der Community fehlt jedoch ein entsprechender Algorithmus für Vielkanaldaten.

ML Algorithmen werden in der Mikroskopie bei vielfältigen Anwendungen wie dem Entrauschen, der Auflösungssteigerung, der Segmentierung von biologischen Proben oder der Detektion von biologischen Ereignissen eingesetzt. Diese Methoden benötigen jedoch oft dedizierte Hard- und Softwaresetups sowie entsprechende Dateninfrastruktur und haben aus diesem Grund oft noch keinen Einzug in die Laborroutine erhalten.

Wir stellen DECODE-Plex vor, ein neuartiger Algorithmus für hochdichte vielkanal SMLM-Daten und adressieren im besonderen Mehrfarben- und Multifokal-SMLM. DECODE-Plex wird mit live simulierten Daten trainiert. Wir zeigen die robuste Performance von DECODE-Plex auf simulierten und experimentellen Daten verschiedener Dichten.

Darüber hinaus stellen wir DECODE-OpenCloud vor, eine Cloud-basierte Lösung für ML Algorithmen in der Mikroskopie. DECODE-OpenCloud abstrahiert Hardware und Wartung von den Algorithmen und ermöglicht so deren einfache Benutzung via API. Ungenutzte Rechenleistung kann einfach integriert werden, um so von zentralisierter Wartung und geteilten Resourcen zu profitieren. Weitere Algorithmen können ohne großen Aufwand integriert werden; wir stellen drei Algorithmen als Referenz vor: Lokalisation mit DECODE und DECODE-Plex und Driftkorrektur mit *COMET*.

Acknowledgments

I would like to thank and express my deep gratitude to my supervisors, Jonas Ries and Jakob Macke. They allowed me to perform research in an open environment, always in search for the best idea. I will look back smiling when I remember our discussions at the whiteboard in the lab.

I want to thank all present and former lab members of the Ries lab and the Macke lab for the time which we spent doing research, at the retreat, and having a coffee. In particular, I would like to thank Takahiro Deguchi, Arthur Jaques, Philipp Hoess, Aline Tschanz, Yu-Le Wu, Christopher Heidebrecht, Soheil Mojiri, Gabin Agbale, Alejandro Linares, Zach Marin, Auguste Schulz, Jaivardhan Kapoor, Julius Vetter, Michael Deistler, Jan Boelts, Artur Speiser, Nestor Castillo, Sara Klingelhöfer, Markus Mund, Joran Deschamps, and Ulf Matthis. I would like to thank Franziska Weiler for administrative support. I thank Benjamin Brandstetter for feedback on this thesis. I would like to thank my friends and family for their advice and support throughout my journey.

Insbesondere möchte ich meiner Mutter Anne-Rose Mindnich-Müller für die gesamte Unterstützung während meiner Ausbildung danken.

Finally *I say thank you for the music.*

Contents

List of Figures

Chapter 1

Introduction

Vision is the core sense we utilize to capture the world that surrounds us. It has coined the phrase *seeing is believing*. Microscopy extends the limits of human vision and allows us to study otherwise invisible structures.

The history of microscopy is rich and led to many great scientific discoveries. Robert Hooke studied plants and insects through what he named *magnifying glasses* and first used the term *cell* in his work (Hooke, Robert 1665). Walther Flemming laid foundational work on observing cell division and discovering chromosomes (Flemming 1880). Abbe and Rayleigh studied the relationship between the wavelength of light and the point up to which structures can be distinguished. Their theories result in the *Abbe limit* and the *Rayleigh criterion*, which, for visible light, both conclude a minimally resolvable distance of around 250 nm (see Equation 2.4). The core determining factor of the resolvable distance is its linear dependence on the wavelength. For that reason, e.g. electron microscopy can achieve much higher resolutions than light microscopy, in an order of 0.1 nm because of the small wavelength of energetic electrons (Franken et al. 2020). All microscopy techniques have strengths and weaknesses, and light and electron microscopes are no exception. While microscopes with shorter wavelengths can, in principle, achieve higher resolution, their use might be prohibitive, e.g. because thick specimens should be imaged, due to phototoxicity in live-cell imaging, due to radiation exposure or the need for specific sample preparation. In section 1.1, we introduce superresolution microscopy in more detail, which is the focus of this study.

As microscopy has advanced, so too have the methods for processing and analyzing such data. Advancements in Machine Learning (ML) and in particular in Deep Learning (DL) have profoundly impacted the field of biomedical sciences and in microscopy in particular. They have since become an integral part of most imaging techniques, allowing for improved image quality or the automation of microscopes. In the following, we briefly outline important aspects of ML in biomedical science in section 1.2 and elaborate on the challenges of integrating these

compute-intensive algorithms into the daily scientific routine.

1.1 Superresolution Microscopy

Superresolution microscopy is a class of light microscopy techniques, with a resolution greater than what is derived from diffraction theory (see section 2.1.1). None of these techniques contradict the theories established by Abbe and Rayleigh, their working principle are such that their achievable resolution is determined by other factors than the minimum distance two point sources can be distinguished from one another.

Several of these techniques have been proposed over time, and we shall briefly introduce their working principle in the following. The focus of this thesis is Single Molecule Localization Microscopy (SMLM), which we introduce in more detail in section 1.1.1 and outline its theoretical foundations in section 2.1.

Structured Illumination Microscopy (SIM)

The first superresolution technique was Structured Illumination Microscopy (SIM), which uses a patterned Illumination with a moveable diffracting grating (Lukosz and Marchand 1963; Gustafsson et al. 2008). The resulting *Moire* pattern, which stems from the interference of the patterned illumination and the sample structure itself is then used to reconstruct the a superresolved microscopy image. Typically, SIM achieves a resolution gain of a factor of two both laterally and axially (Gustafsson et al. 2008).

Stimulated Emission Depletion (STED)

Stimulated Emission Depletion (STED) is a superresolution technique that achieves higher resolutions than Structured Illumination Microscopy (SIM). At its core is the *donut* shaped beam that allows for the Point Spread Function (PSF) to have smaller effective extent than the diffraction limit. The core idea is to use a superposition of excitation and a depletion laser, where the latter is phase-modulated to form the characteristic donut shape (Hell and Wichmann 1994, Klar and Hell 1999). The excitation laser is of a smaller wavelength (i.e., higher energy) than the depletion laser. The purpose of the depletion laser is to stimulate red-shifted emission of the fluorophores, which can be filtered out by common color filters. The effective emission of the fluorophores is then confined to the inner part of the donut, which has less spatial extent than the diffraction limit. The beam is scanned over the biological site. The resolution obtainable by STED is closely coupled to the effective depletion to suppress fluorescence in the outer part of the donut. Here, high laser powers can lead to photobleaching, i.e., the destruction of fluorophores.

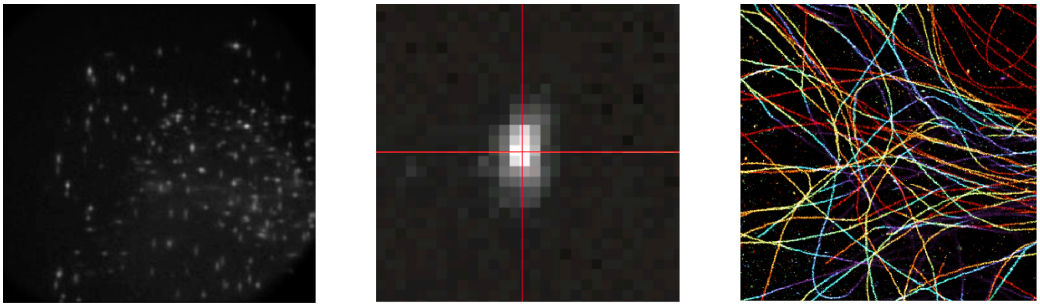


Figure 1.1: The working principle of Single Molecule Localization Microscopy (SMLM). A stack of frames, each with a sparse set of emitters, is acquired. The emitters are localized by a fitting algorithm, and the localizations are extracted. The localization set is used to render the final superresolution image.

Minimal emission fluxes (MINFLUX)

MINFLUX is a recent instance of localization microscopy also influenced by aspects of *STED*. It is a classical *localization microscopy* technique in that its output is a list of localizations. However, its working principle is not by software-based fitting but instead by live localization of emitters via a hardware-implemented feedback loop (Balzarotti et al. 2017). At its core is the doughnut-shaped *excitation* that is used in *STED* for depletion. The position of the excitation beam is then adjusted in specific patterns around the estimated position of the emitter by a feedback loop implemented on a *FPGA* device. The excitation beam is moved in dynamic patterns, and the emitter's position is then inferred from the respective photon counts at the various positions around the emitter. MINFLUX can observe 3D motor protein stepping in living cells directly (Deguchi et al. 2023).

1.1.1 Single Molecule Localization Microscopy (SMLM)

Single Molecule Localization Microscopy (SMLM) is a class of super-resolution microscopy techniques that output a set of localizations of individual fluorophores in a sample. The final microscopy image is rendered based on this set of localizations (see Figure 1.1). The basic idea of SMLM is to activate only a small subset of fluorophores in the sample at a time, then to undergo a *fitting* procedure to estimate the position of the fluorophore. The key is that the fitting procedure can estimate the position of the fluorophore with greater precision than the extent of the microscope's Point Spread Function (PSF). With its core principle of SMLM comes its inherently slow imaging speed since many frames need to be acquired. Typical imaging times of SMLM acquisitions are in an order of an hour.

It is a strength of SMLM to be able to colocalize different objects at the same biological site

through multiple targets which is called *multi-color* SMLM. In practice, this is realized by labeling the sample with multiple, spectrally different, fluorophores (see section 2.1). In the state-of-the-art *ratiometric* imaging, these different targets are imaged simultaneously in multiple channels. The targets can be distinguished by the resulting ratio of their signal in all channels. In *biplane* SMLM, a similar *multi-channeled* setup can be used to resolve the axial position of emitters instead of distinguishing multiple targets from one another. Both are introduced in section 2.1.

The density at which the sample is imaged is closely related to the imaging speed. The density is influenced both by the sample preparation and can also be tuned during the sample acquisition by changing the laser power. When not dictated by the sample preparation or the need for live-cell imaging, higher densities are favorable to achieve faster imaging, but this will come at the cost of degraded imaging quality (Diekmann et al. 2020). High densities result in more localizations per time however the task of fitting localizations get more difficult, as the emitters will have neighbors in close proximity (see Figure 2.1b). For classical localization algorithms, this results in two intertwined problems: First, it is unclear how many emitters are present in a certain ROI, and second, the Maximum-Likelihood-Estimation (MLE) optimization task will be influenced by that and harder to solve.

The working principle of SMLM is, computationally speaking, an inverse problem. We try to infer the cause (i.e., the structure of the biological site) by looking at the observation (the camera frames). This renders it to be an interesting application for a simulation-based Machine Learning approach. Having an accurate simulator for our microscopic setup, we can easily compute new samples and let a Deep Neural Network perform the inverse task of reconstructing the structure that caused the observation. Several recent algorithms in SMLM applied this approach successfully (Nehme, Weiss, et al. 2018; Nehme, Freedman, et al. 2020; Speiser et al. 2021).

Simulation-based ML algorithms have been successfully applied at high-density SMLM data (Nehme, Weiss, et al. 2018; Nehme, Freedman, et al. 2020; Speiser et al. 2021; Fu et al. 2023). However, the field lacks an algorithm for multi-channeled high-density data to fill in the gap for multi-color or biplane imaging.

1.2 Machine Learning in Biomedical Applications

In recent years, advancements in Machine Learning, particularly in Deep Learning, have led to dramatic achievements in biomedical sciences. Common anaylsis tasks in the biomedical realm are segmentation (Ronneberger, Fischer, and Brox 2015; Schmidt et al. 2018; Buchholz et al. 2020; Weigert, Schmidt, Haase, et al. 2020; Isensee et al. 2020), object detection (Hung

et al. 2020, Midtvedt et al. 2022) or image restauration techniques such as denoising (Krull, Buchholz, and Jug 2019; Buchholz et al. 2020).

In the context of SMLM, ML algorithms have not only been proposed for the very localization step (Nehme, Weiss, et al. 2018; Nehme, Freedman, et al. 2020; Speiser et al. 2021; Fu et al. 2023) but also operating on the super-resolved image directly (Ouyang, Aristov, et al. 2018a) or for downstream structural analysis of the localizations (Wu et al. 2023).

Given the impact of ML in the biomedical field, it is vital to consider the translational aspects of integrating such algorithms into routine scientific workflows and addressing the challenges of productionizing and maintaining these applications in high-throughput environments. This particularly includes all aspects of running and maintaining high-performance compute clusters. Training recent LLMs poses unprecedented requirements on computing infrastructure, which is often out of reach for public institutions and only accessible to a small set of companies. In large-scale production-oriented environments, however, aspects of high availability, robustness, and monitoring are of utmost importance. We briefly introduce these aspects in section 1.2.2.

Utilizing scientific algorithms in a routined, large-scale workflow is often somewhere in between low-entry barrier graphical solutions, which often lack robustness automation capabilities, and production-ready environments, which require substantial infrastructure overhead and maintenance (see sections 1.2.1 and 1.2.2).

With this study, we present DECODE-OpenCloud aiming at that very spot for the routined scientific usage of ML algorithms in the realm of microscopy (sections 3.1.2 and A.2).

1.2.1 Translational Aspects of Machine Learning in Science

With the increased computational complexity of biomedical algorithms, these posed a significant burden on all downstream aspects of ML applications, i.e., concerning software engineering, computational resources, data management among others. To this day, many techniques have not found their way into the daily routine of researchers, because of the additional requirements these DL methods pose. Often, they need frequent retraining, which introduces latency as compared to classical, online methods, or require manual intervention, e.g., upon training failure. Compared to *classical* algorithms, these frequent semi-manual interactions make it difficult to completely encapsulate these algorithms and utilize them *as-is* over a long time span.

Software frameworks like *ilastik* (Berg et al. 2019), *ImJoy* (Ouyang, Mueller, et al. 2019) or *CellProfiler* (McQuin et al. 2018) embed various, complex and sophisticated ML algorithms in an easy to use, fully-fledged software solution for researchers.

Low entry barrier solutions like Jupyter Notebooks (Kluyver et al. 2016) gained significant popularity and underwent active development in the Python developer community. It has be-

come a mean of presenting scripted code with visual output to a broad audience even without prior computational experience. Oftentimes, institutions or enterprise environments host these Notebooks on *JupyterHubs.*[1] *Google Colaboratory*[2] (often referred to as simply Google Colab or Colab) is a publicly available platform where Jupyter-like notebooks are hosted and compute resources are provided by Google. It gained popularity in the scientific community as a means to host easy-access notebooks even including accelerated compute resources (i.e., GPU). However, these instances come with their own limitations as they offer limited support for productionized requirements, which include but are not limited to pinning software packages and environments, specifying minimum requirements for computing hardware and more.

1.2.2 Machine Learning in Large Scale Environments

In high-throughput, large-scale environments, productionized setups are needed to accommodate the needs for low manual interaction, robustness, and reliability. Typically, these workflows need to accomodate the following aspects:

Data Management Data management comprises all aspects from low-level storage and versioning to preprocessing, quality checking, logging, and rights management due to ethical or legal concerns. Typically, cloud-based bucketized storage or shared file systems are used for storing data. Data pipelines are often established with tools like *Apache Airflow*[3], *MLFlow*[4], or similar tools. These tools help automate and manage complex data workflows, ensure data integrity, and maintain accurate records of data transformations and processing steps.

Model Development Model development involves creating, training, and validating machine learning models. This phase is crucial for ensuring the models perform well on unseen data and meet the desired accuracy and performance metrics. Validation techniques are applied to assess the model's effectiveness and to tune hyperparameters for optimal performance.

Compute Resources Compute resources are essential for training and deploying machine learning models. These resources can include hypervisors, AWS, or other cloud-based solutions that provide the necessary computational power. The choice of compute resources depends on the specific requirements of the task, such as the amount of data, the complexity of the models, and the need for scalability.

[1] https://jupyter.org/hub
[2] https://colab.research.google.com
[3] https://airflow.apache.org/
[4] https://mlflow.org/

Deployment The deployment type is likely the greatest difference between machine learning in production and in an experimental, single-researcher scenario. Deployment needs to accommodate aspects of accessibility and interconnectivity of productionized services. Typically, models are wrapped and hidden behind a Representational State Transfer Application Programming Interface (REST-API), which serves the inference task. This allows for seamless integration with other services and applications, providing a robust and scalable solution for delivering ML capabilities to end users.

Monitoring and Maintenance Monitoring and maintenance are critical for ensuring deployed machine learning models' long-term success and reliability. Tools like TensorBoard and MLFlow are commonly used to track model performance, log important metrics, and identify potential issues. Continuous monitoring allows for timely updates and adjustments to the models, ensuring they remain effective and accurate over time. typically, a mixture of managed and custom services are used from a Hypervisor like *AWS*[5], *Google Cloud*[6] or *StackIT*[7] among others.

1.3 Thesis Outline

This chapter set the context of this thesis. The two main contributions of this thesis are:

DECODE-Plex A multiplexed, multi-channel high density fitting algorithm for SMLM, in particular for multi-color and biplane applications (see section A.1)

DECODE-OpenCloud A hybrid-cloud environment where researchers can share compute resources and algorithms in the realm of microscopy (see section A.2)

Chapter 2 provides more scientific background on the physical aspects of Single Molecule Localization Microscopy (SMLM) outlining the relevant aspects of the microscopic setup, its Point Spread Function (PSF) (section 2.2), the camera models and the characteristics of emitters (i.e., the photophysical model). In chapter 3 the manuscripts and publications are summarized and the author contributions are named. The manuscripts themselves can be found in sections A.1, A.2. Chapter 4 puts the thesis in a broader context, reflects and outlooks future directions.

[5] `https://aws.amazon.com`
[6] `https://cloud.google.com`
[7] `https://www.stackit.de`

Chapter 2

Scientific Background

In the following, we provide background on the most important aspects of multi-channeled Single Molecule Localization Microscopy (SMLM), in particular those which are relevant to DECODE-Plex due to its simulated training approach.

We outline the theoretical background of diffraction theory (section 2.1.1) and the resolution limit of SMLM and then introduce experimental realizations of a typical SMLM setup. This chapter particularly sheds light on the aspects that are important to the simulated training approach of DECODE-Plex. For this reason, we discuss the other two core components of our simulated training approach, the Point Spread Function (section 2.2) and the camera characteristics (section 2.3).

2.1 Single Molecule Localization Microscopy

2.1.1 Diffraction Limit of Light

The basic resolution limit of classical light microscopy comes from the fact that an optical system with light at a wavelength λ has a lower bound on the size of its optical response even for infinitely small objects. The PSF of an optical setup is the system's response to a point source. The image I of an object O is then described by the convolution of an object O with the PSF

$$I(\mathbf{r}) = O(\mathbf{r}) \circledast PSF(\mathbf{r}) \tag{2.1}$$

An ideal, abberation-free system has a PSF that is described by the Airy disk, which is deduced from the Fourier transform of the aperture function.

$$I(\theta) = I_0 \left[\frac{2J_1(k\,a \sin\theta)}{k\,a \sin\theta} \right]^2 \tag{2.2}$$

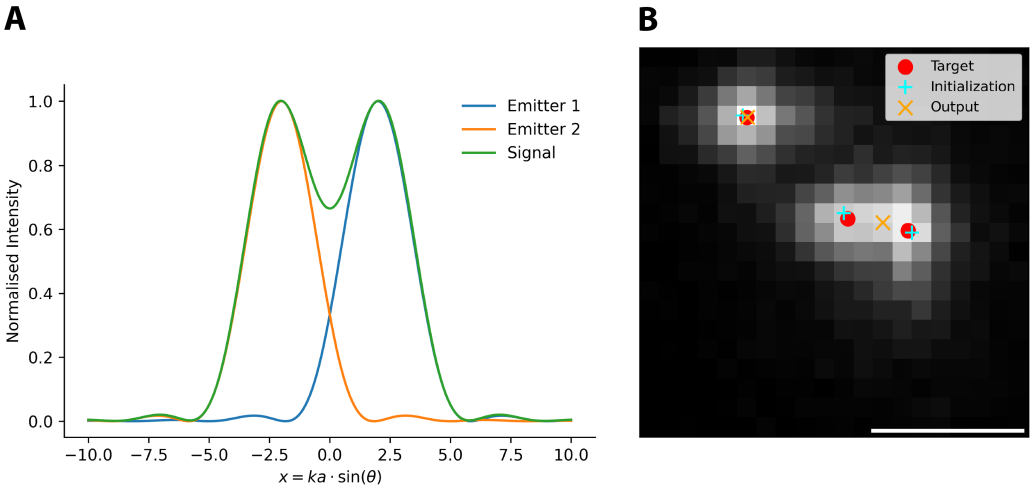


Figure 2.1: Emitters in close proximity. (a) Two point sources at Rayleigh distance and their observed signal. The two sources become increasingly indistinguishable as the emitters come closer together, in particular in conjunction with inevitable shot-noise due to the quantum mechanical nature of the photons and the effects of pixelation. **(b) Localiation in the presence of close emitters.** Illustrative optimization failure in the presence of close emitters for SMLM. The isolated emitter is detected and localized correctly, while the optimization algorithm converges to a suboptimal local minimum for emitters in close proximity. Scalebar 1 µm.

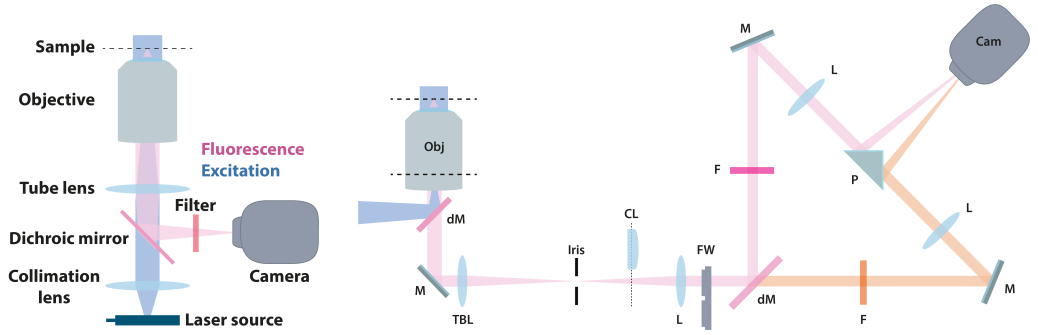
with $I(\theta)$ being the intensity of the light at an angle θ relative to the optical axis, I_0 is the intensity at the center of the Airy disk, J_1 is the Bessel function of the first kind of order 1, k is the wavenumber of the light, and a is the radius of the aperture.

Figure 2.1a shows to point sources in close proximity. From here, it is apparent, that there must be a fundamental limit on the minimum spatial frequency of structures that can be distinguished under the assumption of the already idealized PSF. If the point sources are closer together than a critical distance, their optical response will not be distinguishable from a single point source. This particularly holds true under the influence of inevitable noise sources (see section 2.3). While deriving it from different ends, both the Rayleigh criterion and the Abbe criterion arrive at similar minimally resolvable distances

$$d_{\text{Rayleigh}} = \frac{0.61\lambda}{\text{NA}} \tag{2.3}$$

$$d_{\text{Abbe}} = \frac{\lambda}{2\text{NA}} \tag{2.4}$$

with NA being the numerical aperture of the optical system; $NA = n\sin(\theta)$ with n being the refractive index of the medium and θ being the opening angle of the lens (Lord Rayleigh F.R.S. 1879; Abbe 1873).



(a) Minimal 2D SMLM setup.

(b) 3D High-throughput, dual-channel SMLM setup. *M* mirror, *dM* dichroical mirror, *F* filter, *FW* automated filter wheel, *L* (generic) lens, *CL* automated cylindrical lens, *TBL* tube lens, *P* prism, *Obj* objective, *Cam* camera.

Figure 2.2: Simplified SMLM setup. (a) 2D and **(b) dual-channel 3D** setup (2.2b). The dual-channel microscopy setup shows a single-camera dual-channel setup; other variants feature multiple cameras. The figures have been made and used with permission by Dr. Joran Deschamps. They are reprinted from his PhD thesis Deschamps, Mund, and Ries 2017.

2.1.2 Resolution

A rough approximation of the resolution limit in SMLM is

$$\sigma = \frac{\sigma_{PSF}}{\sqrt{N}} \tag{2.5}$$

where σ_{PSF} is the standard deviation of the PSF and N is the photon count (Lelek et al. 2021). Here, the effects of pixelation, noise, and non-Gaussian noise are not considered. Following Mortensen et al. 2010 leads to a more accurate estimate of the localization precision

$$\sigma_{\text{loc}} \geq \sqrt{\left(\frac{\sigma_{PSF}^2 + s_{px}^2/12}{N}\right)\left(\frac{16}{9} + \frac{8\pi\left(\sigma_{PSF}^2 + s_{px}^2/12\right)bg^2}{Ns_{px}^2}\right)} \tag{2.6}$$

where s_{px} is the pixel size and bg the background intensity; see also Cramer-Rao Lower Bound (CRLB) in section 2.2.

2.1.3 Experimental Realization

To experimentally distribute the signal of the biological site over many frames, mostly the effect of photoswitching is used in which a fluorophore switches between the *on* and *off* state. Switching between the states is a stochastic event; however, the overall switching probabilities can often be influenced by experimental modification, i.e., varying laser powers or labeling strategies. The fluorophores can also enter an irreversible *off* state in which they are *bleached*. Single Molecule Localization Microscopy is the common denominator term for different experimen-

tal realizations of imaging *blinking* fluorophores, which in turn have individual names.

In Photo-activated Localization Microscopy (PALM), fluorescent proteins activated under UV light were used (Betzig et al. 2006, Hess, Girirajan, and Mason 2006, Biteen et al. 2008). Stochastic optical Reconstruction Microscopy (STORM) used dyes as synthetic fluorophores that can achieve photoswitching (Rust, Bates, and Zhuang 2006, Heilemann et al. 2008, Fölling et al. 2008).

Self-blinking dyes are another approach to SMLM. They intrinsically blink by switching between the fluorescent and non-fluorescent states without external activation.

The working principle of PAINT (Sharonov and Hochstrasser 2006) and DNA-PAINT (Schnitzbauer et al. 2017) is slightly different. Here, fluorophores do not photoswitch to achieve the necessary *blinking*, instead, they repeatedly bind and unbind to the target strain. Unbound fluorophores freely diffuse and lead to unspecific non-localized background.

Modelling Fluorophore photophysics For the purpose of a localization algorithm, the mechanism that leads to the blinking is of less concern than the explicit statistical properties of the emitters. Most importantly, the photon distribution, lifetime, and background characteristics are defined by the experimental realization and biological site are relevant to the localization algorithm. Following Annibale et al. 2011 and Sage et al. 2019, we can model the blinking dynamics of an individual emitter by the four-state model. Here, emitters turn from the initial *off-state* to a blinking state with a stochastic lifetime following an exponential distribution, afterward, they re-enter the off-state. In principle, these emitters could re-appear at a later point in the experiment however these long-ranged dynamics are usually not considered by localization algorithms, rendering the model an effective three-state model. In contrast, the high short-term temporal correlation due to the lifetime is utilized by many localization algorithm as a post-processing step, the *grouping*. Here, the localizations on neighboring frames are matched and then undergo weighted averaging (Chao, Ward, and Ober 2010, Li, Mund, et al. 2018). In *DECODE* (and *DECODE-Plex*) the high temporal correlation is explicitly used as *temporal context*; here, the input to the network is a window of three frames (Speiser et al. 2021).

2.1.4 3D Single Molecule Localization Microscopy

Any microscopy technique is at the advantage of being able to resolve complex biological structures (e.g., Nuclear Pore Complex (NPC)) in all three geometric dimensions. For SMLM, several approaches have been proposed which break the symmetry along the z-axis for out of focus emitters by altering the PSF in varios ways (PSF engineering) or multi-plane imaging. The most prominent variant of PSF engineered variants of SMLM for 3D imaging is the astigmatic approach following B. Huang et al. 2008. Here, a cylindrical lens is simply placed into the beam

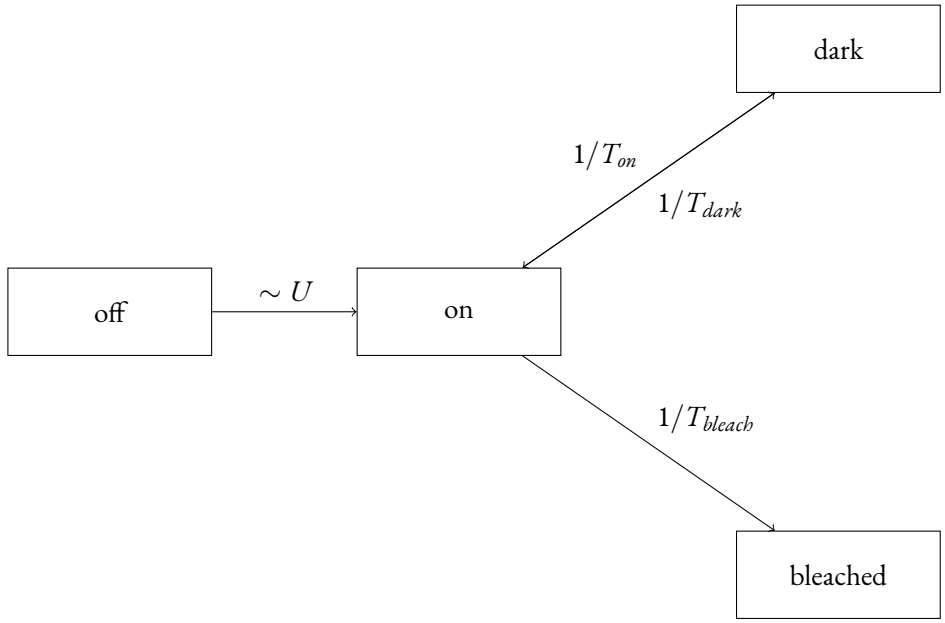


Figure 2.3: Four state emitter model following Sage et al. 2019. The emitter switches from the *off* to the *on* state following a uniform distribution. The emitter can then change either to the (reversible) dark or (irreversible) bleached state with the respective lifetimes.

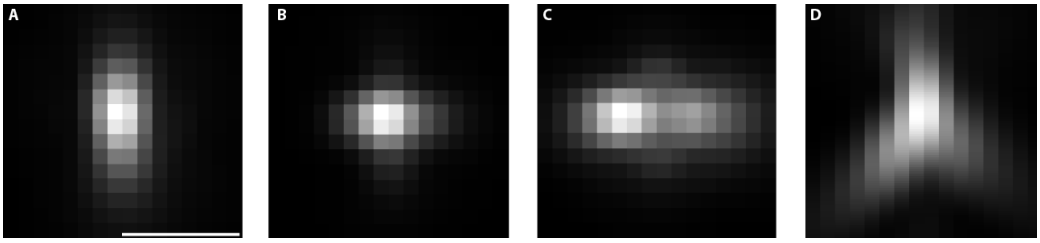


Figure 2.4: Astigmatic PSF. (a-c) xy slices below-, in-, and above the focal plane. The elongation axis of the PSF depends on the sign of z and breaks the symmetry as compared to 2D setups without a tube lens. Scalebar: 1 µm. **(d) xz slice.**

path, which elongates the PSF horizontally or vertically depending on the z position (see Figure 2.2 and 2.4). Other variants alter the PSF by introducing a phase mask or even using more complex optoelectronic devices (Pavani et al. 2009, Shechtman et al. 2014). Biplane 3D SMLM captures the PSF at different focal planes and can infer the axial position by the joint information of both channels or cameras (Juette et al. 2008).

2.1.5 Multi-color SMLM

Multi-color microscopy and multi-color SMLM in particular are common techniques to study the relations of different proteins at the same structure. In SMLM this can be achieved either by spectrally distinct fluorophores (Dempsey et al. 2011) and direct color assignment, or by the

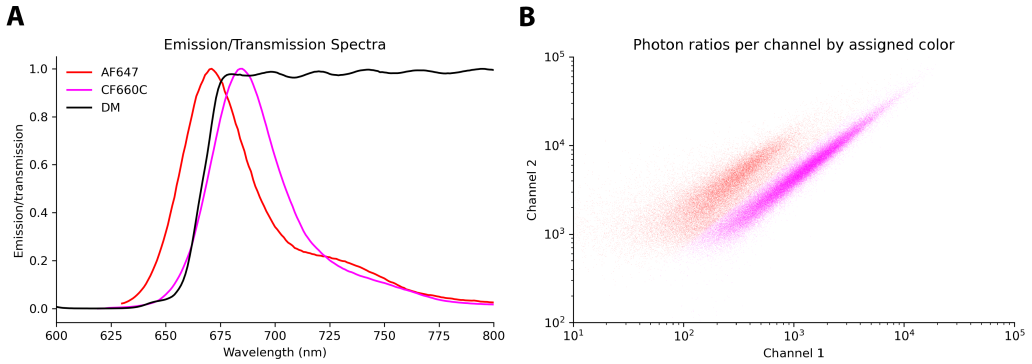


Figure 2.5: Multi-Color Photon Emission. (a) Photon Emission spectra for Alexa Fluor 647 (AF647), Cyanine-based Fluorescent 660 (CF660) and dichroic beamsplitter transmission (DM) characteristic. **(b) Experimental photon ratios for a dual-color experiment** with AF647, CF660. The scatterplot was color-coded by the color assignment.

ratiometric approach (Bossi et al. 2008, Lehmann et al. 2016, Y. Zhang et al. 2020).

Ratiometric Multi-Color SMLM Here, two or more colors are imaged in typically two channels/cameras with a dichroic mirror in the beam path (see Figure 2.2b). The color of the fluorophore is inferred by comparing the photon count ratio in both channels, which requires registration post-localization or simultaneous fitting of of all channels (Li, Shi, et al. 2022; see 2.1.7).

For the ratiometric approach, the photon count between two channels is transformed depending on the wavelength of the emitters and the characteristics of the dichroic mirror (see Figures 2.2b, 2.5a). Figure 2.5b shows an exemplary photon count distribution in a two-color experiment. Typically, the color-assignment is a downstream task in which the color is assigned by segmenting the experimental photon ratios into as many populations as there are colors. Moreover, if the true photon ratio for each color is known and the fitting algorithm outputs uncertainty estimates for the photon counts, a probabilistic color assignment can be performed, which is useful to filter out fuzzy localization (see methods of the DECODE-Plex manuscript). In biplane imaging, the photon ratio between the two channels will be constant and can be fully represented by channel-wise PSF normalization.

2.1.6 Spatial Transformation in Multi-Channel SMLM

For a typical multi-channeled SMLM setup like in Figure 2.2b, the spatial transform between a pair of channels can be described by a simple affine or projective transformation.

$$T = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{pmatrix} \tag{2.7}$$

$$\vec{r_2} = T\vec{r_1} \tag{2.8}$$

where $\vec{r_2}, \vec{r_1}$ are the first and second channel positions respectively. In case more than two channels are present, pairs of transformations between a shared reference, and the individual channels are computed.

In contrast to the PSF, the spatial transformation is not only implicitly presented to the Neural Network (NN) through sample generation, but made explicit as an additional input. We compute a *transformation indicator*, which simply comprises the offset between the first and second channels for all pixels and pass them along with the input frames.

2.1.7 High-Density, High-throughput SMLM

SMLM is an intrinsically *slow* technique as the distribution of the signal of a sample over time is at its core principle. Naturally, the acquisition time is directly linked to the density of emitters at which a sample is captured. The density of emitters can sometimes be predetermined by experimental factors (e.g., due to ultra-high labeling) or tuned during the acquisition by tuning the laser power so that the desired density is achieved. Notably, high-density acquisitions complicate the localization task. Classical MLE based fitting algorithms perform a peakfinding step before optimization. However, when emitters are in close proximity to one another, these problems are intertwined. An incorrect number of emitters in a Region of Interest (ROI) will lead to subpar optimization and even when the amount of emitters is correctly guessed, the optimization is non-convex and includes suboptimal minima. Interestingly, prior to DL based approaches, the best performing fitting algorithms for high-density datasets were in fact *low density* fitting algorithms (Sage et al. 2019). Recent simulation-based, deep learning approaches perform detection and localization simultaneously (Nehme, Weiss, et al. 2018; Nehme, Freedman, et al. 2020; Speiser et al. 2021).

In a laboratory environment, as far as high-throughput acquisitions are concerned, several other factors than the sole duration of a single acquisition must be considered. Besides increas-

ing the density of localizations, several other measures can be undertaken to maximize the utility of a SMLM setup. For example, automated (over-night) acquisitions can be of great utility even when the imaging time per sample is unchanged (Yan et al. 2018, Mund et al. 2018).

Besides changes to the experimental or labeling protocol, recent DL techniques that directly operate on the rendered microscopy image can dramatically speed up acquisition times (Ouyang, Aristov, et al. 2018a, Weigert, Schmidt, Boothe, et al. 2018).

Localization Algorithm

Key to SMLM is the precise localization of the emitters in each frame. Many algorithms have been proposed, making it difficult for users to choose among them. The *Super-resolution fight club* (Sage et al. 2019), often referred to as the *SMLM Challenge*, provides an extensive overview over 2D and 3D localization algorithms and evaluates them both on simulated as well as experimental data. *Classical* localization algorithm typically features a two-step *detection* and subsequent *localization* approach, where the detection is performed by a peakfinder and localization by Maximum-Likelihood-Estimation (MLE). Prominent members are *Thunderstorm* (Ovesný et al. 2014) as it is readily available as a *FiJi*-plugin (Schindelin et al. 2012). Other common algorithms, in particular for three-dimensional localization, include *cspline* (Babcock and Zhuang 2017) or the spline-based fitter integrated in the *SMAP* software (Li, Mund, et al. 2018, Ries 2020). Recent, DL based techniques perform detection and localization simultaneously as they extract the set of localizations in a post-processing step from a carefully crafted image- or voxel-based output representation (Nehme, Weiss, et al. 2018; Nehme, Freedman, et al. 2020; Speiser et al. 2021).

Rendering

The final step in SMLM is filtering localizations to throw out spurious or bad localized emitters and subsequent rendering to convert the set of localizations back to a super-resolved microscopy image. Typically, the localizations are plotted on a fine grid (around 10 nm) and plotted as 2D histograms (van Oijen et al. 1998, Hess, Girirajan, and Mason 2006) or are rendered using Gaussian kernels. Here, the width is proportional to a localization precision estimate (should the localization algorithm provide that) and intensity scaled with the photon count. Baddeley, Cannell, and Soeller 2010 show that the choice of the rendering algorithm has significant influence on the interpretability and achievable resolution and propose alternative techniques based on *quad-trees* and *delaunay triangulation* kernels.

2.2 Point Spread Function

As introduced in section 2.1.1, the PSF of an optical system is the system's response to a point source. In the context of such systems, the PSF and the resulting resolution of the image are tightly coupled. Modeling the PSF is vital to localization microscopy, as it is the basis for the very localization of the emitters. Any deviation between the unobservable true PSF and the modeled PSF will lead to biases in the localization and/or to increased localization errors (Liu et al. 2024). In experimental setups, the PSF will vary (1) over time due to temperature drifts or other environmental factors, (2) over the field of view due to optical aberrations and (3) over the depth of the sample due to refractive index mismatches.

Cramer-Rao Lower Bound (CRLB) The Cramer-Rao Lower Bound (CRLB) is often taken as a measure of localization precision in the field of SMLM. The background is that the CRLB describes the lower bound on the variance of an unbiased estimator and is computed by the inverse of the Fisher Information matrix I

$$\mathrm{cov}_\theta(T(X)) \geq \mathrm{I}^{-1}(\theta) \tag{2.9}$$

where θ is the parameter to be estimated and $T(X)$ is the (unbiased) estimator. The CRLB on the parameters θ are the diagonal elements of the inverse of the Fisher Information matrix. Importantly, that means that an *efficient estimator*, i.e. a good localization algorithm, should reach equality of Equation 2.9 under the knowledge of the true PSF and noise model. It should be stressed that any reported CRLB is under the assumption of the PSF model for which it is being calculated and is, therefore, not necessarily the true lower bound on the localization error since the true PSF is unknown. Moreover, the CRLB often can not be reliably computed in multi-emitter scenarios because the Fisher information matrix can be singular and inversion ill-defined in this case (Schoen 2014, Chao, Sally Ward, and Ober 2016).

If we neglect the Gaussian readout-noise (i.e., shot noise only; see section 2.3), we can write out the likelihood function L for the observed data x and our pixelated PSF model with mean values $\mu_k(x, y)$ with parameters θ as

$$L(\mu \mid \mathbf{x}) = \prod_k \frac{\mu_k^{x_k}}{x_k!} e^{-\mu_k} \tag{2.10}$$

with k being the pixel index. The Fisher information matrix I reduces to

$$I_{i,j} = \mathop{\mathbb{E}}_{\mathbf{x}} \left[\sum_k \left(\frac{x_k}{\mu_k} \frac{\partial \mu_k}{\partial \theta_i} - \frac{\partial \mu_k}{\partial \theta_i} \right) \left(\frac{x_k}{\mu_k} \frac{\partial \mu_k}{\partial \theta_j} - \frac{\partial \mu_k}{\partial \theta_j} \right) \right] \tag{2.11}$$

$$= \mathop{\mathbb{E}}_{\mathbf{x}} \left[\sum_k \left(\frac{x_k - \mu_k}{\mu_k} \right)^2 \frac{\partial \mu_k}{\partial \theta_i} \frac{\partial \mu_k}{\partial \theta_j} \right] \tag{2.12}$$

$$= \sum_k \frac{1}{\mu_k} \frac{\partial \mu_k}{\partial \theta_i} \frac{\partial \mu_k}{\partial \theta_j} \tag{2.13}$$

following Schoen 2014.

2.2.1 Point Spread Function Models

We introduce the most common PSF models to Single Molecule Localization Microscopy in the following.

Gaussian PSF For 2D SMLM, commonly, a 2D Gaussian PSF is used for localization, which is a reasonable approximation of the idealized Airy disk PSF. With the scalar version of the well-known Gaussian with mean value μ at $\mathbf{r}$ and parameters for $\theta : \left(\theta_x, \theta_y, \theta_{\text{phot}}, \theta_\sigma\right)$

$$\mu(\mathbf{r}; \theta) = \frac{\theta_{\text{phot}}}{2\pi\theta_\sigma^2} e^{\frac{-(x-\theta_x)^2-(y-\theta_y)^2}{2\theta_\sigma^2}} \tag{2.14}$$

Neglecting pixelation effects and background, the CRLB can then be easily estimated from the diagonal Fisher information matrix I following Equation 2.13

$$I = \text{diag}\left(\frac{\theta_{\text{phot}}}{\theta_\sigma^2}, \frac{\theta_{\text{phot}}}{\theta_\sigma^2}, \frac{1}{\theta_{\text{phot}}}, \frac{4\theta_{\text{phot}}}{\theta_\sigma^2} \right) \tag{2.15}$$

The CRLB is then

$$\text{cov}_\theta \geq \text{diag}\left(\frac{\theta_\sigma^2}{\theta_{\text{phot}}}, \frac{\theta_\sigma^2}{\theta_{\text{phot}}}, \theta_{\text{phot}}, \frac{\theta_\sigma^2}{4\theta_{\text{phot}}} \right) \tag{2.16}$$

which for θ_x, θ_y scales as expected with the inverse of the photon count, i.e., the localization precision scales with the inverse square root of the number of photons in accordance with Equation 2.5.

Spline-based PSF In recent years, spline-based PSF models have been proposed due to their generality and flexibility to adapt to arbitrary PSF without detailing any physical priors (Bab-

cock and Zhuang 2017, Li, Mund, et al. 2018). A spline is simply a piecewise polynomial function with smooth transitions between the pieces. In the three-dimensional form, a cubic spline-based model can be written as

$$\mu_{i,j,k}(x,y,z) = \theta_{phot} \sum_{m=0}^{3}\sum_{n=0}^{3}\sum_{p=0}^{3} a_{i,j,k,m,n,p} \left(\frac{x-x_i}{\Delta x}\right)^m \left(\frac{y-y_j}{\Delta y}\right)^n \left(\frac{z-z_k}{\Delta z}\right)^p \tag{2.17}$$

where $a_{i,j,k,m,n,p}$ are the spline coefficients, x_i, y_j, z_k are the spline centers, and $\Delta x, \Delta y, \Delta z$ are the respective voxel sizes. The Fisher information and hence the CRLB can be easily computed, as it reduces to sums of simple polynomial derivatives. One drawback of the spline-based PSF model is the great number of parameters, which amount to 64 coefficients per voxel, e.g. for a common calibration with ROI-size of 25x25 pixels in x, y, ± 1 µm range in z with step size 10 nm leading to $8 \cdot 10^6$ parameters.

Zernike-based PSF If the PSF is computed in the Fourier domain, i.e., by the pupil function with complex amplitudes, The PSF can be expressed elegantly based on Zernike polynomials. The Zernike polynomials form an orthogonal basis and are often used to describe optical aberrations. Compared to the spline-based PSF model, much fewer parameters are needed, and the model has a direct physical interpretation of the aberrations of the optical system. For example, the second order Zernike polynomials Z_2^2, Z_2^{-2} describe *Astigmatism*, the third order Z_3^1, Z_3^{-1} *Coma*.

The Zernike polynomials can be written as

$$Z_n^m(\rho,\phi) = R_n^m(\rho)\ \cos(m\,\phi) \tag{2.18}$$

$$Z_n^{-m}(\rho,\phi) = R_n^m(\rho)\ \sin(m\,\phi) \tag{2.19}$$

$$R_n^m(\rho) = \sum_{k=0}^{\frac{n-m}{2}} (-1)^k \binom{n-k}{k} \binom{n-2k}{\frac{n-m}{2}-k} \rho^{n-2k} \tag{2.20}$$

with ρ the radial coordinate and ϕ the azimuthal angle, m even and $-m$ odd polynomials, n the polynomial order, $n \geq m \geq 0$. The PSF μ for an emitter at position x_i, y_i, z_i is then

$$\mu\left(x-x_i, y-y_i, z_i\right) = \theta_{phot} \left[F^{-1}\left(h\left(k_x,k_y\right) e^{i2\pi k_z z_i} e^{i2\pi\left(k_x x_i + k_y y_i\right)}\right)\right]^2 \tag{2.21}$$

where h is the pupil function expressed by the Zernike base, $\vec{k}$ is the wave vector, following Liu et al. 2024.

2.2.2 Point Spread Function Retrieval

For 3D SMLM, the PSF model is often retrieved experimentally by measuring bright, immobilized calibration beads (Babcock and Zhuang 2017; Li, Mund, et al. 2018). We outline the retrieval procedure obtaining a *spline-based* PSF and *projective transformation* for the dual-channeled case following Li, Shi, et al. 2022.

Beads are acquired at different fields of view to ensure a reasonably robust PSF model. In the calibration procedure, for each channel, the beads are coarsely localized by peak finding and the ROI are registered and interpolated using 3D cross-correlation and cubic interpolation. Subsequently, the cubic spline model's parameters are extracted, and the bead raw data itself is fitted for validation. In the dual-channel case, the beads are linked across two channels, and the parameters of the projective transformation matrix are extracted. The beads are then aligned and interpolated, and the dual-channel PSF model is calculated.

In addition, recent methods have proposed to retrieve the PSF directly from in-situ data itself (Xu et al. 2020, Liu et al. 2024). When successful, this method is preferable over a calibration procedure not only because of reduced effort but also because sample the sample itself can alter the PSF slightly, which would be unnoticed in bead calibration procedure.

2.3 Camera Models

For SMLM, mostly EMCCD and sCMOS cameras are used. The resulting noise characteristics depend on which of the two camera types is used. Following F. Huang et al. 2013 the noise model is comprised of Shot noise, noise due to amplification gain (EMCCD only), and readout noise.

Shot noise For an expected photon count $\lambda_{0,k}$ in pixel k, quantum efficiency qe and spurious charge c_s, the observed signal follows the probability distribution $p(s_k)$ as

$$\lambda = \lambda_0 \cdot qe + c_s \tag{2.22}$$

$$p_{\text{shot}}\,(s_k) = \frac{\lambda^{sk} e^{-\lambda}}{s_k!}. \tag{2.23}$$

EMCCD Amplification gain EMCCD cameras amplify the signal by a multiplication electronic with variable gain factor $\theta = EM_{gain}$. The observed analog signal x is approximately a Gamma distribution

$$\rho_{\text{gain}}\,(x \mid k, \theta) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-\frac{x}{\theta}} \tag{2.24}$$

whith shape parameter $k = n_{e^-,0}$ being the *photoelectrons*.

Read noise The read noise commonly follows a Gaussian distribution with μ as the expected value in analog units after amplification (for EM-CCD) and variance σ^2

$$p_{\text{read}}(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.25}$$

For scientific Complementary Metal–Oxide–Semiconductor (sCMOS) cameras, the read noise should not assumed to be pixel/position independent as this can lead to biases in the localizations (F. Huang et al. 2013, Mandracchia et al. 2020). Equation 2.25 then changes to vectorial form.

Chapter 3

Manuscripts and Publications

In the following, we summarize and outline each of the (unpublished) manuscripts and publications that are part of or connected to this thesis and name the respective author's contributions.

3.1 Manuscripts

3.1.1 DECODE-Plex: high-density multi-channel single-molecule localization

This manuscript is in preparation.

The starting point of this project was the DECODE publication, which several groups have adopted for usage and even extended to field-dependent PSFs (Fu et al. 2023). To this day, an extension to multi-channeled data is missing but is necessary in biplane or multi-color imaging. The *naive* approach to a multi-channeled SMLM setup would be to fit each channel independently. However, linked prediction of all channels is to the advantage of improved localization precision in multi-color setups (Li, Shi, et al. 2022) and essential for biplane imaging.

DECODE-Plex aims to fill in this gap. Multi-channeled SMLM data comes with additional challenges not present to DECODE as we need to account for both the spatial and photon transformation between the two channels and accurately simulate their behavior (see sections 2.1.6 and 2.1.5).

We designed the neural network to feature three U-Net-like backbones (Ronneberger, Fischer, and Brox 2015). One per channel and one which combines both the channel dimension and the temporal dimension similar to the network architecture (see Figure 4 in the manuscript).

Since the image size at inference is potentially unknown or varying at training time, we train the model with a fixed ROI size of 80x80 pixel and input a spatial *transformation indicator* representing the spatially varying offsets between the channels in addition to the normal simulated or experimental SMLM frames. We sample the ROI's position across the extent of the camera chip and feed the network with the respective *transformation indicator* as input. At inference

time, the ROI's position of the experimental data is extracted, and the transformation indicator is fed accordingly.

We extensively validated DECODE-Plex performance with simulated and experimental data. Part of the simulated data was our submission to the SMLM challenge for biplane data where DECODE-Plex outperformed all other algorithms by a substantial margin. Furthermore, we simulated biplane and multi-color data across a wide range of densities and compared its performance against the state-of-the-art algorithm GlobLoc, which, similar to DECODE, allows for linked prediction of coordinates and other emitter attributes. GlobLoc, however, is based on a classical two-step peak finding and maximum likelihood technique. In addition, we acquired live-cell 3D biplane data; DECODE-Plex was able to resolve structures at unseen detail.

We took great care to develop DECODE-Plex in a robust and maintainable fashion. Upon publication, we will make the source code will be made available, including demonstration notebooks, tutorials, and detailed documentation.

Author contributions The manuscript is co-authored with Nestor Castillo (second), Arthur Jaques (third), Jakob Macke (shared corresponding author) and Jonas Ries (shared corresponding author).

Jonas Ries and Jakob Macke initiated the project. The idea for the network input representation to model the spatial transformations came from Lucas-Raphael Müller, Jakob Macke and Jonas Ries gave input to the validation of the multi-channel predictions and the design of the neural network backbone; the contributions to the scientific ideas amount to Lucas-Raphael Müller (60%), Jakob Macke (20%), Jonas Ries (20%). Nestor Castillo recorded the multi-color and biplane live cell data; the biplane challenge data is publicly available Sage et al. 2019. The contributions to data generation amount to Nestor Castillo (100%). Arthur Jaques helped implement the algorithm and software packaging. The contributions to the code base, analysis, and interpretation amount to Lucas-Raphael Müller (70%), Jonas Ries (15%), Jakob Macke (10%), Arthur Jaques (5%). Arthur Jaques prepared Figure 4. The contributions to manuscript writing and revision amount to Jakob Macke (5%), Arthur Jaques (5%), Jonas Ries (15%), Lucas-Raphael Müller (75%).

3.1.2 DECODE-OpenCloud: Sharing Computational Resources for Microscopy Data Analysis

This manuscript is in preparation.

The project's idea was to shed light on the provisioning and orchestration of computational resources for algorithms, particularly in the realm of microscopy. Increasingly complex and deep-learning heavy compute workflows pose significant requirements on computational se-

tups and access to hardware for researchers who want to use said algorithms. With this work, we focus on these aspects by outlining and providing infrastructure which makes it easier for researchers to make use of GPU-accelerated algorithms and developers to integrate programmatically with our proposed backend via an API. To showcase the applicability and generality of our approach, we interfaced three algorithms, namely *DECODE*, *DECODE-Plex* and *COMET* (Speiser et al. 2021; section 3.1.1; Reinkensmeier and Bates 2023).

Other approaches in the field focus on the provision of Jupyter or Colab notebooks, where the latter provide both the simple access of a Jupyter Notebook-like interface as well as computational resources at the discretion of Google. Another option is the use of Command-Line interfaces (CLIs). DECODE-OpenCloud differs in that it covers productionized interfaces (APIs) and well-defined and tested computing infrastructure for reliable, productive usage. While Jupyter notebooks provide easy access to researchers and programmers with various software engineering expertise, they often clutter over time, are not meant for programmatic calls, and lack in-depth support for means of modern software development, i.e., Continuous Integration and Continuous development tools. The Google Colab platform, in addition to rendering notebooks similar to Jupyter notebooks, also provides compute resources at the discretion of Google. Currently, Google Colab does not support all common package managers and compute environments change without further ado, rendering the platform unstable for permanent use.

In summary, our work contributes: (1) We outline a computational infrastructure for large-scale repetitive workflows where individuals and institutions can provide and consume computing resources. (2) We integrate three reference algorithms in the field of SMLM and invite researchers and developers to integrate more algorithms or make use of our API.

Author contributions The manuscript is co-authored with Arthur Jaques (shared first), Jonas Ries (shared corresponding) and Jakob Macke (shared corresponding). The conceptional idea of productionizing the DECODE-Plex algorithm and the general extension of such a framework came from me. Jakob Macke and Jonas Ries helped refine and apply them to the field. The contributions to scientific ideas amount to Lucas-Raphael Müller (70%), Arthur Jaques (20%), Jakob Macke (10%), and Jonas Ries (10%). There was no data generation applicable to this manuscript. Arthur Jaques helped implement the infrastructure code and test the framework. Analysis and interpretation amount to Lucas-Raphael Müller (60%), Arthur Jaques (20%), Jonas Ries (10%), Jakob Macke (10%). Arthur Jaques prepared Figures 1 and 2. The contributions to manuscript writing and revision amount to Jakob Macke (10%), Jonas Ries (10%), Arthur Jaques (20%), Lucas-Raphael Müller (60%).

3.1.3 Additional Contributions

The following publications are not a formal part of this thesis in that they partly overlap with the author's Master thesis (*Deep learning enables fast and dense single-molecule localization [...]*, Speiser et al. 2021) or were established during the author's doctoral studies but are only loosely connected to the methods and techniques developed this thesis focuses on (*Robust hand tracking [...]*, Müller et al. 2022).

Deep learning enables fast and dense single-molecule localization with high accuracy

This paper was published in September 2021 in Nature Methods (Speiser et al. 2021); Artur Speiser and I shared the first authorship.

The algorithm presented in this work, *DECODE*, laid the foundation of this study and tackled the issue of high-density SMLM fitting. In the past, most dedicated high-density algorithms based on a two-step procedure of detection and localization showed subpar performance due to the complexity and the coupling of both problems. With the rise of powerful Deep Learning based methods, our initial conceptual idea was to feature a combination of a Deep Learning fitting algorithm and a subsequent MLE based fitter. The motivation to introduce the latter was to have direct access to the theoretically motivated optimization procedure since both the model PSF model and the noise sources in SMLM are well understood (see chapter 2). They both are the necessary core components of the MLE optimization derivation. Direct access to the optimization is more difficult for the training of NN, especially in our case, where we predict a set of points represented in image space. On the other hand, DL based methods make incorporating auxiliary information easy if the information can be well represented. In the present case, the auxiliary information is simply passed as an additional input into the model. In the case of SMLM, this is particularly helpful since emitters across frames are highly correlated due to their stochastic nature of *on* and *off* switching (see section 2.1.3). We chose to incorporate a window of three frames to aid the prediction. While developing the DECODE algorithm for this work, we found no additional benefit of a downstream iterative fitting step. We speculate that this is due to two things: (A) the MLE fitting step does not have access to the temporal window but only to a single frame, (B) the general ill-posed iterative optimization procedure in multi-emitter configurations.

Besides incorporating temporal context, the representation of point sets in image space is a challenging task. Initially, we used pre-computed target maps, where offset vectors were placed in a 3x3 patch at the localization's position. Later, we changed the loss and the representation to the Gaussian model, which allowed for predictions, including uncertainties, which are particularly helpful for subsequent filtering. The backbone network consists of two U-Net blocks,

where the first one is fed individual frames (i.e., shared weights), and the latter one was designed to combine the temporal information, as it gets the concatenated output of the first network. Our network was evaluated on both simulated and experimental 2D and 3D data. Concretely, we showed its performance on high-density, ultra-high labeled data as well as on 2D live cell data. DECODE was packaged neatly to make it available to a broad audience in the realm of localization microscopy. Many groups have used and adapted DECODE since then.

Robust hand tracking for surgical telestration

This publication was authored and published in May 2022 in the Internal Journal of Computer-Assisted Radiology and Surgery as a proceeding of the International Conference on Information Processing in Computer-Assisted Interventions 2022 (Müller et al. 2022).

The conceptional idea of the project was to aid surgical training in minimally invasive surgeries by means of *surgical telestration*. Compared to state-of-the-art, which was largely based on traditional drawings, we extend the means of inter-surgeon communication by tracking the hand of a demonstrator and overlying it in the laparoscopic surgery screen. The overlay itself was formed by semantic segmentation. In addition to the semantic segmentation task, we perform keypoint regression on all fingers and the central hand joint to path the way for future downstream rendering of an artificial hand, providing more options in adaptive visualization. The Machine Learning backbone of our algorithm is a combination of a bounding box detection algorithm (a fine-tuned Yolov5s model; Ultralytics 2021), a keypoint regressor (a fine-tuned EfficientNet-B3; Lin et al. 2017) and a segmentation model (a fine-tuned FPN-EfficientNet 3 pre trained with noisy student; Tan and Le 2019; Xie et al. 2020). We performed extensive quantitative validation in a prospective validation study and compared the tracking performance to the well-established *MediaPipe* framework (F. Zhang et al. 2020). Our analysis respected the hierarchical structure of the data. We found that taken *as is*, MediaPipe showed significant performance penalties for dark-skinned individuals compared to light-skinned individuals. We found that MediaPipe performed worse on applications involving surgical gloves of any color.

Chapter 4

Conclusions

This work addressed the issue of fitting multi-channeled, high-density data in SMLM and the embedding of such an algorithm in translational work, i.e., to enable researchers to use the software easily and other developers to build upon it. Fitting localizations in SMLM is an inverse problem, as we are trying to find the positions of emitters that caused the observation we record by the camera. Motivated by the successful application of DECODE, a simulation-based DL algorithm, we hypothesized that a related approach could be a good starting point for multi-channeled localization microscopy data. DECODE has proven useful in fitting high-density single-channeled data and has since grown a substantial user base. We find our proposed algorithm, DECODE-Plex, to be successful in fitting multi-channeled localization microscopy data. We embedded it in the DECODE-OpenCloud framework for robust, repetitive workloads and ease of use. We want to discuss issues, limitations, and future work in the following.

4.1 DECODE-Plex

DECODE-Plex outperformed the previous state-of-the-art algorithm GlobLoc by a large margin at fitting high-density data. We compared its performance on both simulated and experimental data, which were comprised of high-density biplane and high-density multi-colored data. We found DECODE-Plex to consistently show superior detection performance and localization precision. In addition to improved localization results, DECODE-Plex compute intensity scales favorably with the input frame size, whereas classical fitting algorithms scale with the number of emitters, unfortunate in a high-density setting. We now discuss the limitations and future work specific to DECODE-Plex in more detail.

4.1.1 Simulation-based training

DECODE-Plex simulated training approach is its core component. Its performance is directly attributed to an accurate, well-designed simulation pipeline. We found that the simulation-based training approach works well in training our DL backbone when subsequently applied to experimental data. For a model trained by supervised learning, a simulation-based approach is merely the only possibility due to the lack of ground-truth labels or the possibility of human annotation. In our case, the most critical components of the simulation pipeline are emitter sampling, the Point Spread Function, and the camera simulation. We want to discuss these integral aspects:

Emitters Simulating the distribution of the emitters is the first core component of the simulation pipeline, as it defines the distribution of the target parameters the localization algorithms aim to predict. Their most important attributes are the position, photon count, and time of appearance.

The emitter's position is defined by a structure from which we sample. To avoid hallucination biases in our predictions, we decided against sampling from biological structures (see section 4.1.3). The photon count of an emitter is determined by the lifetime before switching off and the photon flux distribution. Arguably, the precise value of the lifetime is not relevant for our purposes as it is described by an exponential distribution, which will lead to a broad distribution of emitters, greatly varying in their specific lifetime. In a multi-color scenario, different colors have different lifetimes; however, this is not reflected in our simulation pipeline since we train the model with independent photon counts in each channel. In real SMLM experiments, fluorophores can re-enter an on-state after a long off-period. We did not model this behavior as our model is only presented to a temporal context of three consecutive frames and should not learn longer-range temporal dynamics. In a multi-color experiment, the photon flux will be a multi-modal distribution with different modes determined by the respective labeling. For that reason, we simply chose a uniform photon flux distribution.

Point Spread Function The PSF is the core component of the simulation that describes the optical setup and the response of the microscope to a point emitter. A mismatch between the true PSF and the PSF used to train DECODE-Plex can lead to artifacts or unnecessarily increased variance (see section 4.1.2, Figure 9 in the DECODE-Plex manuscript). Varying PSFs are also one of the contributors to why DECODE-Plex at this time does not generalize among multiple microscopes or can even be used pre trained (see section 4.1.3).

Spatial Transformation The precise knowledge of the spatial transformation between the channels and a recent PSF calibration is vital. Indeed, we found false positive localizations when an incorrect transformation was presented to the network. In that case, DECODE-Plex tends to output one localization per channel instead of combining them. The resulting rendered superresolution image would then show bad channel separation and ghost structures.

Latency The wait time between an experimental acquisition and fitted data is a major contributing factor to the user who chooses among different localization software. One of the major limitations of DECODE-Plex is its latency induced by the retraining of the model to ensure the closeness of our simulated distribution to the experimental data. The two main factors necessitating a fresh retraining are a different emitter distribution (i.e., photon/background count) and/or a different PSF. This is in contrast to MLE based fitting algorithm where the user simply specifies a calibration file with no additional upfront delay. Particularly in a high-throughput setting, users rather expect real-time results, which can currently, only be provided by classical MLE based fitting algorithm. We propose several future strategies to mitigate the delay in section 4.1.3.

4.1.2 Gridding Artefact

In certain experimental data, we identified the presence of a gridding artifact. We found this artifact in particularly adverse conditions, most prominently for out-of-focus localizations and regions of high local density, for rather dim localizations or combinations of all aforementioned cases. Under common conditions, i.e., for a sufficiently large biological structure, there should not be a strong non-uniformity in the localizations offsets with respect to the pixel center since the pixel grid and any biological structure will almost never share a common frame of reference. Thus, the gridding artifact can be easily quantified by histogramization of the subpixel position. Under these assumptions, any non-uniformity in the histogram indicates biased localizations caused by uneven recall or biased localization. The artifact is problematic because it could be confused with a biological structure. This particularly holds true for structures of a similar spatial frequency, for example, when actin-spectrin structures are imaged, which are spaced at 190 nm. Their scale is not too far from the gridding artifact, spaced in pixel size (often in the order of 100 nm). When the rendering algorithm takes the uncertainty estimate of the localizations into account, the visual presence of the gridding artifact reduces. As a mitigation step, we introduced a debiasing algorithm to actively shift the localizations to form uniform histograms. Comparing pre- and post debiased localizations, we found that not all localizations were shifted by much less than their predicted uncertainty. This indicates that the model underreports its error in these cases and points to a more fundamental problem of the output representation or

simulation procedure.

Possible Sources In principle, this gridding artifact could be due to either a reduced number of localizations at the pixel borders, and/or a biased localization towards the pixel centers. To assess this, we purposefully trained a DECODE-Plex network with a wrong PSF and assessed the output localizations. (Figure 9 of the DECODE-Plex manuscript). This is the first indication that, in fact, *gridding* is largely caused by positionally biased localizations. Still, it should be noted that experimental data is intrinsically more complicated and lacks access to ground truth. Additional factors for experimental data are not limited to but include non-uniform background distributions, PSF mismatch because of field-dependent aberrations or refractive index mismatch. Lelek et al. 2021 show that these artifacts appear even for MLE based fitting algorithm. However, we found far less pronounced artifacts when comparing DECODE-Plex to the GlobLoc algorithm which merely showed increased variance or rejected localizations under these conditions. We speculate that the said artifacts are an implicit consequence of our chosen data representation.

4.1.3 Future Work

We want to outline future ideas to mitigate the aforementioned limitations of our work or to increase the applicability and useability of our software. We identify the generalization aspects as the most important future directions, which, when solved, would drastically reduce the number of retraining necessary. Such a generalizing model would enable zero-latency application of it to new experimental data. We want to elaborate on the dimensions of generalization in the present case and other applicational and workflow aspects.

Emitter Distribution As outlined, we train DECODE-Plex from the ground up for every acquisition in an almost 1:1 training to inference setting with the rationale to achieve optimal predictions for each dataset. This is a somewhat unusual setting in the field, with ever larger models even prohibitive in areas like Natural Language Processing (NLP) (OpenAI et al. 2024). Furthermore, it induces the latency for the user to wait for the finished training. Since the PSF varies on a larger time scale than the emitter's distribution in photon counts and background, future work should be conducted on models trained with a broader emitter distribution first. Such a model could potentially be *informed* about the statistics of the data present, similar to the spatial transformation indicator we introduced.

PSF Generalization Future work should be conducted on training the DECODE-Plex network on a multitude of PSF and informing the model at inference time about the present PSF

in the experimental data. This procedure will depend on an appropriate sampling at training time and a way to inform the model about the PSF present in experimental data. Both, the introduction of the *transformation indicator* as well as a variant of DECODE with *field-dependent* PSFs (Fu et al. 2023) are steps in a similar direction. Arguably a physics-inspired representation like the *zernike-based* PSF (see section 2.2.1) with few necessary parameters is likely better suited than a voxelated or spline-based PSF with millions of parameters, both for the sampling procedure and to inform the model.

Data Representation The data representation of point clouds without finite support is a challenge when combined with conventional two- or three-dimensional convolutional networks that are otherwise more commonly used for pixel-wise predictions. We got the first indication that our chosen representation is a contributing factor to the presence of the observed gridding artifact. Future work should elaborate on different representations of point sets without finite support that are better targeted for the problem at hand. They should support, in principle, arbitrary localization precision (i.e., not limited by the pixel size of typically 100 nm), and not lead to similar biases. In related work, Boyd et al. 2018 chose a representation that maps convolutional features in image space to a fully connected output, effectively enforcing a list representation with fixed in and output capacity in which it is the network's responsibility to learn an image to list translation. Lee et al. 2019 and Biloš and Günnemann 2021 introduce more generalized, improved representations of permutation invariant densities, which could be well suited for localization microscopy data.

Structured Sampling Strategy Instead of sampling the emitter's position from an uninformative, 3D uniform distribution, incorporating prior structural information would be interesting. Ouyang, Aristov, et al. 2018b show that with the incorporation of structural priors, fewer localizations are needed to achieve similarly resolved superresolution images as compared to localization algorithms without such priors. We argue that the challenge here is the acceptance of users towards such algorithms incorporating biologically inspired structural priors since hallucination or other structured artifacts can significantly undermine trust in these algorithms. An alternative possibility would be a semi-supervised or self-supervised variant of our algorithm in which DECODE-Plex would be fed with a representation of its previous localizations.

Integration and Workflow The use of our algorithm will depend on its demonstrated scientific success and, to a large extent, on the ease of usage. To this day, the Thunderstorm algorithm (Ovesný et al. 2014) is one of the most widely used algorithms in the community because of its easy integration into the Fiji software, even though it shows subpar performance when

compared to other algorithms (Sage et al. 2019). Further thought should be spent on integrating DECODE-Plex nicely into a daily lab routine. With DECODE-OpenCloud, major steps towards large-scale repetitive workflows have been undertaken; see section 4.2. In the future, DECODE-Plex could be deeply integrated into scientific software widely used in the field, like *SMAP* (Ries 2020), *Napari* (Sofroniew et al. 2024) or *Fiji* (Schindelin et al. 2012).

4.2 DECODE-OpenCloud

We designed DECODE-OpenCloud to integrate scientific software into routine, large-scale workflows. Our hybrid solution, combining cloud-based orchestration with local and cloud compute runners, ensures high availability and efficient use of local compute power. We release DECODE-OpenCloud to (1) invite researchers to use the software already integrated, (2) developers to integrate more scientific software related to the field, and (3) to take DECODE-OpenCloud infrastructure as inspiration. In the following, we want to discuss particular limitations of DECODE-OpenCloud and potential future aspects.

4.2.1 Limitations

Large data Handling large data is generally challenging but particularly when multiple computing locations are involved. This holds true for DECODE-OpenCloud as well. Our hybrid model, i.e., the combination of local and cloud-backed compute resources, faces this challenge in particular. The necessity for transferring large data varies depending on the algorithm. The presently implemented DECODE and DECODE-Plex algorithms do not necessitate any large data transfer during the training procedure since the training is simulation-based. We argue that models can be applied to local computing hardware as the inference step poses much fewer requirements on the available GPU memory and processing power. Other algorithms, however, will need a significant amount of data for training and/or inference, which would be difficult to realize in a hybrid cloud setting due to the transfer limits and costs (see section 4.2.2).

Monitoring Live monitoring and debugging are more challenging for a remote system like DECODE-OpenCloud as compared to interactive approaches like Google Colab. Here, users can directly with the graphical interface, which makes debugging easier. In recent years, ML monitoring tools like *weights and biases* Biewald 2020 allow sophisticated ML experiment tracking and even remote cancellation. Integrating such a tool into the algorithms, rather than into DECODE-OpenCloud is more apropriate since DECODE-OpenCloud provides a job orchestration and compute platform, it does notintend to modify the algorithms themselves. We implemented basic job monitoring in DECODE-OpenCloud to inform users about failed jobs,

but we do not support interactive compute sessions.

4.2.2 Future Work

We plan to integrate more algorithms into DECODE-OpenCloud and invite contributions from other developers. We also encourage developers of integrated microscopy software suites to use DECODE-OpenCloud programmatically via its API.

Further investigation is needed for optimized data handling in multi-location compute settings. As DECODE-OpenCloud gains popularity, additional measures to distribute compute resources fairly will become important.

4.3 Common Conclusion

In this study, we presented DECODE-Plex, an algorithm for high-density multi-channel Single Molecule Localization Microscopy validated on simulated and experimental data. DECODE-Plex outperformed the previous state-of-the-art MLE based fitting algorithm GlobLoc (Li, Shi, et al. 2022). DECODE-Plex's main challenge is its output representation, partly leading to structural artifacts that require postprocessing and its limited generalization capability across varied experimental conditions and for different microscopes on which future work should be conducted.

We identify the translational aspects in setting up ML and DL algorithms in microscopy as vital for their applicability and accessibility to a broad range of users. To address this, we introduced DECODE-OpenCloud, a framework developed to enhance the accessibility and scalability of ML algorithms in microscopy, with DECODE-Plex integrated into it. With DECODE-OpenCloud, we presented an orchestrator for compute jobs that allows researchers to use the implemented algorithms via a website or programmatically via its API. It follows a hybrid approach, distributing compute jobs to unused local compute or cloud-backed compute runners to ensure high availability and efficient use of resources. DECODE-OpenCloud allows for easy integration of further algorithms that do not rely on heavy data transfer.

We hope the community adopts DECODE-Plex and DECODE-OpenCloud for challenging localization microscopy data, enabling new sample regimes, improved imaging speed, and increased throughput.

Glossary

DECODE A 3D high density fitting algorithm for SMLM, in particular for astigmatic and double-helix PSFs (Speiser et al. 2021).

DECODE-OpenCloud A hybrid-cloud environment where researchers can share compute resources and algorithms in the realm of microscopy (see section A.2).

DECODE-Plex A multiplexed, multi-channel high density fitting algorithm for SMLM, in particular for multi-color and biplane applications (see section A.1).

GlobLoc A MLE based algorithm for multi-channeled SMLM data (Li, Shi, et al. 2022).

SMAP Software framework for SMLM which integrated many algorithms (Ries 2020).

SMLM challenge An ongoing challenge meausring the performance of localization algorithm for SMLM against simulated and experimental data (Sage et al. 2019).

Acronyms

AF647 Alexa Fluor 647.

CF660 Cyanine-based Fluorescent 660.

CRLB Cramer-Rao Lower Bound.

DL Deep Learning.

DNA-PAINT DNA points accumulation for imaging in nanoscale topography.

DNN Deep Neural Network.

LLM Large-Language Model.

MINFLUX Minimal emission fluxes.

ML Machine Learning.

MLE Maximum-Likelihood-Estimation.

NLP Natural Language Processing.

NN Neural Network.

NPC Nuclear Pore Complex.

PAINT Point Accumulation for Imaging in Nanoscale Topography.

PALM Photo-activated Localization Microscopy.

PSF Point Spread Function.

REST-API Representational State Transfer Application Programming Interface.

ROI Region of Interest.

sCMOS scientific Complementary Metal–Oxide–Semiconductor.

SIM Structured Illumination Microscopy.

SMLM Single Molecule Localization Microscopy.

STED Stimulated Emission Depletion.

STORM Stochastic optical Reconstruction Microscopy.

Bibliography

Abbe, E. (1873). "Beiträge Zur Theorie Des Mikroskops Und Der Mikroskopischen Wahrnehmung". In: *Archiv für Mikroskopische Anatomie* 9.1, pp. 413–418. DOI: 10.1007/BF02956173.

Annibale, Paolo et al. (July 2011). "Quantitative Photo Activated Localization Microscopy: Unraveling the Effects of Photoblinking". In: *PLOS ONE* 6.7, e22678. DOI: 10.1371/journal.pone.0022678.

Babcock, Hazen P. and Xiaowei Zhuang (Apr. 2017). "Analyzing Single Molecule Localization Microscopy Data Using Cubic Splines". In: *Scientific Reports* 7.1, p. 552. DOI: 10.1038/s41598-017-00622-w.

Baddeley, David, Mark B Cannell, and Christian Soeller (Feb. 2010). "Visualization of Localization Microscopy Data". In: *Microscopy and Microanalysis* 16.1, pp. 64–72. DOI: 10.1017/S143192760999122X.

Balzarotti, Francisco et al. (Feb. 2017). "Nanometer Resolution Imaging and Tracking of Fluorescent Molecules with Minimal Photon Fluxes". In: *Science* 355.6325, pp. 606–612. DOI: 10.1126/science.aak9913.

Berg, Stuart et al. (Dec. 2019). "Ilastik: Interactive Machine Learning for (Bio)Image Analysis". In: *Nature Methods* 16.12, pp. 1226–1232. DOI: 10.1038/s41592-019-0582-9.

Betzig, Eric et al. (Sept. 2006). "Imaging Intracellular Fluorescent Proteins at Nanometer Resolution". In: *Science* 313.5793, pp. 1642–1645. DOI: 10.1126/science.1127344.

Biewald, Lukas (2020). *Experiment Tracking with Weights and Biases*. URL: https://www.wandb.com/.

Biloš, Marin and Stephan Günnemann (June 2021). *Scalable Normalizing Flows for Permutation Invariant Densities*. arXiv: 2010.03242 [cs, stat]. URL: http://arxiv.org/abs/2010.03242 (visited on 05/13/2024).

Biteen, Julie S. et al. (Nov. 2008). "Super-Resolution Imaging in Live Caulobacter Crescentus Cells Using Photoswitchable EYFP". In: *Nature Methods* 5.11, pp. 947–949. DOI: 10.1038/nmeth.1258.

Bossi, Mariano et al. (Aug. 2008). "Multicolor Far-Field Fluorescence Nanoscopy through Isolated Detection of Distinct Molecular Species". In: *Nano Letters* 8.8, pp. 2463–2468. DOI: 10.1021/n1801471d.

Boyd, Nicholas et al. (Feb. 2018). "DeepLoco: Fast 3D Localization Microscopy Using Neural Networks". In: DOI: 10.1101/267096.

Buchholz, Tim-Oliver et al. (2020). "DenoiSeg: Joint Denoising and Segmentation". In: *Computer Vision – ECCV 2020 Workshops*. Ed. by Adrien Bartoli and Andrea Fusiello. Cham: Springer International Publishing, pp. 324–337. DOI: 10.1007/978-3-030-66415-2_21.

Chao, Jerry, E. Sally Ward, and Raimund J. Ober (July 2016). "Fisher Information Theory for Parameter Estimation in Single Molecule Microscopy: Tutorial". In: *Journal of the Optical Society of America A* 33.7, B36. DOI: 10.1364/JOSAA.33.000B36.

Chao, Jerry, E. Sally Ward, and Raimund J. Ober (July 2010). "A Software Framework for the Analysis of Complex Microscopy Image Data". In: *IEEE transactions on information technology in biomedicine: a publication of the IEEE Engineering in Medicine and Biology Society* 14.4, pp. 1075–1087. DOI: 10.1109/TITB.2010.2049024.

Deguchi, Takahiro et al. (Mar. 2023). "Direct Observation of Motor Protein Stepping in Living Cells Using MINFLUX". In: *Science* 379.6636, pp. 1010–1015. DOI: 10.1126/science.ade2676.

Dempsey, Graham T. et al. (Dec. 2011). "Evaluation of Fluorophores for Optimal Performance in Localization-Based Super-Resolution Imaging". In: *Nature Methods* 8.12, pp. 1027–1036. DOI: 10.1038/nmeth.1768.

Deschamps, Joran, Markus Mund, and Jonas Ries (Feb. 2017). "Towards Quantitative High-Throughput 3D Localization Microscopy". In: *Biophysical Journal* 112.3, 141a–142a. DOI: 10.1016/j.bpj.2016.11.781.

Diekmann, Robin et al. (Sept. 2020). "Optimizing Imaging Speed and Excitation Intensity for Single-Molecule Localization Microscopy". In: *Nature Methods* 17.9, pp. 909–912. DOI: 10.1038/s41592-020-0918-5.

Flemming, Walther (Dec. 1880). "Beiträge Zur Kenntniss Der Zelle Und Ihrer Lebenserscheinungen: Theil II". In: DOI: 10.1007/bf02952594.

Fölling, Jonas et al. (Nov. 2008). "Fluorescence Nanoscopy by Ground-State Depletion and Single-Molecule Return". In: *Nature Methods* 5.11, pp. 943–945. DOI: 10.1038/nmeth.1257.

Franken, Linda E. et al. (2020). "A Technical Introduction to Transmission Electron Microscopy for Soft-Matter: Imaging, Possibilities, Choices, and Technical Developments". In: *Small* 16.14, p. 1906198. DOI: 10.1002/smll.201906198.

Fu, Shuang et al. (Mar. 2023). "Field-Dependent Deep Learning Enables High-Throughput Whole-Cell 3D Super-Resolution Imaging". In: *Nature Methods* 20.3, pp. 459–468. DOI: 10.1038/s41592-023-01775-5.

Gustafsson, Mats G. L. et al. (June 2008). "Three-Dimensional Resolution Doubling in Wide-Field Fluorescence Microscopy by Structured Illumination". In: *Biophysical Journal* 94.12, pp. 4957–4970. DOI: 10.1529/biophysj.107.120345.

Heilemann, Mike et al. (2008). "Subdiffraction-Resolution Fluorescence Imaging with Conventional Fluorescent Probes". In: *Angewandte Chemie International Edition* 47.33, pp. 6172–6176. DOI: 10.1002/anie.200802376.

Hell, Stefan W. and Jan Wichmann (June 1994). "Breaking the Diffraction Resolution Limit by Stimulated Emission: Stimulated-Emission-Depletion Fluorescence Microscopy". In: *Optics Letters* 19.11, p. 780. DOI: 10.1364/OL.19.000780.

Hess, Samuel T., Thanu P.K. Girirajan, and Michael D. Mason (Dec. 2006). "Ultra-High Resolution Imaging by Fluorescence Photoactivation Localization Microscopy". In: *Biophysical Journal* 91.11, pp. 4258–4272. DOI: 10.1529/biophysj.106.091116.

Hooke, Robert (1665). *Micrographia, or Some Physiological Descriptions of Minute Bodies Made by Magnifying Glasses, with Observations and Inquiries Thereupon. By R. Hooke.* London: Printed by JoMartyn, and JaAllestry, printers to the Royal Society.

Huang, Bo et al. (Feb. 2008). "Three-Dimensional Super-Resolution Imaging by Stochastic Optical Reconstruction Microscopy". In: *Science* 319.5864, pp. 810–813. DOI: 10.1126/science.1153529.

Huang, Fang et al. (July 2013). "Video-Rate Nanoscopy Using sCMOS Camera–Specific Single-Molecule Localization Algorithms". In: *Nature Methods* 10.7, pp. 653–658. DOI: 10.1038/nmeth.2488.

Hung, Jane et al. (July 2020). "Keras R-CNN: Library for Cell Detection in Biological Images Using Deep Neural Networks". In: *BMC Bioinformatics* 21.1, p. 300. DOI: 10.1186/s12859-020-03635-x.

Isensee, Fabian et al. (Dec. 2020). "nnU-Net: A Self-Configuring Method for Deep Learning-Based Biomedical Image Segmentation". In: *Nature Methods*. DOI: 10.1038/s41592-020-01008-z.

Juette, Manuel F. et al. (June 2008). "Three-Dimensional Sub–100 Nm Resolution Fluorescence Microscopy of Thick Samples". In: *Nature Methods* 5.6, pp. 527–529. DOI: 10.1038/nmeth.1211.

Klar, Thomas A. and Stefan W. Hell (July 1999). "Subdiffraction Resolution in Far-Field Fluorescence Microscopy". In: *Optics Letters* 24.14, pp. 954–956. DOI: 10.1364/OL.24.000954.

Kluyver, Thomas et al. (2016). "Jupyter Notebooks – a Publishing Format for Reproducible Computational Workflows". In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS Press, pp. 87–90. DOI: 10.3233/978-1-61499-649-1-87.

Krull, Alexander, Tim-Oliver Buchholz, and Florian Jug (June 2019). "Noise2Void - Learning Denoising From Single Noisy Images". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, pp. 2124–2132. DOI: 10.1109/CVPR.2019.00223.

Lee, Juho et al. (May 2019). *Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks*. DOI: 10.48550/arXiv.1810.00825. arXiv: 1810.00825 [cs, stat].

Lehmann, Martin et al. (2016). "Novel Organic Dyes for Multicolor Localization-Based Super-Resolution Microscopy". In: *Journal of Biophotonics* 9.1-2, pp. 161–170. DOI: 10.1002/jbio.201500119.

Lelek, Mickaël et al. (June 2021). "Single-Molecule Localization Microscopy". In: *Nature Reviews Methods Primers* 1.1, pp. 1–27. DOI: 10.1038/s43586-021-00038-x.

Li, Yiming, Markus Mund, et al. (Apr. 2018). "Real-Time 3D Single-Molecule Localization Using Experimental Point Spread Functions". In: *Nature Methods* 15.5, pp. 367–369. DOI: 10.1038/nmeth.4661.

Li, Yiming, Wei Shi, et al. (June 2022). "Global Fitting for High-Accuracy Multi-Channel Single-Molecule Localization". In: *Nature Communications* 13.1, p. 3133. DOI: 10.1038/s41467-022-30719-4.

Lin, Tsung-Yi et al. (July 2017). "Feature Pyramid Networks for Object Detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944. DOI: 10.1109/CVPR.2017.106.

Liu, Sheng et al. (June 2024). "Universal Inverse Modeling of Point Spread Functions for SMLM Localization and Microscope Characterization". In: *Nature Methods* 21.6, pp. 1082–1093. DOI: 10.1038/s41592-024-02282-x.

Lord Rayleigh F.R.S. (1879). "XXXI. Investigations in Optics, with Special Reference to the Spectroscope". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 8.49, pp. 261–274. DOI: 10.1080/14786447908639684.

Lukosz, W. and M. Marchand (July 1963). "Optischen Abbildung Unter Überschreitung Der Beugungsbedingten Auflösungsgrenze". In: *Optica Acta: International Journal of Optics* 10.3, pp. 241–255. DOI: 10.1080/713817795.

Mandracchia, Biagio et al. (Jan. 2020). "Fast and Accurate sCMOS Noise Correction for Fluorescence Microscopy". In: *Nature Communications* 11.1, p. 94. DOI: 10.1038/s41467-019-13841-8.

McQuin, Claire et al. (July 2018). "CellProfiler 3.0: Next-generation Image Processing for Biology". In: *PLOS Biology* 16.7. Ed. by Tom Misteli, e2005970. DOI: 10.1371/journal.pbio.2005970.

Midtvedt, Benjamin et al. (Dec. 2022). "Single-Shot Self-Supervised Object Detection in Microscopy". In: *Nature Communications* 13.1, p. 7492. DOI: 10.1038/s41467-022-35004-y.

Mortensen, Kim I. et al. (May 2010). "Optimized Localization Analysis for Single-Molecule Tracking and Super-Resolution Microscopy". In: *Nature Methods* 7.5, pp. 377–381. DOI: 10.1038/nmeth.1447.

Müller, Lucas-Raphael et al. (Aug. 2022). "Robust Hand Tracking for Surgical Telestration". In: *International Journal of Computer Assisted Radiology and Surgery* 17.8, pp. 1477–1486. DOI: 10.1007/s11548-022-02637-9.

Mund, Markus et al. (Aug. 2018). "Systematic Nanoscale Analysis of Endocytosis Links Efficient Vesicle Formation to Patterned Actin Nucleation". In: *Cell* 174.4, 884–896.e17. DOI: 10.1016/j.cell.2018.06.032.

Nehme, Elias, Daniel Freedman, et al. (June 2020). "DeepSTORM3D: Dense 3D Localization Microscopy and PSF Design by Deep Learning". In: *Nature Methods*. DOI: 10.1038/s41592-020-0853-5.

Nehme, Elias, Lucien E. Weiss, et al. (Apr. 2018). "Deep-STORM: Super-Resolution Single-Molecule Microscopy by Deep Learning". In: *Optica* 5.4, p. 458. DOI: 10.1364/OPTICA.5.000458.

OpenAI et al. (Mar. 2024). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs]. URL: http://arxiv.org/abs/2303.08774 (visited on 07/04/2024).

Ouyang, Wei, Andrey Aristov, et al. (May 2018a). "Deep Learning Massively Accelerates Super-Resolution Localization Microscopy". In: *Nature Biotechnology* 36.5, pp. 460–468. DOI: 10.1038/nbt.4106.

– (2018b). "Deep Learning Massively Accelerates Super-Resolution Localization Microscopy". In: *Nature Biotechnology* 36.5, pp. 460–468. DOI: 10.1038/nbt.4106.

Ouyang, Wei, Florian Mueller, et al. (Dec. 2019). "ImJoy: An Open-Source Computational Platform for the Deep Learning Era". In: *Nature Methods* 16.12, pp. 1199–1200. DOI: 10.1038/s41592-019-0627-0.

Ovesný, Martin et al. (May 2014). "ThunderSTORM: A Comprehensive ImageJ Plug-in for PALM and STORM Data Analysis and Super-Resolution Imaging". In: *Bioinformatics* 30.16, pp. 2389–2390. DOI: 10.1093/bioinformatics/btu202.

Pavani, Sri Rama Prasanna et al. (Mar. 2009). "Three-Dimensional, Single-Molecule Fluorescence Imaging beyond the Diffraction Limit by Using a Double-Helix Point Spread Func-

tion". In: *Proceedings of the National Academy of Sciences* 106.9, pp. 2995–2999. DOI: 10.1073/pnas.0900245106.

Reinkensmeier, Lenny and Mark Bates (May 2023). *Gpufit/Comet*. URL: https://github.com/gpufit/Comet (visited on 07/13/2024).

Ries, Jonas (Sept. 2020). "SMAP: A Modular Super-Resolution Microscopy Analysis Platform for SMLM Data". In: *Nature Methods* 17.9, pp. 870–872. DOI: 10.1038/s41592-020-0938-1.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, pp. 234–241.

Rust, Michael J., Mark Bates, and Xiaowei Zhuang (Oct. 2006). "Sub-Diffraction-Limit Imaging by Stochastic Optical Reconstruction Microscopy (STORM)". In: *Nature Methods* 3.10, pp. 793–796. DOI: 10.1038/nmeth929.

Sage, Daniel et al. (May 2019). "Super-Resolution Fight Club: Assessment of 2D and 3D Single-Molecule Localization Microscopy Software". In: *Nature Methods* 16.5, pp. 387–395. DOI: 10.1038/s41592-019-0364-4.

Schindelin, Johannes et al. (July 2012). "Fiji: An Open-Source Platform for Biological-Image Analysis". In: *Nature Methods* 9.7, pp. 676–682. DOI: 10.1038/nmeth.2019.

Schmidt, Uwe et al. (2018). "Cell Detection with Star-Convex Polygons". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Ed. by Alejandro F. Frangi et al. Cham: Springer International Publishing, pp. 265–273. DOI: 10.1007/978-3-030-00934-2_30.

Schnitzbauer, Joerg et al. (June 2017). "Super-Resolution Microscopy with DNA-PAINT". In: *Nature Protocols* 12.6, pp. 1198–1228. DOI: 10.1038/nprot.2017.024.

Schoen, Ingmar (Nov. 2014). "Localization Precision in Stepwise Photobleaching Experiments". In: *Biophysical Journal* 107.9, pp. 2122–2129. DOI: 10.1016/j.bpj.2014.09.035.

Sharonov, Alexey and Robin M. Hochstrasser (Dec. 2006). "Wide-Field Subdiffraction Imaging by Accumulated Binding of Diffusing Probes". In: *Proceedings of the National Academy of Sciences* 103.50, pp. 18911–18916. DOI: 10.1073/pnas.0609643104.

Shechtman, Yoav et al. (Sept. 2014). "Optimal Point Spread Function Design for 3D Imaging". In: *Physical Review Letters* 113.13, p. 133902. DOI: 10.1103/PhysRevLett.113.133902.

Sofroniew, Nicholas et al. (July 2024). *Napari: A Multi-Dimensional Image Viewer for Python*. Zenodo. DOI: 10.5281/zenodo.12608849.

Speiser, Artur et al. (Sept. 2021). "Deep Learning Enables Fast and Dense Single-Molecule Localization with High Accuracy". In: *Nature Methods* 18.9, pp. 1082–1090. DOI: 10.1038/s41592-021-01236-x.

Tan, Mingxing and Quoc Le (May 2019). "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, pp. 6105–6114. URL: https://proceedings.mlr.press/v97/tan19a.html (visited on 07/16/2024).

Ultralytics (2021). *YOLOv5: A State-of-the-Art Real-Time Object Detection System*. URL: https://docs.ultralytics.com.

van Oijen, A. M et al. (July 1998). "3-Dimensional Super-Resolution by Spectrally Selective Imaging". In: *Chemical Physics Letters* 292.1, pp. 183–187. DOI: 10.1016/S0009-2614(98)00673-3.

Weigert, Martin, Uwe Schmidt, Tobias Boothe, et al. (Dec. 2018). "Content-Aware Image Restoration: Pushing the Limits of Fluorescence Microscopy". In: *Nature Methods* 15.12, pp. 1090–1097. DOI: 10.1038/s41592-018-0216-7.

Weigert, Martin, Uwe Schmidt, Robert Haase, et al. (Mar. 2020). "Star-Convex Polyhedra for 3D Object Detection and Segmentation in Microscopy". In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE Computer Society, pp. 3655–3662. DOI: 10.1109/WACV45572.2020.9093435.

Wu, Yu-Le et al. (Jan. 2023). "Maximum-Likelihood Model Fitting for Quantitative Analysis of SMLM Data". In: *Nature Methods* 20.1, pp. 139–148. DOI: 10.1038/s41592-022-01676-z.

Xie, Qizhe et al. (June 2020). "Self-Training With Noisy Student Improves ImageNet Classification". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695. DOI: 10.1109/CVPR42600.2020.01070.

Xu, Fan et al. (May 2020). "Three-Dimensional Nanoscopy of Whole Cells and Tissues with in Situ Point Spread Function Retrieval". In: *Nature Methods* 17.5, pp. 531–540. DOI: 10.1038/s41592-020-0816-x.

Yan, Rui et al. (Mar. 2018). "Spectrally Resolved and Functional Super-resolution Microscopy via Ultrahigh-Throughput Single-Molecule Spectroscopy". In: *Accounts of Chemical Research* 51.3, pp. 697–705. DOI: 10.1021/acs.accounts.7b00545.

Zhang, Fan et al. (2020). "MediaPipe Hands: On-device Real-time Hand Tracking". In: *arXiv preprint arXiv:2006.10214*. arXiv: 2006.10214.

Zhang, Yongdeng et al. (Feb. 2020). "Nanoscale Subcellular Architecture Revealed by Multicolor Three-Dimensional Salvaged Fluorescence Imaging". In: *Nature Methods* 17.2, pp. 225–231. DOI: 10.1038/s41592-019-0676-4.

Appendix A

Appendices

A.1 DECODE-Plex: high-density multi-channel single-molecule localization

DECODE-Plex: high-density multi-channel single-molecule localization

Lucas-Raphael Müller[1,2],
Nestor Miguel Castillo Duque de Estrada[3], Arthur Jaques[1,2],
Jakob H. Macke[1,6*], Jonas Ries[1,2,3,4,5*]

[1]Machine Learning in Science, Excellencecluster Machine Learning, Tübingen, Germany.
[2]Cell and Biophysics, European Molecular Biology Laboratory, Heidelberg, Germany.
[3]Max Perutz Labs, Vienna Biocenter Campus, Vienna, Austria.
[4]Department of Structural and Computational Biology, University of Vienna, Vienna, Austria.
[5]Faculty of Physics, University of Vienna, Vienna, Austria.
[6]Tübingen AI Center, University of Tübingen, Tübingen, Germany.

*Corresponding author(s). E-mail(s): jakob.macke@uni-tuebingen.de; jonas.ries@maxperutzlabs.ac.at;
Contributing authors: lucas.mueller@embl.de;

Abstract

Single-molecule localization microscopy (SMLM) excels in imaging and differentiating cellular structures of different kinds at nanometer resolution, but the need for sparse emitters in standard analysis slows down imaging and limits labeling density. We have created DECODE-Plex, a deep-learning-based framework to localize dense single molecules with overlapping point spread functions simultaneously in multiple channels. It precisely models the photophysical and optical factors of multi-channel systems. We validated DECODE-Plex on experimental live-cell bi-plane and ultra-high density dual-color data. Packaged for ease of use and installation and made available as open source, DECODE-Plex will enable many groups to improve the imaging speed and quality of multi-channel SMLM.

Keywords: SMLM, Multi-Channel, Multi-Color, Bi-Plane, Deep Learning, Simulation, GPU

1 Introduction

Single-molecule localization microscopy (SMLM) is a super-resolution method that can resolve cellular structures at nanometer resolution[1–3]. The basis of SMLM is to label structures of interest with switchable fluorophores and to acquire many (tens of thousands) camera frames in which only a small fraction of the fluorophores is activated to avoid overlapping emitters. The position of these sparse emitters is then localized using, e.g., maximum-likelihood fitting [4–6]. The need for sparsity and, consequently, many camera frames leads to poor temporal resolution and low throughput. This major limitation of SMLM was recently diminished by using deep learning algorithms like DeepSTORM[7], DeepSTORM3D[8] and DECODE [9], which allow for fitting fluorophores at high densities with overlapping point-spread functions (PSFs), outperforming even dedicated multi-emitter fitting algorithms[10].

However, none of these deep-learning-based approaches are compatible with multi-channel SMLM approaches, including multi-color, multi-plane, or 4Pi SMLM.

Multi-color SMLM allows for the visualization of the molecular organization of different target molecules relative to each other by spectrally distinct color separation[11] or *ratiometric* post-hoc color-assignment, in which the photon count ratios in two channels are compared [12–14]. 3D SMLM either uses *engineered* PSFs to break the symmetry around the focus (e.g. astigmatic [15], double-helix [16] or tetrapod [17]), or two or more focal positions imaged simultaneously in multiple channels (biplane SMLM [18]). Algorithms that fit multi-channeled data individually per channel cannot extract the full information and do not reach the best possible localization precision described by the Cramér–Rao lower bound (CRLB) [6]. This limitation was overcome with the GlobLoc approach ([6]) that uses maximum likelihood estimation on all channels simultaneously to maximize precision. However, it is not compatible with high emitter densities beyond the single-molecule regime.

Here, we present DECODE-Plex, a new multi-channel high-density fitting algorithm for multi-color and biplane SMLM. DECODE-Plex consistently achieves better detection performance and localization precision in biplane and multi-color scenarios compared to glocLoc. It performs detection, localization, and combination of all channel information simultaneously while allowing for flexible linking of the photon count estimates depending on whether dual-channel multi-color or biplane data is present.

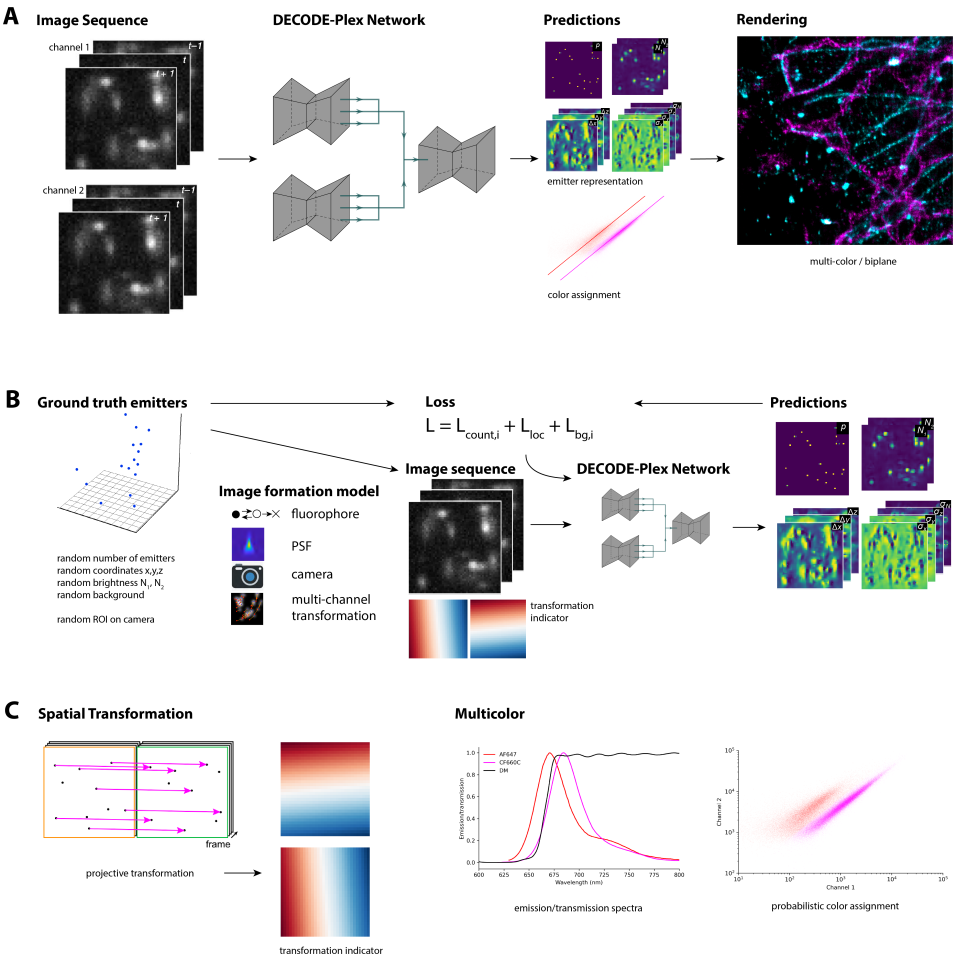


Fig. 1: **DECODE-PLEX for high-density single-molecule localization. (a) DECODE-Plex architecture.** The network uses information from multi-channel input simultaneously; it maps its predictions onto frames representing a probability for detection, the photon count in each channel, uncertainties related to these predictions, and a background. For biplane SMLM the photon counts are linked, for multi-color SMLM the photon counts in each channel are predicted and color assignment is performed by likelihood estimation (see c). **(b) Training sequence.** DECODE-Plex is trained with simulated data. Emitters are sampled randomly, in addition a position on the camera is sampled to cover all possible positions at inference time. Synthetic images are generated by the image formation model, which uses an auxiliary *transformation indicator* that represents the pixel-wise displacements between the channels. **(c) Details of multi-channel transformation logic** for spatial and photon transformation. The *Spatial Transformation* is defined by a projective transformation matrix that maps the localizations from channel N to channel 1. For each pixel and channel, the spatial displacement with respect to the reference channel is computed, and a transformation indicator map is generated. The *Multicolor Photon Transformation* is used to assign a color to a localization based on the ratio of estimated photon counts in each channel.

2 Results

DECODE-Plex builds up on the architecture of DECODE [9] and introduces a new representation for input and output. In contrast to DECODE, DECODE-Plex utilizes information from multiple input channels simultaneously. Since the position of the ROI on the camera can be unknown at training time, the channels are linked by a spatial *transformation indicator* that is provided to the model as input (see Figure 1 (b)). It indicates the spatial offset in each pixel between the target and the reference channel and allows DECODE-Plex to fit experimental data of different sizes flexibly. As DECODE, DECODE-Plex also evaluates the temporal context, i.e., it considers that the same fluorophore can be visible in several adjacent camera frames. For each frame fed through the network, the network outputs a detection probability, positions in x/y/z using offset vectors with respect to the pixel center, and a photon count prediction per channel (independent photon counts in each channel for multi-color data) or a linked photon count for biplane data. For the latter biplane mode, we sample and predict a single total photon count; the simulator distributes the photons to both channels with respect to the normalization of the PSF determined during calibration. A respective uncertainty estimate accompanies each output predictor. The uncertainty estimates of the photon count, in combination with the expected ratio between the dark and bright channels, allow for the probabilistic assignment of the emitters' color.

We developed and evaluated DECODE-Plex both on multi-color and biplane data (see Figures 2,3) and compared its performance to the state-of-the-art *GlobLoc* algorithm [6], which allows linking of the fitted parameters across channels. To quantify the performance on simulated data, we followed the metrics established by the *SMLM Challenge* [10] which is an ongoing benchmark evaluation to aid users in the choice of localization algorithm. We extended these metrics for multi-color data (see section 4.4) and report the *color assignment accuracy* (eq. 4.4.1) and *rejection rate* (eq. 4.4.1), i.e., the fraction of correctly assigned colors and the fraction of rejected color assignments due to fuzzy predictions.

On simulated low-density data, DECODE-Plex approaches the combined Cramér–Rao Lower bound (CRLB) (Figure 7), i.e., the best theoretically possible precision and thus performs equally well as *GlobLoc* [6]. Furthermore, for the simulated data, DECODE-Plex outperforms GlobLoc across all multicolor and biplane data density regimes in all metrics (Figure 2). DECODE-Plex inference time scales mainly with the frame size, GlobLoc's computation time scales with the number of emitters, leading to subpar computation times for high-density regimes. For the remaining biplane modality of the *SMLM Challenge*, we see substantial improvement in the detection and localization performance (see Figure 2e)

Next, we evaluated DECODE-Plex on experimental data and compared its performance to GlobLoc. As a high-density dual-color sample, we imaged microtubules (labeled with CF660C) and the endoplasmic reticulum (AF647) in a 3D astigmatic dual-channel setup (Figure 3 (a-b)) at a density about one order of magnitude denser than we usually use for single-emitter SMLM with GlobLoc [6]. We trained DECODE-Plex using background and photon count parameters extracted from a pre-fit of a small subset of the data using GlobLoc. We used a dual-channel spline-based PSF calibrated with bead stacks.

For these experimental high-density datasets, we consistently find better results with DECODE-Plex than with GlobLoc; in particular, regions with high local density were significantly better resolved with DECODE-Plex. Thus, DECODE-Plex enables imaging of ultra-densely labeled samples, which, even without UV activation, exceed the single-emitter regime (Figure 3). Compared to measurements at single-emitter densities, DECODE-Plex reduces the imaging time by around one order of magnitude.

These high imaging speeds are critical for live-cell measurements, where the time resolution has to match the biological process under investigation. We demonstrated the performance of DECODE-Plex by imaging microtubules in living cells in 3D with the bi-plane approach. The microtubules were labeled with the self-blinking fluorophore HMSiR-tubulin at high concentrations ([19]). Indeed, we could observe rearrangements of the microtubule cytoskeleton on the nanometer scale with 12.5s temporal resolution. Compared to imaging in the single-emitter regime, DECODE-Plex allowed us to increase the labeling density by one order of magnitude without increasing the light dose. This is important, as even in our measurements, phototoxicity was the limiting factor, perturbing the microtubule organization after approx. 3 minutes of imaging.

3 Discussion

We presented DECODE-Plex, a simulation-trained deep learning algorithm for high-density multi-channel SMLM applications. DECODE-Plex differs from previous algorithms in that it can simultaneously process multi-channel information and temporal context and perform detection and localization. DECODE-Plex outperformed the state-of-the-art GlobLoc algorithm for the biplane and multi-color modality by a large margin, underlining the effectiveness of simultaneous, end-to-end prediction and utilization of temporal and multi-channel information. DECODE-Plex's inference speed scales with frame size and training times are in the order of 6 hours.

Machine Learning-based fitting algorithm [7–9] incorporate the PSF implicitly into the neural network's parameters during training by consistently using training samples generated with a specific PSF. To the best of our knowledge, there is no Machine Learning based fitting algorithm yet, that allows for arbitrary experimental data without retraining and adapting to an unseen PSF. In the case of DECODE-Plex, a new model must be trained for a new PSF (compare Figure 9). Other important simulation parameters comprise the photon and background distribution; however, we argue that these can be trained broadly and refined for specific data. Arguably, a semi-automated retraining procedure of DECODE-Plex upon microscope re-calibration is advised.

DECODE-Plex's performance heavily relies on an accurate PSF model to achieve precise localizations. Many algorithms in SMLM suffer from biased localizations when the PSF model for the optimization step differs from the actual, unknown PSF [20]. Similarly, Machine Learning based fitters [7–9] perform poorly when the PSF used for training does not match the PSF during the fitting step. We observe a similar artifact under hard conditions, particularly for dim, far out-of-focus localizations or localizations in dense regions. Part of the DECODE-Plex is an optional post-processing histogram equalization procedure, in which the fitted localizations

are de-biased according to their uncertainty to form a uniform histogram across the subpixel position (see 4.2.2 and Figure 8).

DECODE-Plex will enable a large and growing community to perform high-density localization microscopy, greatly increasing throughput, imaging speed, and new sample regimes. We provide packaged and pre-compiled software that requires little prior knowledge. DECODE-Plex is accompanied by a detailed set of documentation and tutorials, allowing all users to benefit from our framework.

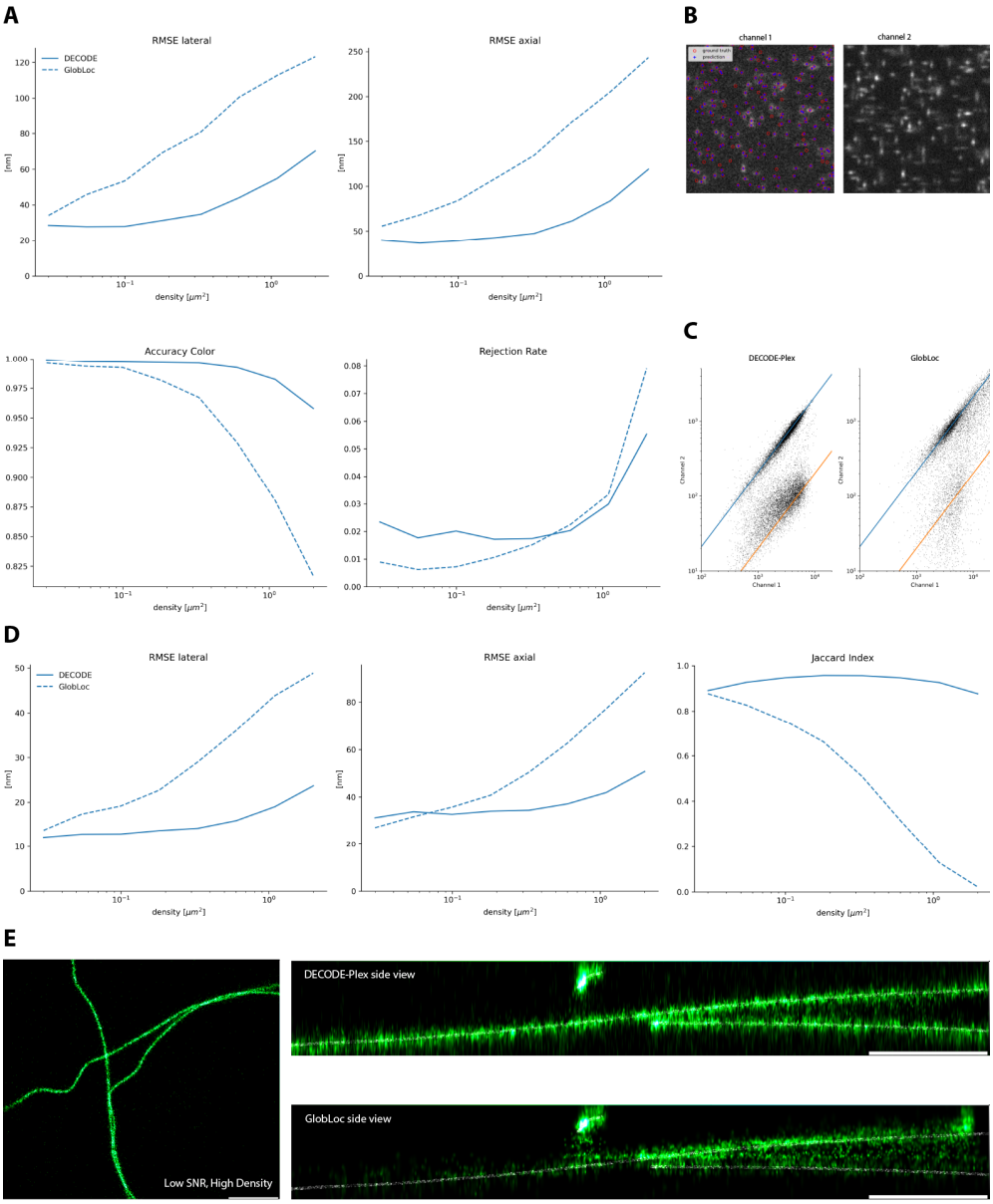


Fig. 2: **Performance on simulated high-density data. (a) Performance on multi-color data** using an experimentally extracted transformation and the photon ratios for AF647 and CF680 (ratios 0.21 and 0.02 dark to bright channel for our microscope setup); RMSE lateral and axial, Accuracy of color prediction and the fraction of rejected emitters due to a fuzzy color assignment **(b) Dual-channel, dual-color high-density frames (c) Predicted photon distribution of DECODE and GlobLoc**, ground truth photon count ratios marked as solid lines (AF647 and CF680). **(d) Performance on simulated biplane dataset** using an experimentally extracted transformation and calibrated PSF. Localization performance for simulated bi-plane data. RMSE lateral and axially and Jaccard index. Example dual-channel frames for two densities. **(e) Biplane challenge results.** DECODE-Plex and the GlobLoc algorithm reconstructions on the high density, low signal bi-plane challenge data (validation data). Overview on the left in xy-view and side-view for DECODE-PLEX and GlobLoc, respectively. Scale bars, 1 um.

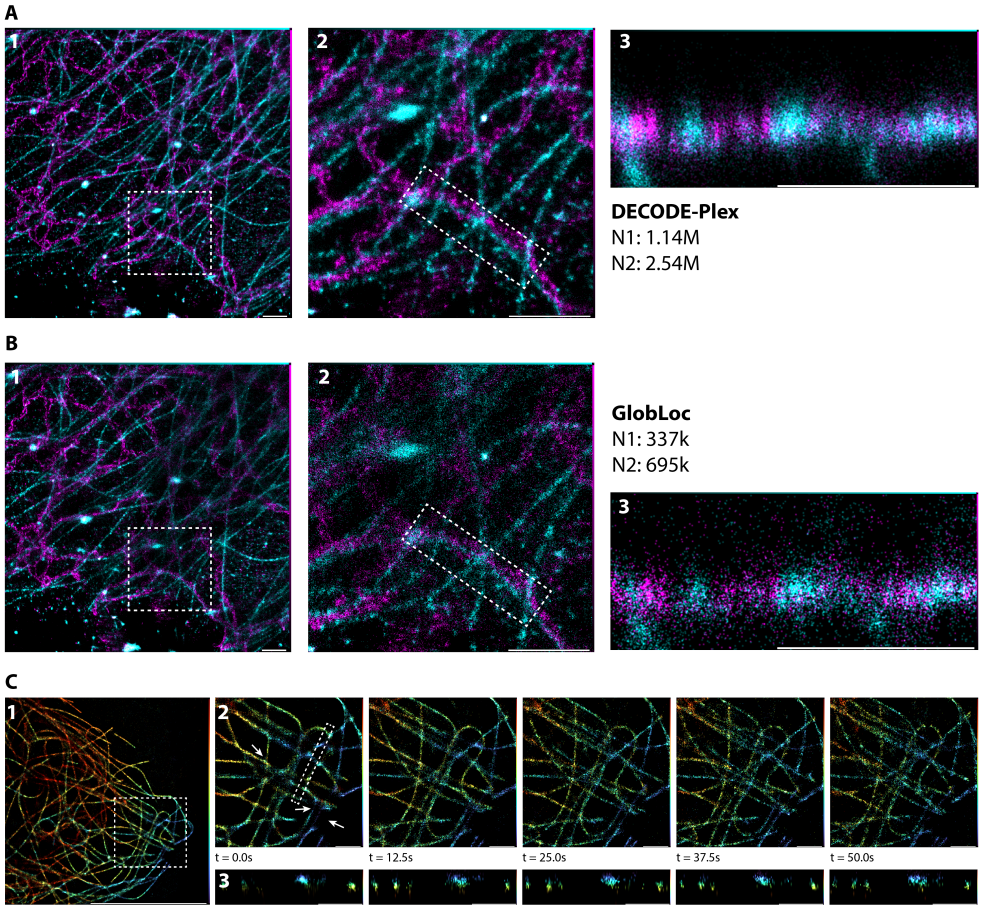


Fig. 3: **Performance on experimental high-density data. (a) Reconstructions of DECODE-PLEX and (b) GlobLoc on experimental high-density data** microtubules (CF660) and the endoplasmic reticulum (AF647). Overview, zoom and side view. Scale bars 1 μm (1,2,3). Localizations were debiased (DECODE-Plex) and drift-corrected (DECODE-Plex/GlobLoc) and filtered on localization precision below 30 nm **(c) Fast live-cell SMLM**, microtubule labelled with self-blinking fluorophore HMSiR-tubulin.

4 Methods

4.1 DECODE-PLEX network architecture

Our architecture builds upon a combination of modified 2D U-Net [21] models as the neural network's backbone (see Figure 4). In contrast to the original implementation, we use an exponential linear unit activation function [22].

DECODE-Plex uses one U-Net-like model per imaging channel. The frames of the temporal window share the same model and are fed in a sliding window approach. The spatial transformation indicator is fed as additional input and is made available to all models.[1] The channel dimension is then stacked and fed through a shared U-Net that combines both the multi-channel and the temporal information. The up-and-down sampling stages consist of three steps with 64 initial features, doubling/halving the features per stage.

Our network aims to predict (1) a probability map p_k of having found an emitter (following a Bernoulli distribution) in near proximity, (2) the photon count per channel $N_{k,i}$, the three-dimensional displacements with respect to the center of the pixel $(\Delta_x, \Delta_y, \Delta_z)$, respective uncertainties for both photon and displacements $(\sigma_{x,k}, \sigma_{y,k}, \sigma_{z,k}, \sigma_{N,k,i})$ and a background prediction for each channel. Notably, the predicted positions are always with respect to one reference coordinate system for which we conveniently, yet arbitrarily, choose the first channel's position.

4.1.1 Loss function for multi-channeled detection, localization, and uncertainty estimates

We build upon *DECODE's*[9] loss function as a combination of count loss L_{count}, localization loss L_{loc} and background loss L_{bg}.

$$L = L_{\text{count}} + L_{\text{loc}} + L_{\text{bg}} \tag{1}$$

The photon count in the localization and background loss is extended to multiple channels in case the photon count is multi-channeled (i.e. for multi-color). In biplane mode, we link the photon count but still use a multi-channeled background prediction and, therefore, multi-channeled background loss.

$$L_{\text{loc}} = -\frac{1}{E} \sum_{e=1}^{E} \log \sum_{k=1}^{K} \frac{p_k}{\sum_j p_j} P\left(\mathbf{u}_e^{\text{GT}} \mid \boldsymbol{\mu}_k, \Sigma_k\right) \tag{2}$$

4.1.2 Training details

Our model is trained purely on live simulated training data (see 4.3). We perform training on 80x80 pixel-sized frames to give enough space for positional transformation between the channels. We sample the position on the camera to accommodate for the fact of varying positional transformations at different positions of the multi-camera

[1] In total, in a two-camera/channel setup, one input sample consists of 8 input channels (2ch·3win+2aux).

setup (see 4.3.4). We use the AdamW optimizer [23] with a group learning parameter of $1 \cdot 10^{-4}$ to $4 \cdot 10^{-4}$ depending on SNR and modality for the network parameters. We decay the learning rate by a factor of 1/2 scheduled by the ReduceLROnPlateau[2] to dynamically accommodate for differences in the difficulty of the training. We exclude dim emitters with less than 100 photons (summed over all channels) as training targets yet still simulate them.

4.2 Localization extraction and Post-Processing

4.2.1 Localization Extraction

Localization candidates are extracted from the frame representation by thresholding the probability output p_k. Ideally, p_k is sparse and well-separated into a bi-modal distribution (0 and 1), and its sum corresponds to the true number of emitters. Small probability clusters can emerge under hard conditions instead of single sparse probability outputs. These local neighborhoods are integrated and form a cumulative probability threshold for localization extraction. The localization offsets and the uncertainty predictors are extracted at the thresholded pixels, and the localization position is computed as the pixel center positions plus the respective offsets.

4.2.2 De-Biasing

Under hard conditions, the network tends to bias the localization position with respect to the within-pixel position (see Figures 8, 9). The effect can be measured by computing the subpixel position histogram across the x and y dimensions. We allow for optional post-processing of these localizations to equalize the histogram. We observed the biasing to be correlated with the z position and, therefore, performed histogram equalization in steps. Typically, we choose steps of size 50 nm. In each z step, we histogramize the localizations according to their subpixel offset and rescale their subpixel offsets weighted by the localizations' lateral uncertainty estimate.

4.2.3 Color Assignment

We perform the color assignment as a post-processing step. The emission wavelengths and beamsplitter characteristics determine the (true) photon count ratio between the two channels. The ratio can be empirically measured or theoretically computed from the specification of the labels and optical components.

Since DECODE-Plex outputs uncertainty measures for the photon counts, we can assign the color probabilistically using maximum likelihood estimation given the true photon ratios. The probability p_k of assigning an emitter to color k is then

$$p_k = \frac{L_k}{\sum_{j=1}^{C} L_j} \tag{3}$$

where L_k are the likelihoods of the observed photon ratios p_i/p_{tot} with uncertainty estimates $p_{\sigma,i}$ given the true ratios r_k for colors k and channels i.

[2] https://pytorch.org/docs/2.1/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

4.3 Training Data Simulation

We continuously simulate training data for DECODE-Plex and use each sample only once to avoid overfitting the models' backbone. The performance of DECODE-Plex is, therefore, tightly linked to the closeness to the experimental data of interest to said simulation and the accuracy of the image formation model.

4.3.1 Point-Spread-Function

The PSF describes the image formed by the characteristics of the microscope and the object in the object plane ($\mathrm{PSF}(\mathbf{r})$ and $O(\mathbf{r})$). The resulting observed image is the convolution of both:

$$I(\mathbf{r}) = O(\mathbf{r}) \circledast \mathrm{PSF}(\mathbf{r}) \tag{4}$$

We use Cubic-Spline PSFs as an accurate yet almost arbitrarily flexible model of arbitrary PSFs. We follow [5] and [4] and use our custom, optimized CUDA-kernel, which allows us to simulate 100 000 frames in less than 0.5s (on an RTX 4090). Since the choice of the center of the PSF is arbitrary for a spline-modelled PSF, we estimate the PSF and the multi-channel transformation jointly, as these are not independent of one another.

4.3.2 Camera

DECODE-Plex does not differentiate between a multi-camera setup and a multi-ROI, single-camera setup to model multiple channels. In all scenarios, we apply global, pixel-value preserving transformations to maximize the spatial overlap of all channels. This particularly includes de-mirroring and global shifts by multiples of pixels. The procedure ensures that the positional differences between the images of the same source emitter in the object plane are not too far apart for the Convolutional Neural Network to combine all information. Note that the effective receptive field of the CNN is typically far more narrow than the theoretical one [24].

4.3.3 Camera Noise

Following ref. [10], we model the camera noise model by its Poissonian shot noise due to the stochastic nature of the photon interaction with the camera chip, EMCCD amplification noise due to gain which follows an approximate Gamma distribution and additive readout noise which a Gaussian distribution can reasonably approximate.

4.3.4 Multi-channel transformation

We assume the positional transformation in the multi-channel setting to follow an affine or projective transformation T.

$$\vec{r_2} = T \cdot \vec{r_1} \tag{5}$$

$$T = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{pmatrix} \tag{6}$$

where $\vec{r_2}$ are the coordinates in the second and $\vec{r_1}$ the coordinates in the reference channel.

T is estimated jointly with the multi-channeled PSF following the GlobLoc routine [6]. Both the PSFs and the transformation will typically extracted from a bead calibration. However, performing a single-channel fit and refining the transformation based on experimental data afterward is also possible.

At training time, we do not enforce any constraints on the relation between the photon counts in the respective channels to avoid biasing the photon predictions. Instead, we sample the photon count independently in multi-color mode. In biplane mode, only a single photon count is sampled and distributed across the channels according to the PSF's normalization.

4.3.5 ROI Sampling

Since the frame size can vary between training and inference, we feed the transformation of the experimental data at inference time as an auxiliary frame for which we compute the displacement between the reference and the respective channels in x and y for each pixel.

We sample the simulated camera position at training time to vary the ROI and displacement between the reference and respective channels (see 4.1.2).

4.3.6 Photophysics

We sample each fluorophore's photon flux ψ and initial appearance t_0 from a Uniform distribution U. With a uniform distribution, we argue that there is less risk of bias toward one of multiple colors in a multi-color experiment which typically peak at different photon flux counts and have different lifetimes. In the case of a multi-color experiment, where the photon counts are unlinked, we sample the photon flux per channel independently. Furthermore, we draw its on-time from an exponential distribution parametrized by λ. The emitters are then distributed and discretized over the frames, which converts the photon flux by integration over time to a photon count. We do not model long-term re-appearance.

4.3.7 Estimating simulation parameters

Commonly, only the background and photon distribution will vary significantly between different SMLM experiments on a daily timescale. Covering the entire background regime is vital such that the model does not confuse background fluorescence with actual emitters. We found no penalty in training in a broad background regime. We propose one of the two options for regular usage of our algorithm: (A) Either train a broad distribution of photon and background and refine on data of particular interest or (B) perform a pre-fit with GlobLoc [6] and use these parameters for training a specific DECODE-Plex model directly. With larger changes to the PSF (e.g., due to changes in the optical setup), new training is mandatory. We advise a semi-automated procedure in which a new DECODE-Plex training is triggered when a new microscope calibration is available.

4.4 Evaluation

We validated the performance of DECODE-Plex on simulated data using metrics established by the *SMLM-Challenge* [10] and extended these metrics to multi-color. We compared the performance to GlobLoc [6], a state-of-the-art maximum likelihood (MLE) multi-channel fitter.

The *Detection Accuracy* describes how many true emitters are found and how many false localizations are reported and is measured here as the Jaccard Index (JI), a relation between true positives (TP), false positives (FP), and false negatives (FN): $JI = TP/(TP + FN + FP)$. True positives (TP) are determined by an assignment algorithm, typically a *cost-minimizing assignment algorithm* like the *Hungarian algorithm* [25]. Following the conventions established by the *SMLM-Challenge*, we use a lateral threshold of 250 nm and an axial threshold of 500 nm.

To evaluate on simulated datasets, we follow the principles outlined in the SMLM challenge [10]. Its two main components are detection performance and localization precision. The detection performance is typically measured by the Jaccard index given by

$$JI = TP/(FN + FP + TP) \tag{7}$$

Where TP, FP, FN are the true positive, false positive, and false negatives, respectively. These sets of localizations are assigned by a matching algorithm closely resembling the Hungarian assignment algorithm [25], again following the conventions of the SMLM challenge. Localizations can be matched to a pair if their lateral distance is less than 250 nm and their axial distance is less than 500 nm. The localization precision can be quantified for true-positive and matched positive pairs and is typically reported as the r.m.s.e. laterally, axially, or volumetric:

$$\text{r.m.s.e.}_d = \left(\frac{1}{\text{TP}} \sum_{i=1}^{\text{TP}} \sum_{k=1}^{d} \left(x_{i,k} - x_{i,k}^{\text{GT}} \right)^2 / d \right)^{1/2} \tag{8}$$

where d is the dimensionality, x_k and x^{GT} are the predicted and ground truth coordinates along their respective axes, respectively.

For ranking and ease of comparison, the SMLM Challenge introduced the single-valued metric *efficiency* E, which is defined as

$$E = 1 - \sqrt{(1 - \text{JI})^2 + \alpha^2 d \cdot \text{r.m.s.e.}_d^2} \tag{9}$$

following ref.[10]; where α is a scaling factor set to $\alpha = 1 \cdot 10^{-2}$ nm^{-1} and $\alpha = 0.5 \cdot 10^{-2}$ nm^{-1} for 3D and 2D data respectively. Efficiency balances the detection and localization performance.

4.4.1 Multi-Color Metrics

In addition to the metrics outlined above, the assignment of emitters to the correct color is vital to multi-color SMLM. Here, we simply report the Accuracy *Acc*, which

describes the fraction of correct color assignments, and the Rejection rate Rej, which describes the fraction of unassigned emitters.

$$Acc = \frac{TP_{color}}{TP} \tag{10}$$

where TP_{color} are the true positives with correct color assignment and TP are the true positives.

$$Rej = \frac{TP_{rej}}{TP} \tag{11}$$

where TP_{rej} are the true positives rejected due to fuzzy color assignment, and TP are the true positives.

4.4.2 Validation datasets

Multicolor data

We simulated the multi-color dataset of Figure 5 with photon flux 5000/5000, background count of 20/135, and lifetime averages of 3.0 and 1.0 for both colors, respectively. We simulated a minimum of 10000 emitters for each density step.

Biplane data

We simulated the biplane dataset of Figure 6 with a photon flux of 5000, background 0, and a lifetime average of 1.0. We simulated a minimum of 50000 emitters for each density step.

4.5 Sample preparation

Seeding of U-2 OS cells on glass coverslips

Glass coverslips were treated overnight with a solution of 18.5 % (v/v) Hydrochloric acid in Methanol with constant stirring, then washed thoroughly with deionized water and dried. Wild-type U-2 OS cells were appropriately seeded in the treated coverslips to reach between 30-50% confluence after 1-2 days of incubation at 37°C and 5% CO2 in DMEM supplemented with MEM NEAA (1x), GlutaMax (1x), FBS (10%) and ZellShield (1x).

Dual-color data

U-2 OS cells on glass coverslips were quickly rinsed with pre-warmed PEM buffer (80mM PIPES, 2mM MgCl2, 5mM EGTA, pH 6.8) and fixed with a solution of 4% paraformaldehyde, 0.1% glutaraldehyde, and 4% sucrose in PEM buffer for 10 min at 37°C. Then cells were rinsed in phosphate buffer (PB) (0.1 M sodium phosphate, pH 7.3) before quenched for 7 min at room temperature with a solution of 10mg/mL sodium borohydride in PB buffer. Next, cells were permeabilized and blocked for 3 hours with immunocytochemistry buffer (ICC buffer: 0.2% bovine serum albumin, 0.1% Triton X-100 in PB buffer), followed by 30 min incubation in a drop of Image-iT FX Signal Enhancer. Permeabilized cells were incubated overnight at 4°C with

solution of antibodies (anti α-Tubulin 1:200, anti β-Tubulin 1:200, anti- Nogo-B 1:500) in ICC buffer, followed by three washes, 5 min each, in ICC buffer, and incubation with secondary antibodies (AF647 anti-sheep IgG 1:500, CF660C anti-mouse IgG 1:500) in ICC buffer, and finally washed with ICC buffer, three times, 5 min each, before mounting the sample for image acquisition in blinking buffer (10 mM NaCl, 35 mM cysteamine, 10% glucose, 500 g/L catalase from bovine liver, 40 g/L glucose oxidase, in 50 mM Tris/HCl pH 8.0).

Live-cell biplane data

The living cells in coverslips were incubated for 1 hour in the same medium, but with HMSiR-Tubulin [19] added to a final concentration of 2 uM. Then, cells were briefly washed two times with fresh medium to reduce the background of free dye, and a new medium was finally added to the sample in the mounted chamber for imaging.

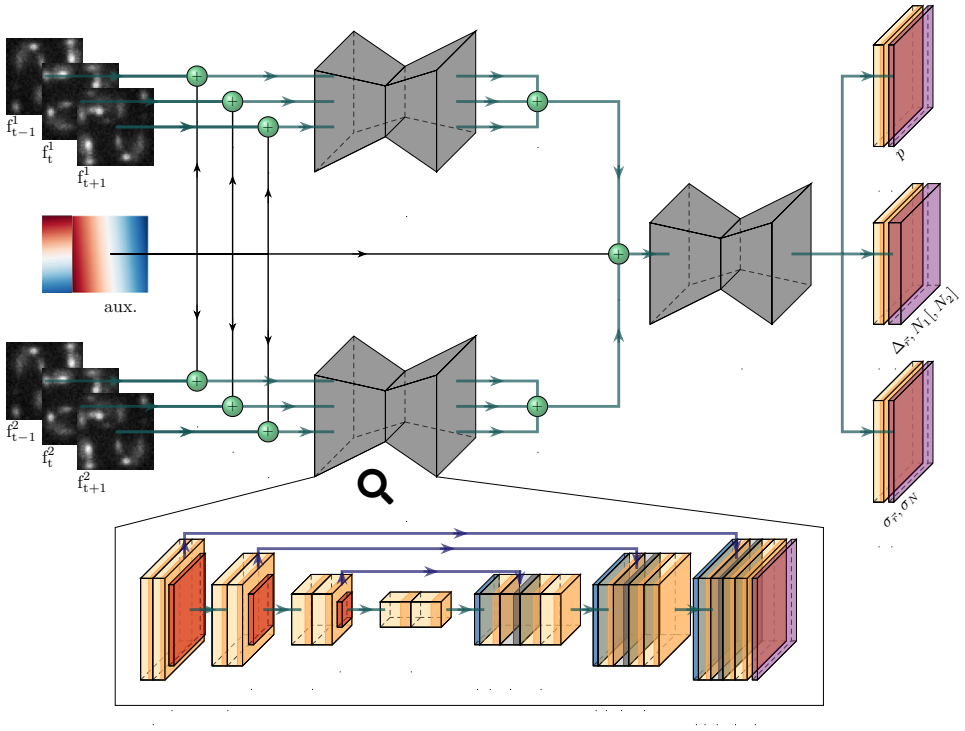


Fig. 4: **Neural network Architecture** DECODE-Plex's neural network backbone consists of three stages: One *per-channel* stage, one *union* stage applied on the concatenated feature maps, and one *output* stage producing the predictions. The *per-channel* stage is composed of a U-Net architecture (expanded at the bottom) and is applied to the prediction and context frames separately. Each channel has its own U-Net: In the figure, we show the case for a dual-channel scenario. The *union* stage is a U-Net architecture applied to the concatenation of the resulting feature maps for the prediction and context frames of all channels. The *output* stage is divided into different prediction heads, each composed of two convolutional layers. Three output heads produce probability, localization, and uncertainty estimates. An optional fourth output head can produce a background prediction. All U-Nets have three up- and downsampling stages and 64 filters in the first stage, with each stage consisting of three fully convolutional layers with 3×3 filters. The resolution is halved and the number of filters is doubled in each downsampling stage, and vice versa in the upsampling stage. In the sketch, blue arrows show skip connections. To inform the network about the channel transformation, auxiliary channels are concatenated to the input at both the *per-channel* and *union* stages.

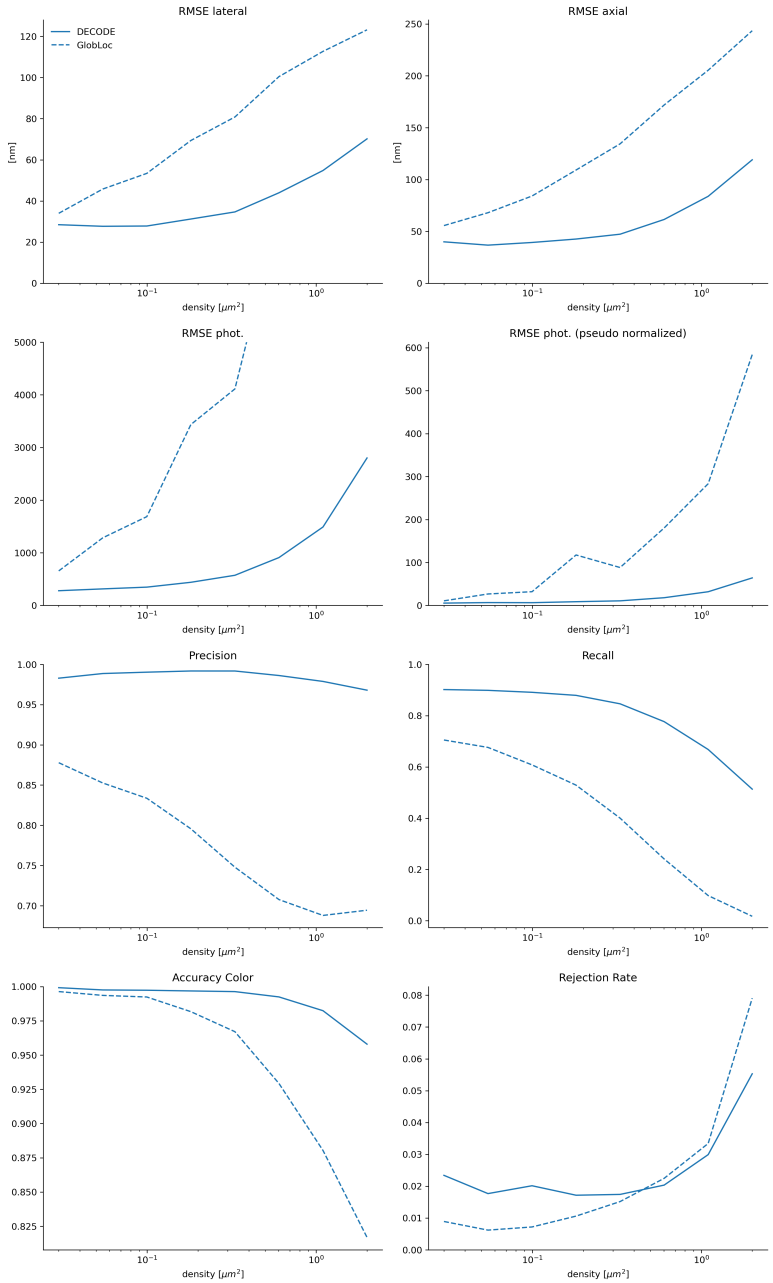


Fig. 5: **Comparison of performance metrics for dual-color data across densities** DECODE-Plex outperforms globLoc across all densities. The localizations of both algorithms were filtered to their common usage: DECODE-Plex's predictions were filtered for a probability threshold $p < 0.5$, globLoc's localizations were filtered for a relative log-likehood $LL_{rel} > -1$ and a photon count of $phot \geq 100$.

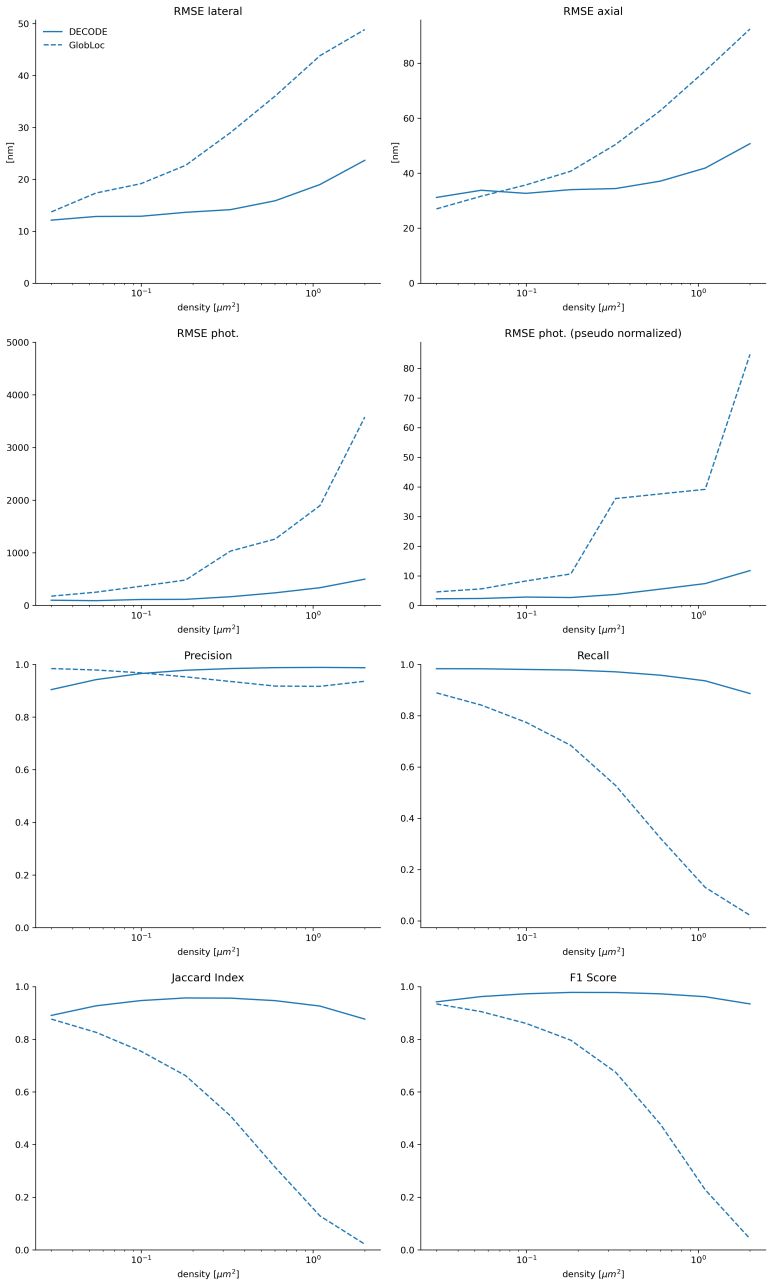


Fig. 6: **Comparison of performance metrics for biplane data across densities** DECODE-Plex outperforms globLoc across all densities. The localizations of both algorithms were filtered to their common usage: DECODE-Plex's predictions were filtered for a probability threshold $p < 0.5$, globLoc's localizations were filtered for a relative log-likehood $LL_{rel} > -1$ and a photon count of $phot \geq 100$.

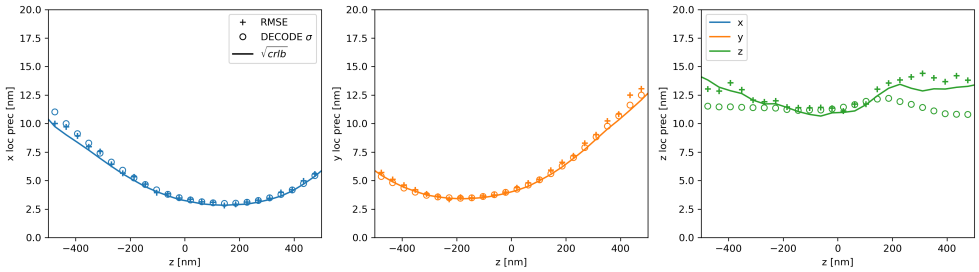


Fig. 7: **Localization error and Cramer-Rao Lower-Bound (CRLB) for dual-channel fitting**. The empirical r.m.s.e. achieved by DECODE-Plex closely match the CRLB across the z-range. DECODE-Plex uncertainty estimates closely resemble the CRLB in both lateral dimensions, in the axial dimension, the uncertainty values are slightly underreported. An experimental PSF and transformation were used for simulation.

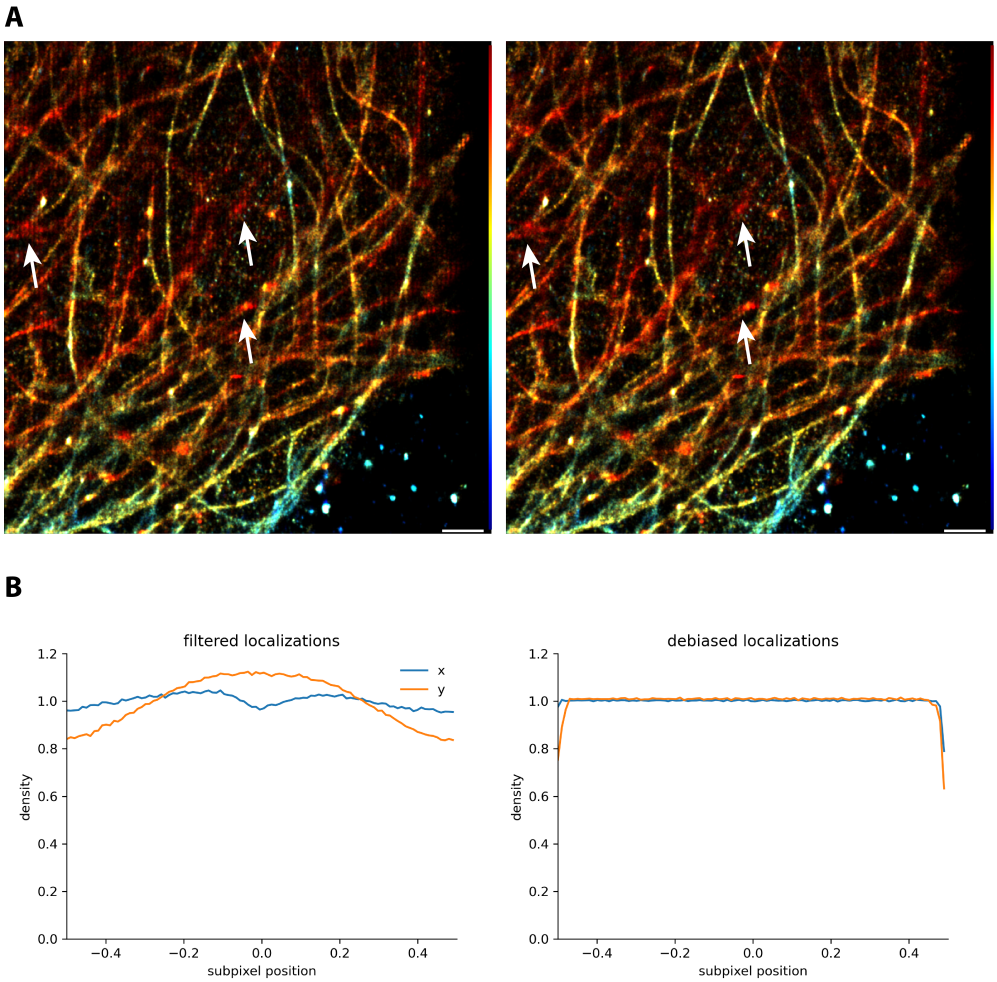


Fig. 8: **Pixelation artifacts on experimental data. (a) Microtubules and the Endopslamatic Reticulum** labeled with CF660 and AF647, respectively. Dense, out-of-focus localizations show a gridding effect due to a bias towards the pixel center. Color-coded by the z position. **(b) Filtered and de-biased localizations.** Default filtering with a probability threshold of $p > 0.5$ and $locprec > 40$nm still contains biased localizations. De-biasing, which actively shifts the localizations to equalize the subpixel position histogram, reduces the pixelation artifact.

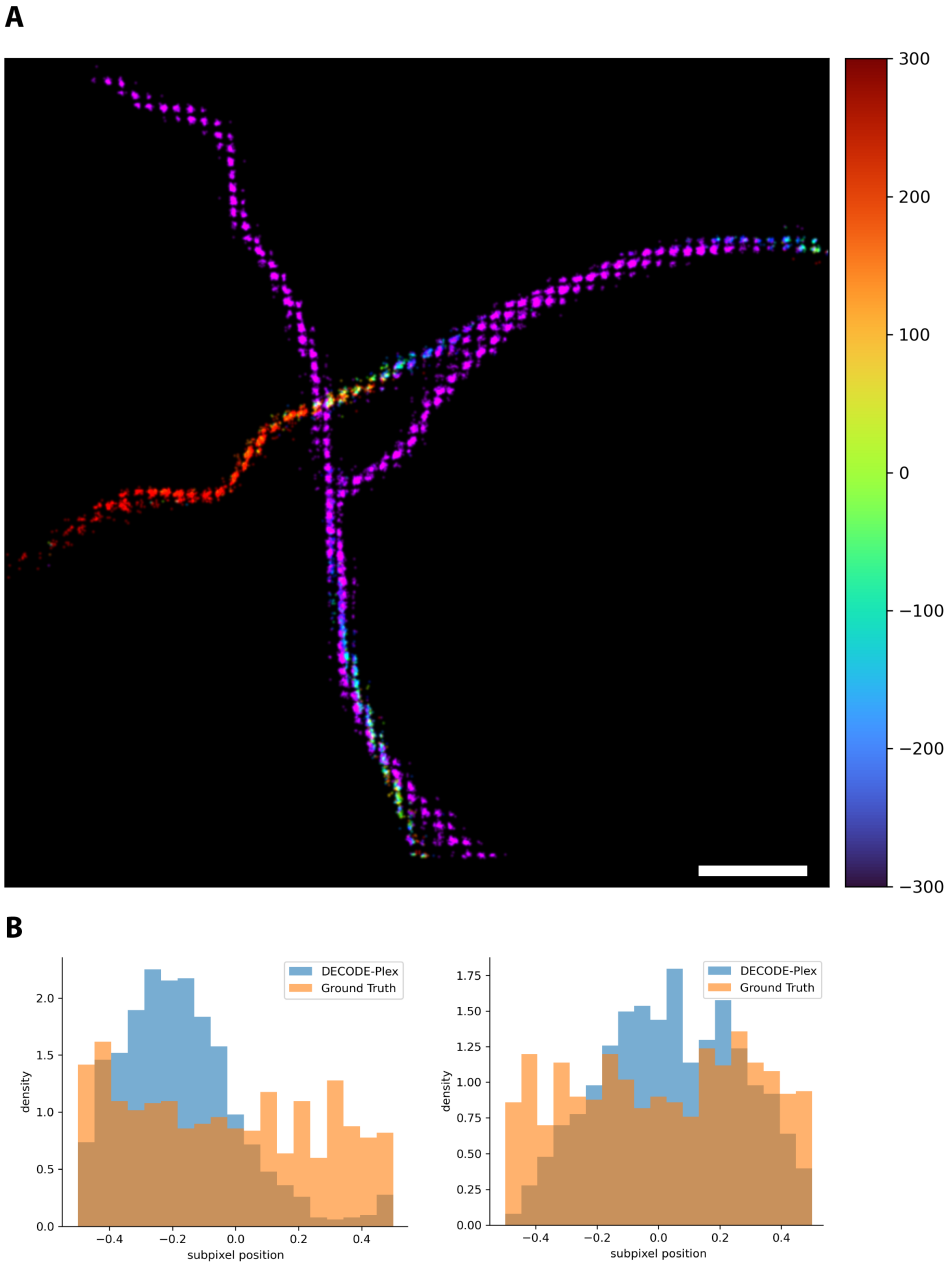


Fig. 9: **Incorrectly trained model. (a)** Localizations for the challenge data when an incorrect PSF model was used for training. DECODE-Plex outputs gridded localizations (compare [20]). **(b) subpixel positions for ground-truth emitters and matched DECODE-Plex localizations.** DECODE-Plex localizations are non-uniformly distributed across the subpixel position whereas the matched ground-truth localizations are fairly uniformly distributed. DECODE-Plex *localizations* are biased relative to the pixel center.

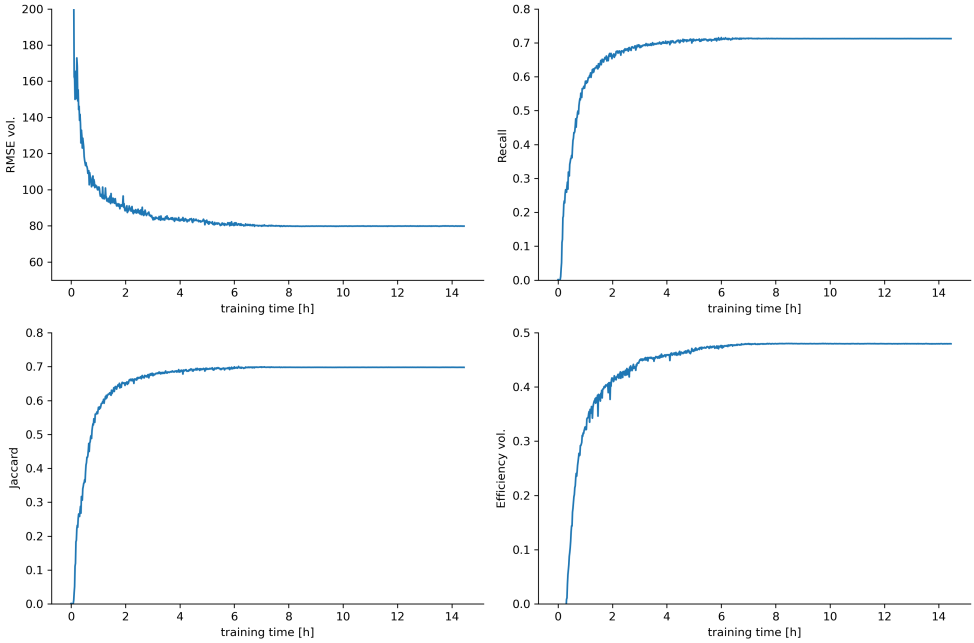


Fig. 10: **Training convergence as a function of the training time.** Convergence of DECODE-Plex performance for several metrics. Training times are measured on a single Nvidia RTX 4090 GPU. The training data covers a common multi-color scenario at high SNR.

References

[1] Betzig, E., Patterson, G.H., Sougrat, R., Lindwasser, O.W., Olenych, S., Bonifacino, J.S., Davidson, M.W., Lippincott-Schwartz, J., Hess, H.F.: Imaging Intracellular Fluorescent Proteins at Nanometer Resolution. Science **313**(5793), 1642–1645 (2006) https://doi.org/10.1126/science.1127344 . Accessed 2024-06-19

[2] Rust, M.J., Bates, M., Zhuang, X.: Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). Nature Methods **3**(10), 793–796 (2006) https://doi.org/10.1038/nmeth929 . Accessed 2024-06-19

[3] van de Linde, S., Löschberger, A., Klein, T., Heidbreder, M., Wolter, S., Heilemann, M., Sauer, M.: Direct stochastic optical reconstruction microscopy with standard fluorescent probes. Nature Protocols **6**(7), 991–1009 (2011) https://doi.org/10.1038/nprot.2011.336 . Accessed 2024-06-26

[4] Babcock, H.P., Zhuang, X.: Analyzing Single Molecule Localization Microscopy Data Using Cubic Splines. Scientific Reports **7**(1), 552 (2017) https://doi.org/10.1038/s41598-017-00622-w . Accessed 2018-12-11

[5] Li, Y., Mund, M., Hoess, P., Deschamps, J., Matti, U., Nijmeijer, B., Sabinina, V.J., Ellenberg, J., Schoen, I., Ries, J.: Real-time 3D single-molecule localization using experimental point spread functions. Nature Methods **15**(5), 367–369 (2018) https://doi.org/10.1038/nmeth.4661 . Accessed 2018-11-29

[6] Li, Y., Shi, W., Liu, S., Cavka, I., Wu, Y.-L., Matti, U., Wu, D., Koehler, S., Ries, J.: Global fitting for high-accuracy multi-channel single-molecule localization. Nature Communications **13**(1), 3133 (2022) https://doi.org/10.1038/s41467-022-30719-4 . Accessed 2024-03-04

[7] Nehme, E., Weiss, L.E., Michaeli, T., Shechtman, Y.: Deep-STORM: Super-resolution single-molecule microscopy by deep learning. Optica **5**(4), 458 (2018) https://doi.org/10.1364/OPTICA.5.000458 . Accessed 2018-11-29

[8] Nehme, E., Freedman, D., Gordon, R., Ferdman, B., Weiss, L.E., Alalouf, O., Naor, T., Orange, R., Michaeli, T., Shechtman, Y.: DeepSTORM3D: Dense 3D localization microscopy and PSF design by deep learning. Nature Methods (2020) https://doi.org/10.1038/s41592-020-0853-5

[9] Speiser, A., Müller, L.-R., Hoess, P., Matti, U., Obara, C.J., Legant, W.R., Kreshuk, A., Macke, J.H., Ries, J., Turaga, S.C.: Deep learning enables fast and dense single-molecule localization with high accuracy. Nature Methods **18**(9), 1082–1090 (2021) https://doi.org/10.1038/s41592-021-01236-x

[10] Sage, D., Pham, T.-A., Babcock, H., Lukes, T., Pengo, T., Chao, J., Velmurugan, R., Herbert, A., Agrawal, A., Colabrese, S., Wheeler, A., Archetti, A., Rieger, B., Ober, R., Hagen, G.M., Sibarita, J.-B., Ries, J., Henriques, R., Unser, M.,

Holden, S.: Super-resolution fight club: Assessment of 2D and 3D single-molecule localization microscopy software. Nature Methods **16**(5), 387–395 (2019) https://doi.org/10.1038/s41592-019-0364-4 . Accessed 2019-08-23

[11] Dempsey, G.T., Vaughan, J.C., Chen, K.H., Bates, M., Zhuang, X.: Evaluation of fluorophores for optimal performance in localization-based super-resolution imaging. Nature Methods **8**(12), 1027–1036 (2011) https://doi.org/10.1038/nmeth.1768 . Accessed 2024-06-18

[12] Bossi, M., Fölling, J., Belov, V.N., Boyarskiy, V.P., Medda, R., Egner, A., Eggeling, C., Schönle, A., Hell, S.W.: Multicolor Far-Field Fluorescence Nanoscopy through Isolated Detection of Distinct Molecular Species. Nano Letters **8**(8), 2463–2468 (2008) https://doi.org/10.1021/nl801471d . Accessed 2024-06-18

[13] Lehmann, M., Lichtner, G., Klenz, H., Schmoranzer, J.: Novel organic dyes for multicolor localization-based super-resolution microscopy. Journal of Biophotonics **9**(1-2), 161–170 (2016) https://doi.org/10.1002/jbio.201500119 . Accessed 2024-06-18

[14] Zhang, Y., Schroeder, L.K., Lessard, M.D., Kidd, P., Chung, J., Song, Y., Benedetti, L., Li, Y., Ries, J., Grimm, J.B., Lavis, L.D., De Camilli, P., Rothman, J.E., Baddeley, D., Bewersdorf, J.: Nanoscale subcellular architecture revealed by multicolor three-dimensional salvaged fluorescence imaging. Nature Methods **17**(2), 225–231 (2020) https://doi.org/10.1038/s41592-019-0676-4 . Accessed 2024-06-18

[15] Huang, B., Wang, W., Bates, M., Zhuang, X.: Three-Dimensional Super-Resolution Imaging by Stochastic Optical Reconstruction Microscopy. Science **319**(5864), 810–813 (2008) https://doi.org/10.1126/science.1153529 . Accessed 2024-06-18

[16] Pavani, S.R.P., Thompson, M.A., Biteen, J.S., Lord, S.J., Liu, N., Twieg, R.J., Piestun, R., Moerner, W.E.: Three-dimensional, single-molecule fluorescence imaging beyond the diffraction limit by using a double-helix point spread function. Proceedings of the National Academy of Sciences **106**(9), 2995–2999 (2009) https://doi.org/10.1073/pnas.0900245106 . Accessed 2024-06-18

[17] Shechtman, Y., Sahl, S.J., Backer, A.S., Moerner, W.E.: Optimal Point Spread Function Design for 3D Imaging. Physical Review Letters **113**(13), 133902 (2014) https://doi.org/10.1103/PhysRevLett.113.133902 . Accessed 2024-06-18

[18] Juette, M.F., Gould, T.J., Lessard, M.D., Mlodzianoski, M.J., Nagpure, B.S., Bennett, B.T., Hess, S.T., Bewersdorf, J.: Three-dimensional sub–100 nm resolution fluorescence microscopy of thick samples. Nature Methods **5**(6), 527–529 (2008) https://doi.org/10.1038/nmeth.1211 . Accessed 2024-06-18

[19] Gerasimaitė, R., Bucevičius, J., Kiszka, K.A., Schnorrenberg, S., Kostiuk, G., Koenen, T., Lukinavičius, G.: Blinking Fluorescent Probes for Tubulin Nanoscopy in Living and Fixed Cells. ACS Chemical Biology **16**(11), 2130–2136 (2021) https://doi.org/10.1021/acschembio.1c00538 . Accessed 2024-07-22

[20] Lelek, M., Gyparaki, M.T., Beliu, G., Schueder, F., Griffié, J., Manley, S., Jungmann, R., Sauer, M., Lakadamyali, M., Zimmer, C.: Single-molecule localization microscopy. Nature Reviews Methods Primers **1**(1), 1–27 (2021) https://doi.org/10.1038/s43586-021-00038-x . Accessed 2024-06-18

[21] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pp. 234–241. Springer, Cham (2015)

[22] Clevert, D.-A., Unterthiner, T., Hochreiter, S.: Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). arXiv (2016). https://doi.org/10.48550/arXiv.1511.07289 . http://arxiv.org/abs/1511.07289 Accessed 2024-06-26

[23] Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization. arXiv (2019). https://doi.org/10.48550/arXiv.1711.05101 . http://arxiv.org/abs/1711.05101 Accessed 2024-06-26

[24] Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. arXiv (2017). http://arxiv.org/abs/1701.04128 Accessed 2024-04-03

[25] Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1-2), 83–97 (1955) https://doi.org/10.1002/nav.3800020109

A.2 DECODE-OpenCloud: Sharing Computational Resources for Microscopy Data Analysis

DECODE-OpenCloud: Sharing Computational Resources for Microscopy Data Analysis

Lucas-Raphael Müller[1,2†], Arthur Jaques[1,2†], Jonas Ries[1,2,3,4,5*], Jakob Macke[1,6*]

[1]Machine Learning in Science, Excellencecluster Machine Learning, Tübingen, Germany.
[2]Cell and Biophysics, European Molecular Biology Laboratory, Heidelberg, Germany.
[3]Max Perutz Labs, Vienna Biocenter Campus, Vienna, Austria.
[4]Department of Structural and Computational Biology, University of Vienna, Vienna, Austria.
[5]Faculty of Physics, University of Vienna, Vienna, Austria.
[6]Tübingen AI Center, University of Tübingen, Tübingen, Germany.

*Corresponding author(s). E-mail(s): jonas.ries@maxperutzlabs.ac.at; jakob.macke@uni-tuebingen.de;
Contributing authors: lucas.mueller@embl.de;
†These authors contributed equally to this work.

Abstract

Fast-paced advancements in machine learning algorithms enable increasingly complex workflows in microscopy but pose significant computational demands and require access to performant hardware. Computational tools that efficiently process repetitive or large-scale workflows have the potential to accelerate scientific discovery. Here, we present DECODE-OpenCloud, a cloud-backed solution that enables researchers to easily run compute-intensive algorithms for microscopy without caring where the job actually runs. DECODE Open-Cloud distributes the jobs dynamically to unused local computational resources or cloud-based services, allowing for high availability. With DECODE-OpenCloud's release, we include three algorithms for super-resolution microscopy: DECODE and DECODE-Plex, which are deep-learning-based high-density localization frameworks for single-molecule localization microscopy (SMLM) for fast 3D super-resolution, and COMET, a performant drift-correction algorithm. We

designed DECODE-OpenCloud to be flexible and easily expanded to other algorithms for microscopy.

Keywords: Machine Learning, Cloud, Open Source, GPU

1 Introduction

With recent developments in machine learning, an increasing number of algorithms pose significant requirements for computational hardware, maintenance, and organization, as they heavily use accelerated computing and graphical processing units (GPUs). Local hardware setups are frequently overspecified for a large portion of the time and underspecified at rare peak loads. Moreover, running and maintaining such hardware and software poses a burden for researchers with little time or computing budget who want to use the software while not caring about any hardware constraints.

In addition to hardware constraints, developing and running algorithms reliably on different computational setups frequently remains challenging, as it requires the algorithm developer to keep track of all hardware constraints and software libraries, including versions used in development. The algorithm user, on the other hand, needs to make sure to satisfy all requirements for the particular algorithm. More and more efforts are being made to make scientific software reproducible and maintainable. However, a significant burden for scientific efforts to be translatable to productized use is open source code availability, reproducibility, and ease of productionized use. Recent efforts have been made to either centralize the usage of many algorithms in one library or provide basic computing infrastructure. Examples of this are ZeroCostDL4Mic [1], which provides a set of algorithms for Image-segmentation, Object Detection, and Image denoising and restoration in one place in the form of Google Colab Notebooks where Google provides free computing power. More broadly, frameworks like the sbi-framework [2], and the Huggingface libraries [3] are all efforts in the field to consolidate the implementation of several algorithms into a single library, aiding developers for simple usage without complicated compilation or installation steps.

However, integrating algorithms as code is only part of making recent algorithms available for frequent, productionized usage. For example, the usage of Google Colab instances is often insufficient as it lacks control over details of the computer environment, which are frequently updated and lead to breakage of once-working algorithms. Moreover, there is a limited possibility of interacting with Google Colab as a proper software service via APIs.

With DECODE-OpenCloud, we aim to fill this gap by showcasing how such infrastructure can be implemented and providing ready-made implementations of three algorithms for localization microscopy (i.e., localization algorithms DECODE [4], its multi-channel extension DECODE-Plex, and COMET drift correction[**?**]). Finally, we release the source code of our infrastructure implementation.

The high-level idea and detailed implementation of DECODE-OpenCloud is depicted in Figure 1. Any researcher can interact with our cloud service using the frontend website, where they can upload and download load data, specify configuration,

and submit compute jobs. DECODE-Cloud will then distribute the computation jobs to the attached computing power or, depending on the configuration, pass on stale computing jobs to cloud-backed runners. We note that we designed DECODE-Cloud in this hybrid way to save on cloud computation costs and aim to distribute computational load among locally unused hardware. We specifically invite researchers with idle hardware to attach their hardware in a give-and-take fashion.

2 Methods

Figure 1b outlines the core components and services involved in implementing DECODE-OpenCloud, where applicable we name the concrete AWS component used in our case. Figure 2 specifies a typical job's request flow, which includes the user's interaction through the front end, data handling, and job distribution; we outline the detailed flow below.

2.1 Job flow

A user interacts with the user-facing API through the front end website or programmatically via API calls. Authentication is implemented by Tokens and backed by AWS Cognito. The user then uploads configuration and data files to cloud storage backed by an AWS S3 service. All job information is stored in a database where new jobs are posted. The second part of our cloud infrastructure, the worker-facing API, is called by locally attached compute power (see give-and-take idea above) or periodically checked by an AWS lambda function for stale, unfinished jobs, should there be peak load and insufficient local compute power. The worker-facing API's backend then selects a compute job that matches the capabilities of the compute instance (local or cloud compute) and returns all necessary information for the job, i.e., which docker image is used and pre-signed URLs to necessary files. Local or cloud runners then download the necessary files and the docker image or use a cached version and start the compute job. During execution of the job, a thin wrapper around the compute job periodically sends status signals to the worker-facing API to allow for job status tracking. Upon job completion, the worker uploads the specified files and logs, and the user is notified via mail about the job's success. Note that files are downloaded and uploaded by both users and workers using pre-signed URLs to avoid the additional load that would be caused by directly sending files through the APIs.

2.2 Algorithm definition

In Decode-OpenCloud, an algorithm is defined by a small set of parameters that need to be provided for it. This makes sure that the overhead for adding additional software is low. Figure 3 outlines a concrete definition of one of the algorithms made readily available with DECODE-OpenCloud. In essence, the integration of an algorithm means specifying its entry points in the docker container, the run command, necessary files to download, and result files, which should be uploaded after the completion of the job. Moreover, default minimum hardware specs are defined, which can be overridden by specifying different values during job submission.

2.3 Algorithm interaction

While we provide a basic frontend website for a researcher's interaction with DECODE-OpenCloud, we note that the core implementation is simply a REST-API. We invite algorithm developers who want to integrate their software with DECODE-OpenCloud and already have a user interface to use the user-facing API for an optimal user experience.

2.4 Job failure

Should exceptions occur during the computation, the worker will send the logs of its failures to the worker-facing API to aid a debugging procedure. A silent job failure (e.g. due to a local internet outage) is captured by the absence of status signals, in which case another compute instance can take over the job's computation.

2.5 Core Components and Cloud Provider

Implementing DECODE-OpenCloud necessitates a small set of infrastructure elements provisioned by the cloud platform or a local environment. We chose Amazon Web Services (AWS) for the current cloud backing of DECODE-OpenCloud but note that other cloud providers or local infrastructure could equally provide the necessary compute components. DECODE-OpenCloud's core components are:

1. *Compute Instances* — Host both the user-facing and worker-facing API (AWS AppRunner/EC2).
2. *File Storage* — Handles file storage needs (AWS S3).
3. *Relational Database System* — Tracks jobs (AWS Relational Database System).
4. *Docker Image Hub* — Stores docker images (AWS Elastic Container Registry).
5. *Identity Provider* — Used for authentication (AWS Cognito).
6. *Serverless Helper Functions* — Check the worker-facing API for stale jobs (AWS Lambda).

where the respective AWS services are noted in brackets. We set up the necessary *infrastructure as code*, which is released along with DECODE-OpenCloud.

3 Results

3.1 DECODE-OpenCloud framework

With the release of the DECODE-OpenCloud framework, we showcase its use and extensibility for the community with three algorithms, DECODE [4], its multi-channel extension DECODE-Plex, and COMET[**?**] for localization microscopy. We focused on developing the framework and its components so that they can be easily extended to other algorithms. Algorithm developers simply provide a basic Dockerfile that captures all package requirements. Typically, this will be a five-line Dockerfile based on a Python or conda image, for which we provide a template and reference implementation. Notably, algorithm developers do not need to handle file uploads and downloads, status pinging, or queue logic.

Our framework orchestrates computational load, i.e., compute jobs, with relatively low time and resource overhead. Jobs start in about 10-30 seconds when free computing resources become available. The worker pulls the docker image corresponding to the job once, automatically caches it, downloads the job-specific files, runs the actual compute job, and uploads the outcoming files. If new software is added to DECODE-OpenCloud, workers automatically pull corresponding docker images to avoid unnecessary individual maintenance.

3.2 Reference Implementations

DECODE [4] and its multi-channel extension DECODE-Plex are two instances of simulation-based inference algorithms for localization microscopy that need frequent retraining due to the nature of the application. Commonly, with temporally varying PSF and varying experiments, the training of a new model is indicated. In extreme cases, this corresponds to a 1:1 ratio of experimental acquisitions and trained DECODE (-Plex) models, which induces a high load on computational resources and manual orchestration. DECODE-OpenCloud helps in that it allows for automatic training pipelines and, subsequently, the distribution of the trained models.

COMET[5] is a GPU-accelerated drift-correction algorithm for SMLM. Its input is simply the set of localizations and a few configuration parameters, which we included in DECODE-OpenCloud without requiring further software changes or assistance.

4 Discussion

Translating scientific software to its productive usage is vital in advancing scientific knowledge to benefit the public and the scientific community. However, many algorithms lack easy installation and induce high maintenance efforts or computation costs even if open-sourced. This work aims to fill the gap by outlining and provisioning a framework and its concrete implementation and public accessibility.

With the initial release of DECODE-OpenCloud, we showcase its usage with three algorithms for localization microscopy and invite other algorithm developers to integrate with our framework, and any individual with powerful yet partially unused hardware attach these resources to our framework.

DECODE-OpenCloud has three significant advantages over state-of-the-art approaches.

1. *Ease of use*—Our framework allows any researcher to interact easily with DECODE-OpenCloud either with the provided website or with directly integrated software. In particular, researchers who do not have access to the required hardware (e.g., GPUs) can utilize computing resources shared by institutions that do not need their total capacity at that time. This lowers the entry barrier for using compute-intensive algorithms to make research more accessible.
2. *Ease of integration*—Algorithm developers only need to wrap their algorithm in a docker container and provide a small set of parameters.
3. *Resource usage*—The effective usage of computing resources is mandatory both due to economic and environmental considerations. DECODE-OpenCloud satisfies

both concerns as it lowers the necessity for overly scaled hardware requirements in peak load times and makes more efficient use of computation capability when hardware is unused at its physical location. We avoid tight integration directly into local infrastructure as this would be prohibited by data security and maintainability considerations.

4.1 Limitations

The main limitation of DECODE-OpenCloud is transferring data between the user and the compute instance. This necessity comes at a different cost for different algorithms. While DECODE, DECODE-Plex, and COMET come readily implemented with the release of DECODE-OpenCloud and do not need large amounts of data for training or processing, other algorithms might need vast amounts of data (in the order of terabytes) for successful training. In the future, we aim to investigate an option to abstain from uploading data to the cloud and directly mount local paths. This may, however, limit our work's applicability and scale. If DECODE-OpenCloud is operated with cloud fallback computation power, the provider of DECODE-OpenCloud incurs AWS charges.

4.2 Future Work

We will investigate into bookkeeping consumed and provided computation time to ensure a fair distribution of computational resources among users. We hope to have outlined that integrating other algorithms into DECODE-OpenCloud does not require substantial changes to the algorithms, and hope to integrate more algorithms in the future.

4.3 Conclusion

In conclusion, we (1) release DECODE-OpenCloud to speed up the process of scientific software to productionized usage with low additional burden and easy maintainability. (2) As a first step, we integrated three algorithms for localization microscopy to invite researchers to use our framework. (3) Lastly, we hope that more algorithms will integrate with our framework or see similar approaches make use of the publicly available source code of the implementation.

References

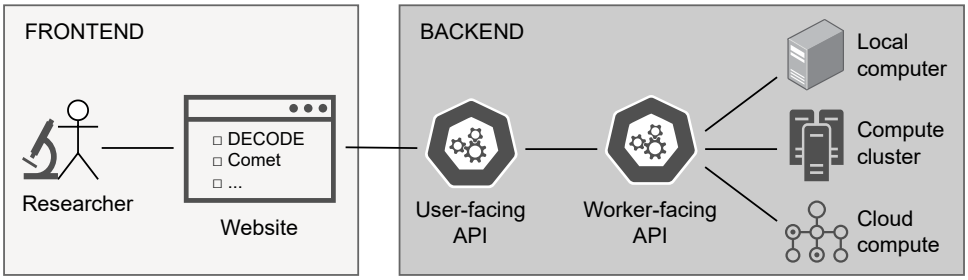
[1] Chamier, L., Laine, R.F., Jukkala, J., Spahn, C., Krentzel, D., Nehme, E., Lerche, M., Hernández-Pérez, S., Mattila, P.K., Karinou, E., *et al.*: Democratising deep learning for microscopy with zerocostdl4mic. Nature communications **12**(1), 2276 (2021)

[2] Tejero-Cantero, A., Boelts, J., Deistler, M., Lueckmann, J.-M., Durkan, C., Gonçalves, P.J., Greenberg, D.S., Macke, J.H.: sbi: A toolkit for simulation-based inference. Journal of Open Source Software **5**(52), 2505 (2020) https://doi.org/10.21105/joss.02505

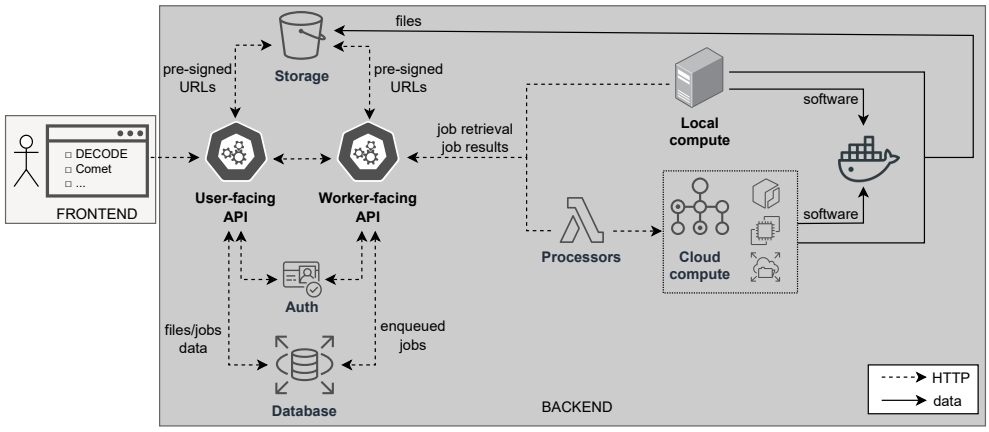
[3] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: HuggingFace's Transformers: State-of-the-art Natural Language Processing (2020)

[4] Speiser, A., Müller, L.-R., Hoess, P., Matti, U., Obara, C.J., Legant, W.R., Kreshuk, A., Macke, J.H., Ries, J., Turaga, S.C.: Deep learning enables fast and dense single-molecule localization with high accuracy. Nature methods **18**(9), 1082–1090 (2021)

[5] Reinkensmeier, L., Bates, M.: Gpufit/Comet (2023). https://github.com/gpufit/Comet Accessed 2024-07-13



(a) High-level overview.



(b) Detailed implementation.

Fig. 1: **DECODE-OpenCloud Architecture. (a) High-level overview.** Researchers choose among the implemented software applications in the front-end website, prepare data and configuration, and submit a compute job. They do not need to care about anything in the *backend*, which automatically handles data flows and the distribution of the computational load. Depending on the computational load and priority, the *backend* will assign the job to *local* or *cloud* computing workers. **(b) Detailed implementation.** Users interact only with a "user-facing" endpoint that accepts new jobs for training, inference, and monitoring. A separate "worker-facing API" connects computing resources with the jobs that researchers have submitted. Both local computes as well as cloud instances query the "worker-facing" API for new jobs. The cloud backend handles user authentication, management, orchestration, and file storage.

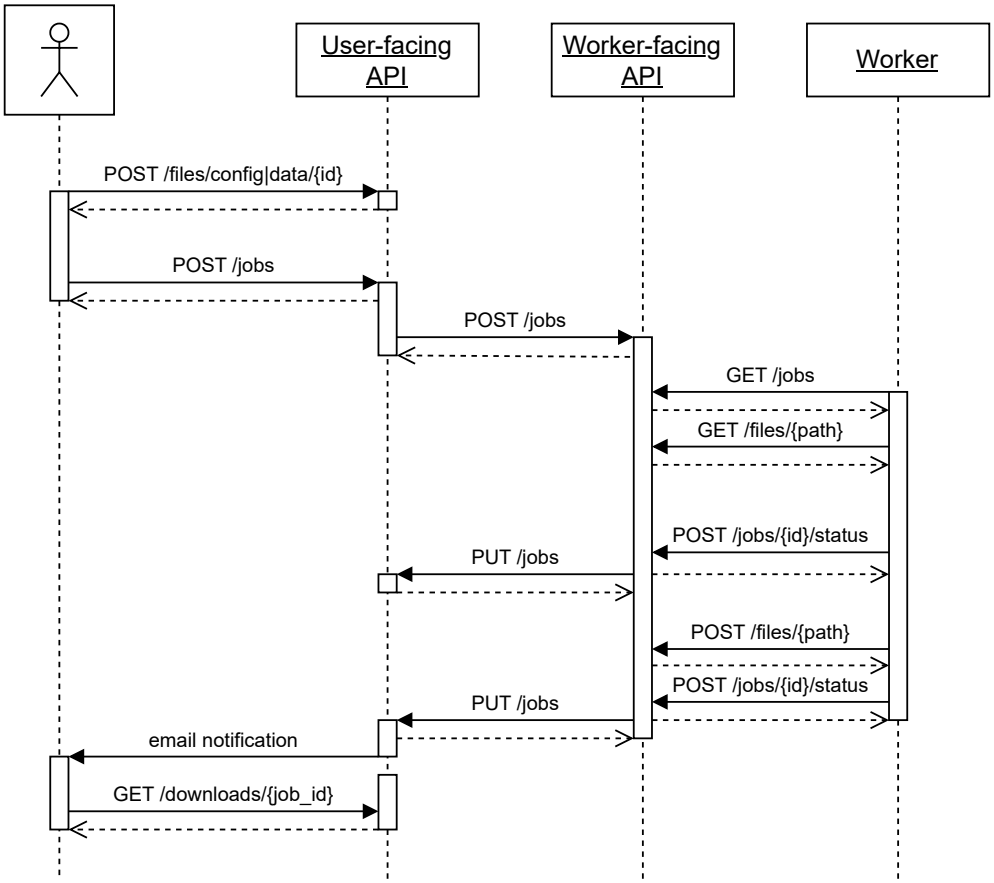


Fig. 2: **Backend implementation HTTP flow for new jobs.** A user specifies a job and POST files and their respective configurations for the user-facing API. The user-facing API POSTs the job to the worker-facing API, which handles the orchestration of computational resources. Following a pull schema, workers ask for unfinished jobs via GET and all necessary files. While the job is running, workers POST status updates to the worker-facing API which in turn updates the status for the user-facing API. Users can download the outputs and are notified upon success or failure.

```yaml
# complete definition of a new algorithm with a hand-ful parameters
    comprising the entrypoint, files and necessary hardware
decode:
  v0_10_1:  # version
    train:  # endpoint
      app:
        cmd:  # command-line arguments
          - "/docker/entrypoint.sh"
          - "--train"
          - "--calib_path=$(find /files/data -name '*.mat' | head -n
              1)"
          - "--param_path=$(find /files/config -name '*.yaml' | head
              -n 1)"
          - "--model_path=/files/model"
          - "--log_path=/files/log"
        env: []  # environment variables
      handler:
        image_url: "public.ecr.aws/g0e9g3b1/decode:v0_10_1"  # docker
            image
        files_down:  # file to download prior to execution
          config_id: config
          data_ids: data
          artifact_ids: artifact
        files_up:  # files to upload post execution
          log: log
          artifact: model
      aws_resources:  # default resources for running
        hardware:
          MEMORY: 8000
          VCPU: 4
          GPU: 1
        timeout: 18000
```

Fig. 3: **Specification for cloud-backed algorithms.** Algorithms are defined with a small set of parameters that the authors must provide to be included in DECODE-OpenCloud. These include the entrypoint of their docker container, the public docker image URL, minimum hardware requirements as well as relevant paths of input and output files.