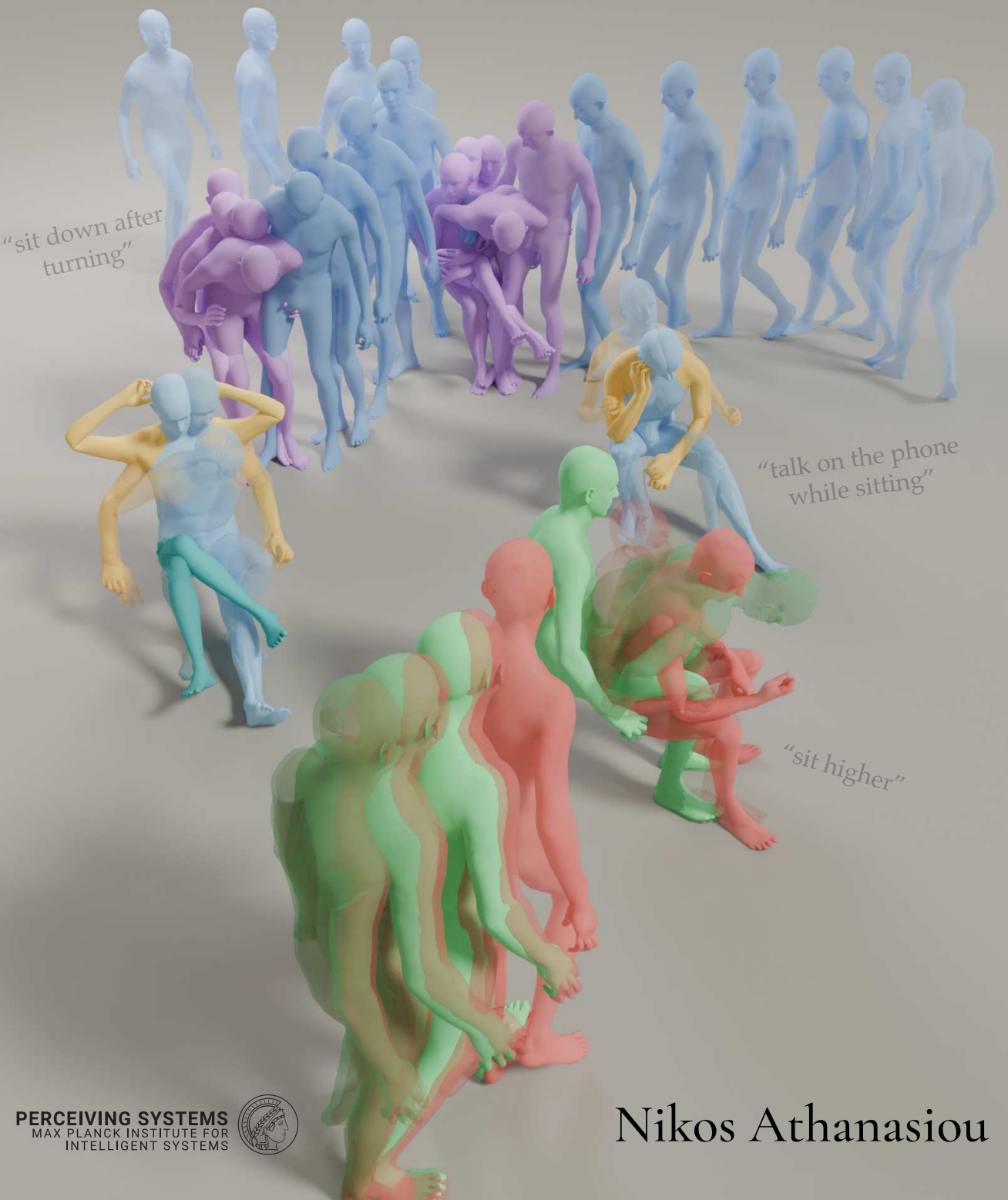
Towards *Fine-grained* 3D Human Motion Generation from Textual Instructions



PERCEIVING SYSTEMS
MAX PLANCK INSTITUTE FOR INTELLIGENT SYSTEMS

Nikos Athanasiou

Towards Fine-grained 3D Human Motion Generation from Textual Instructions

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

vorgelegt von

Nikolaos Athanasiou

aus Mytilini/Griechenland

Tübingen
2025

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

Tag der mündlichen Qualifikation:	15.01.2026
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Michael J. Black
2. Berichterstatter:	Prof. Dr. Gerard Pons-Moll

To Marianthi and Strato, who, under difficult circumstances, showed that love is a great value, and how kindness can be leveraged for our own good. To the unique and boisterous personality.

"Experience is what you get when you didn't get what you wanted.

When you see yourself doing something badly and no one bothering to tell you anymore, that's a very bad place to be. Your critics are the ones telling you they still love you and care.

If you're not listening for your mistakes then your not at the right place.

When you do the right thing good stuff has a way of happening. "

Last Lecture, Randy Pausch

Abstract

3D human motion generation and editing are critical for various applications such as animation, virtual/augmented reality, and human-computer interaction. These tasks have significant implications for unlocking interactive experiences with embodied agents and automating creative processes in virtual/augmented realities and the film and animation industry. Natural language is an intuitive and human-friendly signal to communicate with a computer and give instructions. For that reason, it is a prominent control signal for generative AI models recently. Putting motion generation and text control together, and building controllable motion generation from text instructions in a finegrained manner can greatly benefit the animation and robotics industries, facilitating the creation of 3D human avatars that can be prompted to follow instructions and edit their motions. The primary goal of this research is to explore and develop methods, tools, and datasets for generating and editing 3D human motions from textual descriptions with finegrained control.

Our first contribution, TEACH, addresses the problem of composing actions temporally. TEACH generates 3D human motions that correspond to a series of natural language descriptions along with smooth transitions, maintaining the temporal order of the instructions. Leveraging the BABEL dataset, which includes sequences of actions with transitions, we design a Transformer-based approach that operates non-autoregressively within individual actions and autoregressively across action sequences. This hierarchical method generates smooth and temporally coherent motions from textual descriptions, achieving state-of-the-art results in synthesis of text-conditioned temporal motion compositions.

Our second contribution focuses on spatial hierarchies of human motion. SINC synthesizes 3D human motions based on textual inputs describing simultaneous actions, such as “waving hand” while “walking”. We observe that large language models, like GPT-3, contain structural knowledge about human motions, such as the body parts involved in actions. We use such models and extract information about body part

involvement in motions and use this to create synthetic training data, with our SINC-GPT data creation pipeline. This allows us to train SINC, a model that can handle complex spatial compositions where multiple actions occur concurrently, which was a open challenge for previous work. SINC integrates body part-specific motion data to generate coherent and realistic simultaneous actions, overcoming the limitations of traditional approaches that typically address single actions or sequential compositions. By generating synthetic data that includes these complex interactions, SINC generalizes to new, unseen combinations of actions.

Our third contribution introduces the MotionFix dataset along with the TMED model for editing existing animations. Given a 3D human motion and a textual description of the desired modification, TMED generates an edited motion as specified by the text. Given the absence of data for that task, we build a semi-automatic pipeline to first retrieve candidate pairs of motions that are suitable for editing and then describe their differences with text. This results in a dataset of triplets comprising a source motion, a target motion, and an edit text. Using the MotionFix dataset, we train a conditional diffusion model, TMED, that takes both the source motion and the edit text as input, demonstrating superior performance over models trained solely on text-motion pairs. Our approach establishes a new benchmark with retrieval-based metrics for motion editing.

Our fourth contribution is the BABEL dataset. BABEL comprises a rich resource of semantically annotated 3D motion-capture data, to address critical challenges in understanding and generating human motion. BABEL bridges the gap between large-scale video datasets with semantic action labels and motion-capture datasets that lack semantic richness. By providing frame-level and sequence-level annotations for over 43 hours of MoCap sequences, encompassing more than 250 unique action categories, BABEL enables precise alignment of actions with 3D human motion data. In this thesis, BABEL plays a crucial role in benchmarking and training our models, particularly TEACH and SINC, by offering high-quality labels that support tasks such as action recognition, temporal localization, and motion synthesis. The finegrained annotations of overlapping and sequential actions in BABEL allow us to model complex temporal and

spatial interactions, demonstrating its utility in advancing generative and analytical tasks in 3D human motion.

In summary, our contributions in this thesis are as follows: (i) we present TEACH, a hierarchical approach for temporal action composition that generates realistic human motions from sequential textual descriptions, (ii) we introduce SINC, a method for generating simultaneous actions, which is trained using synthetic data extracted with the help of language models, (iii) we develop MotionFix dataset and introduce a framework for finegrained 3D motion editing using TMED conditionla diffusion model. The work in this thesis, paves the way for advanced applications in animation, virtual reality, and interactive systems.

Zusammenfassung

Die Generierung und Bearbeitung von 3D-Menschbewegungen sind von entscheidender Bedeutung für verschiedene Anwendungen wie Animation, Gaming, virtuelle Realität und Mensch-Computer-Interaktion. Diese Aufgaben haben weitreichende Implikationen für die Schaffung interaktiver Erfahrungen mit verkörperten Agenten und die Automatisierung kreativer Prozesse in virtuellen/augmentierten Realitäten sowie in der Film- und Animationsindustrie. Natürliche Sprache ist ein intuitives und benutzerfreundliches Mittel, um mit einem Computer zu kommunizieren und Anweisungen zu geben. Aus diesem Grund hat sie sich in letzter Zeit zu einem herausragenden Steuersignal für generative KI-Modelle entwickelt. Die Kombination von Bewegungsgenerierung und Textsteuerung sowie der Aufbau einer steuerbaren Bewegungsgenerierung auf Basis von Textanweisungen in fein abgestimmter Weise können die Animations- und Robotikindustrie erheblich voranbringen, indem 3D-Menschavatare geschaffen werden, die Anweisungen befolgen und ihre Bewegungen bearbeiten können. Das Hauptziel dieser Arbeit ist es, Methoden, Werkzeuge und Datensätze zu erforschen und zu entwickeln, die die Generierung und Bearbeitung von 3D-Menschbewegungen aus textuellen Beschreibungen mit fein abgestimmter Kontrolle ermöglichen.

Unser erster Beitrag befasst sich mit dem Problem der zeitlichen Kompositionen durch TEACH, ein Modell zur zeitlichen Aktionskomposition. TEACH generiert 3D-Menschbewegungen, die einer Reihe von Sprachbeschreibungen entsprechen und dabei sanfte Übergänge sowie die zeitliche Reihenfolge der Anweisungen beibehalten. Unter Verwendung des BABEL-Datensatzes, der Sequenzen von Aktionen mit Übergängen umfasst, entwickeln wir einen Transformer-basierten Ansatz, der nicht-autoregressiv innerhalb einzelner Aktionen und autoregressiv über Aktionssequenzen hinweg arbeitet. Diese hierarchische Methode ermöglicht die Generierung glatter und zeitlich kohärenter Bewegungen aus textuellen Beschreibungen und erreicht dabei den Stand der Technik in der Synthese textkonditionierter zeitlicher Bewegungskompositionen.

Unser zweiter Beitrag konzentriert sich auf räumliche Hierarchien menschlicher Bewegungen. SINC synthetisiert 3D-Menschbewegungen basierend auf textuellen Eingaben, die gleichzeitige Aktionen beschreiben, wie zum Beispiel “Winken” während “Gehen”. Wir beobachten, dass große Sprachmodelle wie GPT-3 strukturelles Wissen über menschliche Bewegungen enthalten, etwa welche Körperteile an bestimmten Aktionen beteiligt sind. Dieses Wissen nutzen wir, um Informationen über die Beteiligung von Körperteilen zu extrahieren und damit synthetische Trainingsdaten über unsere SINC-GPT-Datenpipeline zu erzeugen. Dadurch können wir SINC trainieren, ein Modell, das komplexe räumliche Kompositionen verarbeiten kann, bei denen mehrere Aktionen gleichzeitig auftreten – ein bislang offenes Problem. SINC integriert bewegungsspezifische Daten von Körperteilen, um kohärente und realistische gleichzeitige Aktionen zu erzeugen, und überwindet damit die Einschränkungen herkömmlicher Ansätze, die typischerweise nur einzelne Aktionen oder sequentielle Kompositionen berücksichtigen. Durch die Generierung synthetischer Daten, die diese komplexen Interaktionen umfassen, generalisiert SINC auch auf neue, unbekannte Kombinationen von Aktionen.

Unser dritter Beitrag führt den MotionFix-Datensatz zusammen mit dem TMED-Modell zur Bearbeitung bestehender Animationen ein. Gegeben eine 3D-Menschbewegung und eine textuelle Beschreibung der gewünschten Modifikation generiert TMED eine bearbeitete Bewegung entsprechend der Textbeschreibung. Angesichts des Fehlens geeigneter Daten für diese Aufgabe entwickeln wir eine halbautomatische Pipeline, um zunächst geeignete Paare von Bewegungen zu identifizieren und anschließend ihre Unterschiede textuell zu beschreiben. Dies führt zur Erstellung eines Datensatzes aus Triplets, bestehend aus einer Ausgangsbewegung, einer Zielbewegung und einem Bearbeitungstext. Mit Hilfe des MotionFix-Datensatzes trainieren wir ein konditionales Diffusionsmodell, TMED, das sowohl die Ausgangsbewegung als auch den Bearbeitungstext als Eingaben nutzt und eine überlegene Leistung gegenüber Modellen zeigt, die ausschließlich auf Text-Bewegungspaaren trainiert wurden. Unser Ansatz etabliert eine neue Benchmark mit retrieval-basierten Metriken für die Bewegungsbearbeitung.

Unser vierter Beitrag ist der BABEL-Datensatz. BABEL stellt eine reichhaltige Ressource semantisch annotierter 3D-Motion-Capture-Daten dar, um zentrale Herausforderungen beim Verständnis und bei der Generierung menschlicher Bewegungen zu adressieren. BABEL überbrückt die Lücke zwischen groß angelegten Videodatensätzen mit semantischen Aktionslabels und Motion-Capture-Datensätzen, denen semantische Tiefe fehlt. Durch die Bereitstellung von Frame- und Sequenzebenen-Annotationen für über 43 Stunden MoCap-Sequenzen mit mehr als 250 einzigartigen Aktionskategorien ermöglicht BABEL die präzise Zuordnung von Aktionen zu 3D-Menschbewegungsdaten. In dieser Arbeit spielt BABELeine entscheidende Rolle beim Benchmarking und Training unserer Modelle, insbesondere TEACH und SINC, da es hochwertige Labels bietet, die Aufgaben wie Aktionsklassifikation, zeitliche Lokalisierung und Bewegungssynthese unterstützen. Die fein abgestimmten Annotationen überlappender und sequentieller Aktionen in BABELermöglichen es uns, komplexe zeitliche und räumliche Interaktionen zu modellieren und zeigen dessen Nützlichkeit für generative und analytische Aufgaben in der 3D-Menschbewegung.

Zusammenfassend sind unsere Beiträge in dieser Arbeit wie folgt: (i) wir präsentieren TEACH, einen hierarchischen Ansatz zur zeitlichen Aktionskomposition, der realistische Bewegungen aus sequentiellen Textbeschreibungen erzeugt, (ii) wir stellen SINC vor, eine Methode zur Generierung gleichzeitiger Aktionen, die mithilfe synthetischer Daten aus Sprachmodellen trainiert wird, (iii) wir entwickeln den MotionFix-Datensatz und führen ein Framework zur fein abgestimmten Bearbeitung von 3D-Menschbewegungen unter Verwendung des konditionalen Diffusionsmodells TMED ein, und (iv) wir stellen BABEL vor, ein Datenset mit semantisch reichen Annotationen, das Aufgaben wie Klassifikation, zeitliche Lokalisierung und Bewegungssynthese unterstützt. Diese Arbeit ebnet den Weg für fortgeschrittene Anwendungen in Animation, virtueller Realität und interaktiven Systemen.

Acknowledgements

We are all on earth to help others. What I can't figure out is what the others are here for.

Joke by W. H. Auden.

Showing gratitude is one of the simplest yet most powerful things humans can do for each other.

Last Lecture, Randy Pausch

Prelude. Most of the others say that the acknowledgements are the most genuine part of a thesis. I keep precisely equal distances from this opinion/fact. On the other hand, I am a fan of the opinion that "whatever we know someone «taught us»." I owe lots of the things I know and things I have achieved to the people around me, explicitly or implicitly. Not only people, but also institutions that gave me the opportunity to start and experience something that I once dreamt about.

Admins. Thanks, Melanie, for your cool, straightforward attitude. For your clear and always honest tone. For solving every single problem and gathering all needed information in the fastest possible way; for Rocko. Thanks, Nicole, for your kind smile, for helping with everything, for making things always possible, and for keeping company in the office. Thanks, Johanna, for being such a successful replacement for Melanie, which is extremely hard. For being kind and helpful. Benjamin, thanks for being available and immensely helpful whenever I needed you. I cannot imagine another person doing your job better than you. If my papers had an extra author, that would be you. Joan, thanks for being the heart of the cluster and helping me parallelize my jobs so elegantly.

Advisors. Gül, thanks for taking the risk of working with me while I was 2.5 years into my PhD. I am glad you taught me your attention to detail and your constant attempts to better organize and structure my work. Thanks for sharing our work and for being there. Thanks for your time and efforts to make me an organized student, for teaching me how to do

science, and for hosting me whenever I needed it. You always made time for me to improve. I will never forget. Michael, you have been a great mentor, a person, and a scientist I will always look up to. You were a treasure with well-hidden, never-ending nuggets. Above all, thanks for teaching me how to be a good scientist and fueling my passion to solve problems. Thanks for teaching me how to manage conflicts and how to bavigate in tricky situations. Thanks for demonstrating such an eye for detail and intuition and always being the questioner who brings everyone on the same page by asking the right questions. Your entrepreneurial, curiosity-driven research spirit is a trait hardly any people have and that I admire and aspire to. I will never forget. Above all, I am thankful for what I am compared to what I was when I started.

Chosen Family. Maria, we met during the last part, but you are my biggest support. I don't think I can find someone as nice, honest, kind, and disarmingly internally beautiful. You are a special kind of bright light in the lives of others. You are the muse. It is hard to be compared. Christos, thanks for shaping my decisions, helping me to navigate through hard moments, always providing me with a house, hugging me, and putting so much effort and time into me. Sappho: Thanks for the big fights, being a different material mirror of myself. Thanks for the affection and for showing how I can be a good version of myself, keeping a critical eye. Markella, thanks for the long call in the middle of when things were dark. Kary: We have been friends for almost 25 years. During my PhD, you were a great listener and a participant in my struggles. Thanks for the tolerance and for being a reminder of the fact that life goes on. Savvas: Thanks for sticking around and being on my side. Yannis, for being a new friend after a long while, for showing me different perspectives on things, and for being my "home" in Paris. Yiorgos, thanks for welcoming me. Marilenia and Marianna, thanks for being my friend and listening to my complaints. Tasos, thanks for giving me a home, a smile, and the thirst for life.

Family. My grandma Marianthi has given me ultimate virtues of altruism, love, and selflessness. Strato, you passed away during one of my first PhD battles. Not seeing you for a last time is a pain, which I am curing by thinking of you. Moreover, Niko, my other grandpa, your silent love and

sharp mind will be missed. You were a great human being. Last, I would like to thank my parents. Even though you know little about my battles, my struggles, my dreams, and my hopes, you tried to be there. At least, I am trying to mine the positives and look into your bright sides. Thanks, my mother, for instilling the love for reading and for being the giver. Thanks to my father for teaching me the value of detail, quality, and perseverance.
Collaborators. Alpàr, thank you for your prompt help whenever asked, for being a real person, and for staying up for my submission. Markos, thanks for teaching me the value of thinking slower and deeper. Muhammed, I have the pleasure and honor to be your friend and collaborator. You hugged me from day one in PS; you welcomed me into the world of 3D humans when I had no idea about it. I will never forget how promptly you welcomed me to work with you. I hope we will do this again in the future. Working with you has been only pleasant. I am happy to see you grow and keep in touch. Nefeli, you offered a helping hand and a listening ear in times of great need. Mathis, thanks for helping me understand tricky parts of your codes, and giving "based" Paris recommendations. Abhinanda, thank you for your calm and kind interactions. Arjun, thanks for showing me that even in nice places bad things can happen. Bugra, thanks for showing what it is to be great, successful, smart, and super humble. Thanks for staying up all night, being positive and helpful as much as you could. Thanks for demonstrating a great character no matter the differences. Fadime, you were the big smile of the office. You gave me ample positivity whenever I needed it. Angela, thanks for giving me the chance to get to know you. I will cherish our walks to lunch, discussions, and exchange of ideas. It was an honor to spend time with you. Edo, thanks for playing basketball and sharing a deep talk and a couple of beers. Sai, thanks for your critical thinking and trying hard. Paola, thanks for making me feel equal and a part of your research.
Colleagues, and Friends. ShShrisha,hanks for being nice to me, putting up with my borderline jokes, respecting my choices, and being a reliable friend. I hope in the future we spend more time both in and out of research. Peter, thanks for being a real person; don't hesitate to reveal your true self, and keep being curious, energetic, and thirsty to learn. Thanks,

Partha, for our nights out, for inviting me into parts of your life, for driving for me. Thanks, Tithi, for trusting me as an advisor. Thanks, Enes, for putting up with my jokes and company. Thanks, Radek, for cheering me up, supporting me, and inviting me. Vassilis, thanks for making me feel comfortable in my PhD. Dimitris, thanks for offering side-boosts of care and positivity from far away. Soubhik, thanks for your spirit, smile, positive attitude, and nights out. Shashank, thanks for trying to make me feel better, listening, and showing me the world of Indian vegetarian food. Thanks for showing me your version of simplicity and peace. Thanks Suraj, for being nice, for taking care of my house, and for being a smiling lab spirit. Lea, thanks for feeling comfort and trust around me. Omid thanks for the peculiar jokes. Giorgio, Tasos, Prerana & Neelay, thanks for allowing me to invade your office and play ping-pong. Thanks to Taylor Obersat for being cheerful and welcoming, Asuka Betler for helping. Claudia Gallatz, thanks for being kind and patient. Tsvetelina, thanks for helping me when I most needed it. Arina and Tomasz, thanks for your silent acceptance. I admire you and wish you all the best. Berna, thanks for being around and making me feel that I have a friend and a person who cares in Tuebingen. Ilya, we met mostly in the end, but your help and kindness will not be forgotten. Thanks, Yinghao Huang, Yao Feng, and Haiwen Feng, for being in the lab after midnight. Thanks to Yuliang Xiu for being an uncensored gate to Chinese culture, food, and trends. Sai, thanks for being a person who trusted me and saw deeper than the surface. I always enjoyed our conversations, and I hope you consider me to be a nice person. Thanks to Vinie for being a "host".

I would also like to thank all the people who have been lost throughout those years —-as it commonly happens-—, but encouraged me for periods longer than they disappointed me.

Contents

List of Figures

List of Tables

List of Abbreviations

CV	**C**omputer **V**ision
MoCap	**Mo**tion **Cap**ture (referring to a lab setting and people being captured markers on)
AMT	**A**mazon **M**echanical **T**urk
LSTM	**L**ong **S**hort-**T**erm **M**emory
RNN	**R**ecurrent **N**eural **N**etwork
GRU	**G**ated **R**ecurrent **U**nit
GT	**G**round **T**ruth
SMPL	**S**kinned **M**ulti-**P**erson **L**inear model
SMPL+H	**SMPL** with **H**ands
SMPL-X	**SMPL** e**X**pressive, with face expressions and hand articulation
VR	**V**irtual **R**eality
AR	**A**ugmented **R**eality
SOTA	**S**tate- **o**f- **t**he- **A**rt
GT	**G**round- **T**ruth
SINC	**SI**multaneous actio**N** **C**ompositions for 3D human motions model
TEACH	**TE**mporal **A**ction **C**ompositions for 3D **H**uman motions model
MotionFix	**M**otion**Fix** dataset
MF	**M**otion**Fix** dataset shortened name
TMED	**T**ext-driven 3D **M**otion **ED**iting model
AI	**A**rtificial **I**ntelligence
AR	**A**ugmented **R**eality
BABEL	**B**ody-**A**ction **B**enchmark with **E**nglish **L**abels
GPT	**G**enerative **P**re-trained **T**ransformer
LLM	**L**arge **L**anguage **M**odel
VLM	**V**ision **L**anguage **M**odel

TEMOS	**T**ext-to-**E**mbodied **MO**tion **S**ynthesis
VIBE	**V**ideo **I**mplicit **B**ody **E**stimation
GAN	**G**enerative **A**dversarial **N**etwork
ACTOR	**A**ction-**C**onditioned **T**ransf**OR**mer VAE
ADE	**A**verage **D**isplacement **E**rror
2s-AGCN	**2-s**tream **A**daptive **G**raph **C**onvolutional **N**etworks for Skeleton-Based Action Recognition
AMASS	**A**rchive of **M**otion **C**apture **A**s **S**urface **S**hapes
APD	**A**verage **P**airwise **D**istance
APE	**A**verage **P**ositional **E**rror
AVE	**A**verage **V**ariance **E**rror
BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
CEE	**C**ontent **E**ncoding **E**rror
CLIP	**C**ontrastive **L**anguage–**I**mage **P**retraining
EMD	**E**arth **M**over's **D**istance
FDE	**F**inal **D**isplacement **E**rror
FFN	**F**eed-**F**orward **N**etwork
FID	**F**réchet **I**nception **D**istance
FLAME	**F**aces **L**earned with an **A**rticulated **M**odel and **E**xpressions
IK	**I**nverse **K**inematics
IMU	**I**nertial **M**easurement **U**nit
IS	**I**nception **S**core
KIT	**K**arlsruhe **I**nstitute of **T**echnology (common name used to refer to KIT Motion-Language dataset)
KL	**K**ullback-**L**eibler
LBS	**L**inear **B**lend **S**kinning
MANO	hand **M**odel with **A**rticulated and **N**on-rigid def**O**rmations
MDM	**M**otion **D**iffusion **M**odel
MMADE	**M**ulti-**M**odal **A**verage **D**isplacement **E**rror
MMFDE	**M**ulti-**M**odal **F**inal **D**isplacement **E**rror
MMM	**M**aster **M**otor **M**ap (format of motion representation)
MSE	**M**ean **S**quared **E**rror
NLP	**N**atural **L**anguage **P**rocessing
NN	**N**eural **N**etwork

NPSS	**N**ormalized **P**ower **S**pectrum **S**imilarity
RGB	**R**ed **G**reen **B**lue
RIFKE	**R**obust **I**nverse **F**unction **K**ernel **E**stimation
SCAPE	**S**hape **C**ompletion and **A**nimation of **PE**ople
SEE	**S**tyle **E**ncoding **E**rror
STAR	**S**parse **T**rained **A**rticulated Human Body **R**egressor
STMC	**S**patial **T**emporal **M**otion **C**omposition Model (from the publication: Multi-Track Timeline Control for Text-Driven 3D Human Motion Generation)
SUPR	**S**parse **U**nified **P**art-based **R**epresentation
TD	**T**ransition **D**istance
TMR	**T**ext-**M**otion **R**etrieval Model
VAE	**V**ariational **A**uto**E**ncoder
PFNN	**P**hase-**F**unctioned **N**eural **N**etwork
LaFan1	Ubisoft **La** **F**orge **An**imation Dataset
t-SNE	**t**-Distributed **S**tochastic **N**eighbor **E**mbedding
AVA	**A**ction **V**ideo **A**tlas
VIA	**V**GG **I**mage **A**nnotation tool
RGBD	**R**ed, **G**reen, **B**lue, and **D**epth
NTU RGB+D	large-scale dataset for RGB-D human action recognition from **N**anyang **T**echnological **U**niversity
HACS	**H**uman **A**ction **C**lips and **S**egments
BP	**B**ody **P**arts

List of Symbols

$\mathcal{J}$	The 24 joints of the skeleton body of the SMPL model
$\mathcal{S}_i$	A natural language motion description.
W_j^i, W_i	j-th word of i-th description, or i-th word respectively.
$\mathcal{H}_i, \mathcal{H}$	A 3D human motion sequence, where each sequence is parameterized by the SMPL body model 6D rotations and translation deltas of the i-th frame or the entire sequence.
v_j^i, v_i	Text features extracted from the current text instructions j-th word of i-th description, or i-th word respectively.
$\mu^{\text{token}}, \Sigma^{\text{token}}$	Input tokens to a transformer that correspond to the learnable parameters of a Gaussian distribution.
$\mathscr{L}$	Smooth L1 loss
$\hat{H}_{i:j}$	Generated motion from i-th to j-th frame, represented by the features used in Chapters 4, 4.
$H_{i:j}$	Ground-truth motion from i-th to j-th frame, represented by the features used in Chapters 4, 4.
H	Entire ground-truth motion, represented by the features used in Chapters 4, 4 .
$\hat{H}$	Entire generated motion.
$I_{i:j}^j$	Features from the j-th action performed action motion used as past motion conditioning, of frames i-th to j-th.
μ^i, Σ^i	Parameters of a Gaussian distribution for encoding current text features and past motion features.
z^i	Latent vector sampled from the Gaussian distribution $\mathcal{N}(\mu^i, \Sigma^i)$.
F_i	Positional encodings of the i-th motion frame (sinusoidal functions).

SEP	Special learnable token used for separating inputs in transformers.
$s_i = [t_s^i, t_e^i]$	A segment in a motion sequence, defined by its start time t_s^i and end time t_e^i (from BABEL dataset).
z^M	Motion latent vector sampled from the distribution $\mathcal{N}(\mu^M, \Sigma^M)$.
z^T	Text latent vector sampled from the distribution $\mathcal{N}(\mu^T, \Sigma^T)$.
$\mathcal{N}(\mu^M, \Sigma^M)$	Distribution for motion latent vectors.
$\mathcal{N}(\mu^T, \Sigma^T)$	Distribution for text latent vectors.
$L_{\mathcal{KL}}^T$	KL divergence loss for the text latent distribution: $\mathcal{KL}(\mathcal{N}(\mu^T, \Sigma^T), \mathcal{N}(0, I))$.
$L_{\mathcal{KL}}^M$	KL divergence loss for the motion latent distribution: $\mathcal{KL}(\mathcal{N}(\mu^M, \Sigma^M), \mathcal{N}(0, I))$.
$L_{\mathcal{KL}}^{M\|\|T}$	KL divergence between the motion and text latent distributions: $\mathcal{KL}(\mathcal{N}(\mu^T, \Sigma^T), \mathcal{N}(\mu^M, \Sigma^M))$.
$L_{\mathcal{KL}}^{T\|\|M}$	Symmetric KL divergence loss for text and motion latent distributions.
L_R	Reconstruction loss.
L_Z	Loss forcing the text latent vector z^T to be close to the motion latent vector, z^M, and vice versa.
M_T	Target motion represented by the 207 features used in Chapter 5.
M_S	Source motion represented by the 207 features used in Chapter 5.
S^e	A natural language description of the edit between two motions.
ϵ_t	Noise added to the target motion at timestep t.
M_T^t	Noised target motion at timestep t, produced by adding noise ϵ_t to M_T.
$\tilde{M}_T^t$	Denoised motion output from the denoiser network at timestep t.
$\mathbb{E}_{(\cdot)}$	Expectation over the $(\cdot)$ random variables.
$D(\cdot)$	Denoiser network function.
E_T	Encoder for the diffusion timestep t.
E_L	Encoder for the edit text condition S^e.
E_M	Encoder for the motion input M_S or M_T^t.
M_S^{enc}	Encoded source motion: $M_S^{\text{enc}} = E_M(M_S)$.
$M_T^{t,\text{enc}}$	Encoded noised target motion at noise level t: $M_T^{t,\text{enc}} = E_M(M_T^t)$.

s_M, s_L	Guidance scales for the source motion M_S and text condition S^e.
$\epsilon_\theta(\cdot)$	Score estimate for the diffusion model, learned using classifier-free guidance.
s_L, s_M	Guidance scales used for the edit text condition, S^e and the source motion condition, M_S.
a_i	Action segments, representing contiguous sets of frames corresponding to an action.
$a_i \rightarrow a_j$	Transition between action a_i and a_j.
$\mathbf{X} = (x_1, \ldots, x_L)$	Movement sequence, where $x_i \in \mathbb{R}^{J \times 3}$ represents the 3D position of $J = 25$ joints in Cartesian coordinates (x, y, z).
$(X_s, \ldots, X_e)$	A segment of human movement corresponding to a raw BABEL action label.
$(X_s, Y_s)^N$	A set of N movement segments and their corresponding action categories.
$y \in \mathcal{Y}_s$	Action category label corresponding to its segment in BABEL.

Chapter 1

Introduction

1.1 Motivation

What does it mean to move? A simple, yet interesting, and ambiguous question. Our motion means much more than just changing location like a simple object. Our movements define how we interact with the world, perform everyday tasks, express our feelings, and even communicate complex ideas without speaking. Every gesture we make, every subtle shift in how we stand or walk, carries meaning. This rich, physical language makes our actions seem natural and understandable to others. Apart from communication, our movements are fundamental for performing daily tasks, like preparing food or commuting to work, and navigating through the world.

In the future, humans and AI will co-exist. Although recently, AI is getting better at understanding things like text and images, it still falls short in a key aspect; its physical presence. The next big step is to create AI that can actually move and interact within our 3D world — this is often called embodied intelligence. Motion is a key component of embodied agents. To achieve this, we need AI systems, or agents, that can perform realistic human movements, capturing the naturalness of humans. This ability would allow them to help us, work with us, and communicate more naturally in both real and virtual environments.

This goal presents an interesting challenge, since human movement is not just about getting from one place to another. It is a powerful way we express ourselves and connect with others. For AI agents to work

alongside people effectively, they need to understand and use this physical language. This means creating motion that looks natural and can follow instructions.

An intuitive control to connect with AI is language. However, turning text instructions into realistic 3D human motion, is challenging. Text can describe actions in many ways, and sometimes the description is ambiguous. Moreover, human movement involves coordinating many body parts smoothly, and performing multiple motions one after the other — making it complex and compositional. AI models need to learn how to translate the meaning from text into movements that respect the rules of physics and how human bodies work.

Moreover, generating motion from text is not just about producing any plausible sequence of movements. Because a single text instruction can correspond to multiple valid motions, it becomes crucial to have mechanisms that allow fine-grained control over the output. Users should be able to easily edit and adjust generated motions based on high-level ideas, offering flexibility beyond what traditional animation methods can provide. This thesis addresses these challenges by developing better ways to create controllable, realistic human motion directly from text.

The idea of understanding and recreating human motion has fascinated people for centuries. Long before AI, Aristotle wondered how and why we move the way we do [Kosman, 1969; Klette and Tee, 2008]. Newton and Da Vinci laid the first scientific foundations, studying the mechanics of joints, forces, and the structure of the human body [Kosman, 1969]. Such works gave us the first "engineering" perspectives on joints and the human body. Over time, researchers began using early cameras to analyze how athletes moved [Fenn, 1931; Davies, 1968], opening the door to more detailed, data-driven studies. Further progress in modeling human motion led to the first data-driven approaches for generating cyclic actions, such as walking and running [Ormoneit et al., 2000; Taylor et al., 2007].

This led up to a series of work on conditional [Tevet et al., 2023; Petrovich et al., 2021; Petrovich et al., 2022; Athanasiou et al., 2023; Athanasiou et al., 2022; Moltisanti et al., 2022; Li et al., 2021; Gong et al., 2023] and unconditional [Zhang et al., 2021b; Pavllo et al., 2018]

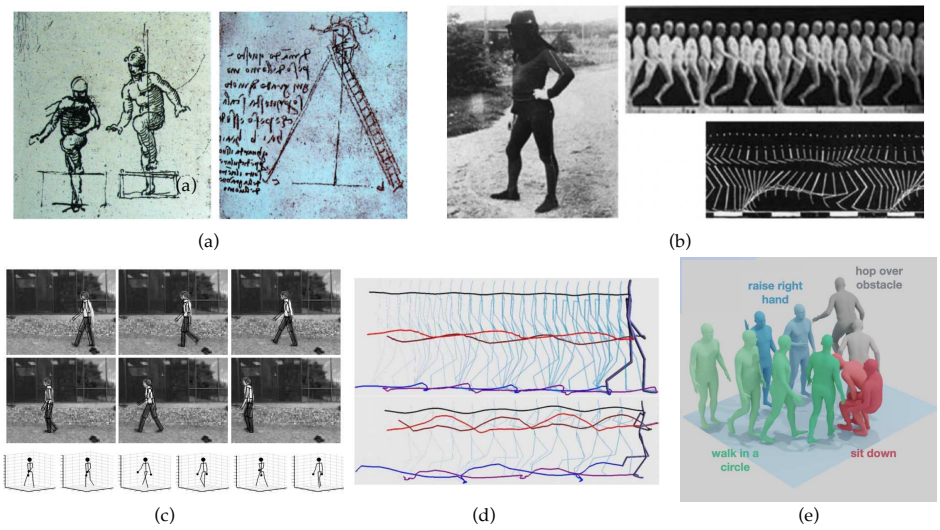


FIGURE 1.1: **Historic overview of research on human motion:** Starting from (a) da Vinci sketches [Klette and Tee, 2008], and (b) study of sprinter motions FIX, human motion generation has progressed to generate cyclic motions like those seen in [Ormoneit et al., 2000; Abdal et al., 2020] (c), [Taylor et al., 2007] (d) and reaching towards generating full sequences of different actions from instructions such as in (e) [Athanasiou et al., 2022], presented in Chapter 3.

motion generation and prediction [Martinez et al., 2017; Mao et al., 2019]. Unconditional generation focuses on motion generation [Zhang et al., 2021b] and prediction [Martinez et al., 2017], while conditional uses signals such as music [Pu and Shan, 2022], actions [Guo et al., 2020], and language instructions [Petrovich et al., 2022]. Recently, many works focus on generating temporal and spatial motion compositions [Huang et al., 2024; Zhang et al., 2023c] and editing them to achieve detailed control [Goel et al., 2024]. This body of work highlights the importance of motion as a part of creating virtual assistants and humans just like us. Our work in Chapters 3, 4 pioneered the learning of such temporal and spatial compositions respectively, and Chapter 5 introduces the task of text-based 3D human motion editing. A brief illustration of this historic overview of the described progress is presented in Figure 1.1.

1.1.1 Why do we need 3D human motion from text?

3D human-like moving agents. The embodiment age of AI is approaching. Interactive VLMs and LLMs have transformed the field by efficiently compressing and understanding visual and textual information. Now, the need for AI agents to have a body is becoming clear. A physical form gives them the ability to perceive, touch, sense, and interact — building a kind of adaptive character. It gives them a way to learn from the world, much like humans do.

Embodied agents are also far more engaging than traditional computers or chatbots [Marsi and Rooden, 2007]. They can assist, entertain, and collaborate with people in ways that feel more natural and enjoyable [Kipp et al., 2006; Beun et al., 2003; Mulken et al., 1998]. To build such agents, human motion is essential. Human movements must look natural, adapt to different situations, and be able to mirror the wide variety of actions we perform every day. These actions happen in sequence (e.g., "chopping onions and then putting them in a bowl"), in parallel (e.g., "grabbing the oil while stirring the pot"), and with fine control (e.g., "stir longer" or "grab the salt from higher up"). Capturing this flexibility and precision is a key step toward creating truly embodied intelligence.

Conditioning models with text instructions. Text has been a popular conditioning signal for generating images [Mansimov et al., 2015; Reed et al., 2016], and has attracted wide and vast interest [Ramesh et al., 2021; Saharia et al., 2022]. Early progress stems from LLMs [Brown et al., 2020; Gemini Team et al., 2024; Liu et al., 2023; Touvron et al., 2023b; Touvron et al., 2023a] and has been extended to MLLMs which include further modalities such as audio and images. Even text-only have been very successful for dialogue-like user experiences. It has also been used to create content and generate data [Brooks et al., 2023]. Recently, it has been used to generate videos [Brooks et al., 2024], and to interact with 3D bodies [Lin et al., 2024; Feng et al., 2024; Delmas et al., 2022; Delmas et al., 2023]. Such tools increase significantly the engagement of users with AI tools. This highlights the importance of text used as a condition in AI

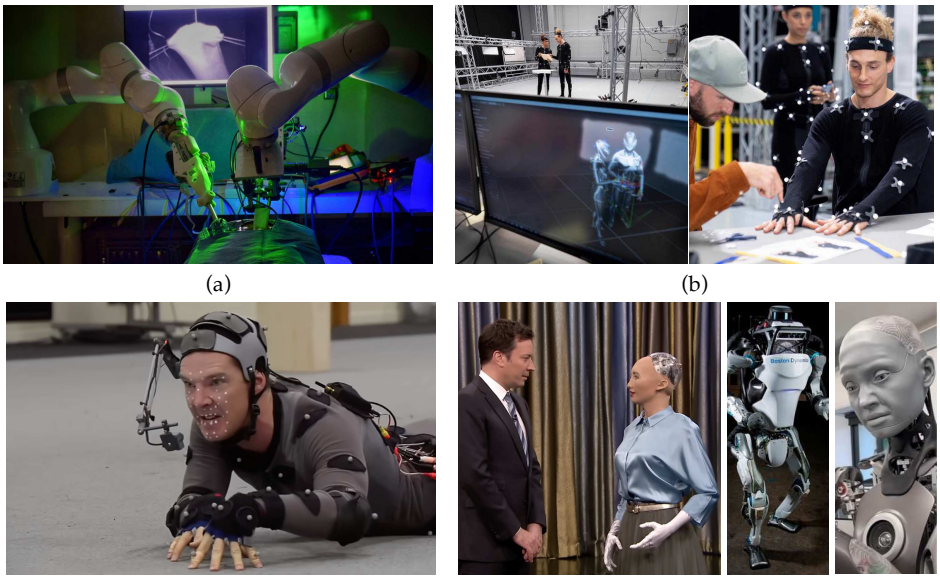


FIGURE 1.2: **3D Human motion generations use cases:** Different applications of 3D Human Motions. We demonstrate (a) a robot performing surgery[1], (b) subjects being set up and getting captured in a MoCap studio[2], (c) the MoCap backstage from the movie: "Behind-the-Scenes of The Hobbit: Desolation of Smaug", (d) different full-body robots interacting with people, or moving[3].

models. Natural Language is an intuitive and abstract control with which the average person is familiar, and makes AI agents approachable in the way that we approach humans.

3D human motion generation until today. Given the growing need for embodied agents controlled by natural language, creating realistic 3D human motion has become essential for many fields. Applications include animation, gaming, virtual reality, robotics, healthcare, and the film industry, as shown in Figure 1.2. These areas demand high-quality, realistic motion to create immersive and interactive experiences.

[1] https://hub.jhu.edu/2022/01/26/star-robot-performs-intestinal-surgery/

[2] https://ps.is.mpg.de/pages/motion-capture

[3] https://www.discovermagazine.com/technology/4-robots-that-look-like-humans, "Tonight Showbotics: Jimmy Meets Sophia the Human-Like Robot"

However, motion capture (MoCap) methods are expensive, time-consuming, and limited in flexibility. Setting up a subject, as shown in Figure 1.2(b), requires a labor-intensive process involving specialized suits and complex multi-camera setups. Actors must follow strict instructions and perform in constrained studio spaces, which makes it difficult to capture diverse, natural, everyday actions. Even simple tasks like interacting with objects involve a long post-processing pipeline, such as cleaning the captured markers and fitting a target body shape to the processed marker positions. Capturing fine details—such as precise body part positioning, motion speed, or individual style—is particularly challenging.

Moreover, traditional pipelines cannot easily capture core traits of human motion:

- **Temporal hierarchies.** Humans naturally perform sequences of actions, one after another. Capturing all possible action combinations is impossible, and simply stitching separate motions together fails to produce realistic results.
- **Spatial hierarchies.** Humans move different body parts independently, often performing multiple actions at once to achieve a goal. The number of possible combinations grows exponentially, making it infeasible to capture them all.
- **Adaptability.** Humans constantly adjust and refine their movements based on instructions (e.g., from a trainer) or the environment. Capturing this ability to adapt and personalize motion is extremely difficult in traditional MoCap settings.

1.1.2 Applications of 3D human motion generation from text

Use cases of text-to-motion generation. 3D human motion generation is fundamental for creating realistic experiences of humans moving and interacting in digital environments.

The need to automate and democratize this process is growing rapidly. In film and animation, lifelike character movements enhance storytelling and viewer engagement. In gaming, realistic motions boost immersion and

player interaction. In virtual and augmented reality, believable human actions are essential for creating responsive, interactive experiences. In healthcare, accurate motion modeling supports rehabilitation and physical therapy through realistic patient simulations. In robotics, replicating human motions improves collaboration and task execution. In surveillance and security, analyzing human movement can strengthen behavior recognition and anomaly detection.

Text, while sometimes imprecise, remains a natural and accessible way to control motion generation, offering a human-friendly interface between users and machines.

Use cases of 3D human motion editing from text. Beyond generation, there is an increasing need for flexible and intuitive motion editing tools. Animators and developers often require precise control over synthesized movements to meet artistic or functional goals. While text is powerful for high-level instructions, it is not always enough to fully describe complex motions. Providing an initial animation serves as a strong prior, allowing users to refine motion by adjusting directions, speed, style, or fine details — for example, raising the left hand higher.

Editing from text enables finer, more semantic control, bridging the gap between creativity and technical precision.

Summary. Advancing 3D human motion generation and editing is crucial for animation, gaming, virtual reality, robotics, healthcare, and the film industry. Human motions are inherently complex and nuanced, making them difficult to model. Traditional motion capture methods, though effective, are costly and time-consuming. Generating and editing motions from natural language offers a scalable, cost-effective alternative and brings us closer to teaching computers the intricate language of human movement.

1.2 Objectives

As previously mentioned, human motion is complex and has a hierarchical structure across both time and its expression space (the human body). This thesis tackles the task of 3D human motion generation from text instructions. In particular, given a sequence of text, we generate the corresponding 3D human motions and their transitions (Chapter 3); given multiple text descriptions, we generate their spatial composition — actions being performed simultaneously (Chapter 4); and given a 3D human motion and a text describing an edit to that motion, we generate the edited motion (Chapter 5).

1.2.1 Temporal hierarchies of human motion

The first focus is generating a sequence of actions and the transitions from one to another given a sequence of text, in Chapter 3. The TEACH model (Temporal Action Composition for 3D Humans) is designed to generate realistic 3D human motions based on natural language instructions. The core goal of TEACH is to generate smooth and realistic human motion sequences that transition between multiple actions described by text. This is achieved through a variational encoder-decoder NN architecture.

TEACH generates multiple sequential actions (e.g., "turn right," "walk forward," "sit down") by incorporating context from the previous action into the generation of the next one. An illustration of this is shown in 1.3 (left). This results in coherent and natural transitions between actions, a challenge that baselines struggle with.

1.2.2 Spatial hierarchies of human motion

Our second focus, in Chapter 4, is the spatial composition of 3D human motions. SINC is designed to tackle the generation of multiple, simultaneous actions from textual descriptions. To achieve this, we introduce a GPT-guided synthetic data generation process, which combines action-relevant body parts from different motions to create new compositions. This method automatically produces training

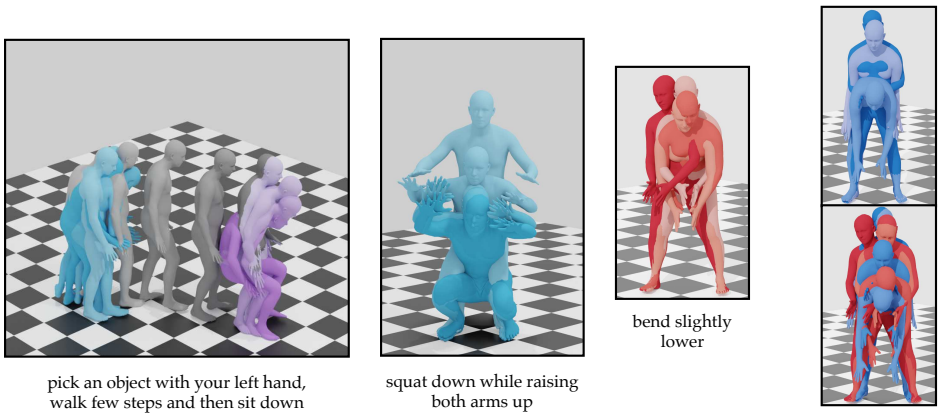


FIGURE 1.3: **Thesis goals:** We illustrate the three main objectives of this work: (left) generating a sequence of actions from a sequence of texts (Chapter 3), (middle) generating fine-grained simultaneous actions from a text describing multiple activities (Chapter 4), and (right) editing an initial motion based on a textual edit description (Chapter 5).

data, addressing the scarcity of real examples for simultaneous action generation. The model learns to associate specific actions to different body parts—for example, performing walking with the legs while simultaneously raising a hand—all within a single, coherent motion. We show an example of a spatial composition in 1.3 (middle).

1.2.3 Finegrained control and editing

Our final focus, in Chapter 5, is 3D human motion editing from text. We introduce a framework that edits an existing motion based on a natural language description of the desired change. A key contribution is the creation of the MotionFix dataset, which pairs source motions with target motions and corresponding textual edit instructions. Using this data, we train a conditional diffusion model called TMED, which learns to modify motions in line with textual edits. TMED enables a range of edits, including adjustments to body part positions, motion speed, and spatial changes. As we demonstrate in 1.3 (right), the model starts from an initial motion–which we refer to as source (in red)– and a text describing the edit and generates an edited motion (shown in blue).

1.3 Challenges

In this section, we present the current challenges of motion generation conditioned on text instructions. Along with the summary of current challenges, we highlight the solutions we offer in different parts of our work.

Existing data have spurious correlations and are limited. In models trained with MoCap datasets, spurious correlations happen when the model mistakenly links unrelated actions due to patterns in the captured data. For example, if raising an arm and walking are often combined in the training data, the model may mistakenly infer they always go together — i.e., assume someone always walks when raising their hand. However, these actions should be independent. This creates false correlation, leading to poor motion-text alignment, where the model connects actions that should not be related. This makes finegrained control hard. Moreover, it is challenging to instruct the actors and capture data for all different combinations of motions. There are exponentially many temporal and spatial compositions. In addition, synthetically creating such compositions via animation systems requires manual work and is not straightforward. We attempt to address this challenge, by creating synthetic data for spatial compositions in Chapter 4, and by generating temporal compositions for arbitrary sequences of actions in Chapter 3. In the case of spatial compositions, we introduce an LLM-based framework to automatically blend compatible motions which can be explored at https://huggingface.co/spaces/atnikos/sinc-synthetic. For more details, we refer the reader to Section 4.5.1 in Chapter 4.

Language is not enough to control motion. Language alone often falls short in providing the detailed control needed for human motion generation, as words frequently lack the specificity required for fine-tuning movements. For example, saying "raise your arm" does not specify how high, how fast, or at what angle the arm should be raised. Subtle aspects like adjustment of joint angles, timing, or simultaneous multipart actions

are difficult to describe in text. Additionally, natural language can be ambiguous, making it difficult to create highly precise and nuanced motions based on text descriptions alone. Text is an intuitive conditioning signal, but it is often too abstract to provide sufficient constraints on its own. Incorporating an additional control, such as an input motion that can be edited using text, introduces a more concrete constraint—enabling finer-grained control over the generated motion. We introduce a new dataset and attempt to solve this problem in Chapter 5.

Evaluating motions is hard. Evaluating human motion generation from text is challenging due to the subjective nature of the task and the complexity of human motion. Unlike tasks with clear, measurable outcomes (e.g., classification, reconstruction and similar "discriminative" tasks), the quality of generated motions is harder to define. Additionally, small errors in motion can have a large impact on the perception of realism. Metrics like realism, smoothness, and consistency with the input text are difficult to quantify. The lack of standardized benchmarks and reliable evaluation metrics further complicates consistent evaluation in this domain. This highlights the need for task-specific metrics and backing up qualitatively any evaluation.

In our work, we introduce several metrics. Transition Distance, in Chapter 3, measures the distance between the last frame of the previous action and the first frame from the next one. This measures how smoothly the model transitions from the one action to the other. TEMOS score, in Chapter 4, compares motions in a motion feature space, proving much more reliable than coordinate-based metrics previously proposed. Finally, motion-to-motion retrieval metrics, in Chapter 5 tackle the challenge of measuring how closely motions align with the edited text and remain close to the initial animation.

Amount of data. The data scarcity challenge in human motion generation from text stems from the limited availability of high-quality motion datasets paired with textual annotations. This problem is even more pronounced in finegrained settings, where constraints are highly specific

(e.g., raising a hand while sitting down). Collecting large, diverse datasets with detailed motion descriptions is both expensive and time-consuming.

Without sufficient data, models struggle to learn a wide range of realistic human motions, leading to poor generalization. Moreover, small datasets increase the risk of overfitting, where models become too specialized to specific examples and fail to capture the natural variability of human movement. This significantly limits the effectiveness of motion generation systems.

Labeling motions with text presents additional challenges. Human movements are complex and often involve subtle variations that are difficult to describe precisely. Different annotators may interpret the same motion differently, leading to inconsistent labels. Describing motions with fine detail, especially for small or overlapping actions, requires careful observation and nuanced language that captures both movement and context. The inherent subjectivity of language further complicates achieving consistent annotations across a dataset.

To address these issues, we contribute the BABEL dataset, presented in Annex A, which provides dense text annotations for 3D human motions. BABEL labels every frame of motion, enabling the study of temporal and spatial compositions naturally, as explored in Chapters 3 and 4.

Motion realism & physical plausibility. Motion realism and physical plausibility are critical challenges in creating believable animations and simulations. To make virtual characters move naturally, it is not enough for the motion to be smooth; it must also respect the rules of real-world physics, such as gravity. Moreover, there should be no self-penetration, and feet-ground contact should be appropriate. Another key difficulty lies in generating physically plausible motion without incurring excessive computational cost — such as physics simulation. When movements do not follow realistic physical behavior, they look uncanny and disrupt the viewer's attention. Furthermore, if a motion generator falls short in obeying physical laws, it makes the job of the animator harder and makes it impossible to use such motions for applications that require precise movement, such as robotics. Addressing this challenge often requires

improving motion generation methods, incorporating real-world data, such as motion capture, and building better models. Better body models could use skeleton rigs that are closer to the real human skeleton, and enable more accurate articulation — for example, by adding more joints in key regions such as the lower foot, which is key to high-quality motion generation.

1.4 Contributions

1.4.1 Publications

The research conducted during this PhD has led to the following publications:

- **Nikos Athanasiou, Mathis Petrovich, Michael J. Black, Gül Varol** "TEACH: Temporal Action Compositions for 3D Humans" (3DV 2022) (Chapter 3).
- **Nikos Athanasiou, Mathis Petrovich, Michael J. Black, Gül Varol** "SINC: Spatial Composition of 3D Human Motions for Simultaneous Action Generation" (ICCV 2023) (Chapter 4).
- **Nikos Athanasiou, Alpár Cseke, Markos Diomataris, Michael J. Black, Gül Varol** "MotionFix: Text-Driven 3D Human Motion Editing" (SIGGRAPH Asia 2024) (Chapter 5).

The author of this thesis also contributed significantly in other related publications:

- **Abhinanda Punnakkal*, Arjun Chandrasekaran* Nikos Athanasiou, Alejandra Quiros-Ramirez, Michael J. Black** "BABEL: Bodies, Action and Behavior with English Labels" (CVPR 2021) (Annex A).
- **Muhammed Koçabas, Nikos Athanasiou, Michael J. Black** "VIBE: Video Inference for Human Body Pose and Shape Estimation" (CVPR 2020).

- **Markos Diomataris, Nikos Athanasiou, Omid Taheri, Xi Wang, Otmar Hilliges, Michael J. Black** "WANDR: Intention-guided Human Motion Generation" (CVPR 2024).

from which the first one is included in the Annex of this manuscript. VIBE [Kocabas et al., 2020] was one of the first methods to estimate human motions from videos, which paved the future for increasing the size of pseudo-groundtruth data with 3D human motions from internet videos. The author also contributed to WANDR [Diomataris et al., 2024], which is an autoregressive motion prior that focuses on wrist control.

1.4.2 Software contributions

The code for all the publications has been released on GitHub, along with the pre-trained models. Together, the projects currently have over a thousand stars on GitHub. Below we provide additional information on each project, including links to the project websites where code, pre-trained models, videos, and demos can be found.

- **TEACH**: The code and pre-trained models for generating 3D human motions from a sequence of text descriptions are available as part of the project at `https://teach.is.tue.mpg.de`. TEACH enables the synthesis of human motions that correspond to a series of actions described in natural language. The website provides an additional version of BABEL with refined text labels, the released code along with all the trained models and baselines. A simple demo is included in the code for a user to type a sequence of text and get the temporally composed motion as output.

- **SINC**: The code and pre-trained models for generating 3D human motions from simultaneous action descriptions are released as part of the project presented in `https://sinc.is.tue.mpg.de`. SINC focuses on spatial compositions, allowing the generation of complex human motions where multiple actions occur concurrently. The code additionally includes the code to

create, and a demo to explore, synthetically created data at https://huggingface.co/spaces/atnikos/sinc-synthetic.

- **MotionFix**: The code, data, and pre-trained models for editing existing 3D human motions based on textual descriptions are released as part of the project presented in https://motionfix.is.tue.mpg.de. The website contains an exploration website of our dataset, a demo for users to edit their motion, the evaluation and pre-trained models used in the paper. MotionFix, along with the TMED model, provides a framework for finegrained motion editing, enabling precise control over motion synthesis and modification based on user input.

These contributions provide robust tools for the research community, facilitating further advancements in the field of 3D human motion generation and editing.

- **BABEL**: The code, data, and pre-trained models for BABEL are released as part of the project presented in https://babel.is.tue.mpg.de. The project website contains an analysis of the action labels and demos of the interfaces which were used during the creation of the dataset.

1.5 Thesis outline

Including this introduction, this thesis is organized into six chapters. An Annex is also provided, which includes some additional analysis and discussion on BABEL dataset [Punnakkal et al., 2021].

In **Chapter 2 (Background)**, we begin by presenting the generative models relevant to conditional and unconditional human motion synthesis, which are building blocks to our research. Next, we present a brief overview of prior work in 3D human motion synthesis from different conditions, and focus on the ones using text and aiming for finegrained control. Finally, we present a summary of evaluation approaches for generative motion models and an overview of the current datasets, as

we contribute a new benchmark (datasets & set of metrics) in Chapter 5, synthetic data (Chapter 4).

In **Chapter 3 (TEACH)**, we address the problem of temporal action compositions. TEACH generates 3D human motions that correspond to a series of natural language descriptions, along with the transitions between the actions. We design a transformer-based approach that operates non-autoregressively within individual actions and autoregressively across action sequences. We introduce several baselines, based on single action models, interpolation techniques and available training data for temporal compositions. Finally, we introduce a new metric, transition distance, to measure the effectiveness of different methods temporally combining motions. Our hierarchical method allows the generation of smoother and temporally coherent motions from textual descriptions compared to the baselines. TEACH is the first method that systematically dealt with temporal compositions of human motion.

In **Chapter 4 (SINC)**, we focus on spatial hierarchies for the synthesis of 3D human motions based on textual inputs describing simultaneous actions. Given that MoCap data rarely contains simultaneous actions, we need to create data to solve this task. To have such data, we need to identify the associated body parts when performing an action. To solve this, we observe that powerful language models, like GPT-3, know about the body parts involved in the action from the text description. Using such body part labels, we extract information about body part involvement in actions and use this to introduce a baseline method from single action motion generation models and create synthetic training data from MoCap. Using SINC synthetic data, we enable the generation of realistic and coherent simultaneous motions, surpassing previous methods and single model baselines, in spatial compositionality. Our model SINC, can generalize to unseen action combinations, and we show quantitatively that it can perform even triplets of actions.

In **Chapter 5 (MotionFix)**, we present MotionFix with the TMED model for editing existing animations. First, we address the challenge of data scarcity by semi-automatically collecting a dataset of triplets comprising a source motion, a target motion, and an edit text. We enable

this semi-automatic approach by using a motion encoder to find suitable candidate pairs, which can be easily described with an edit text. Then, using this dataset, we train a conditional diffusion model, TMED , that takes both the source (initial) motion and the edit text as inputs. Given a 3D human motion and a textual description of the desired modification, our model TMED generates an edited motion as specified by the text. We introduce several baselines based on text-to-motion models, which are also using LLM labels for which parts they should edit. TMED demonstrates superior performance over models trained solely on text-motion pairs. Our approach establishes a new benchmark with retrieval-based metrics for motion editing. TMED model is the first model for the text-based 3D human motion editing.

In **Chapter 6 (Conclusions & Future Directions)**, we present a summary of the contributions made in this work. We conclude by discussing the limitations of our work and outlining potential directions for future research.

In the **Appendix**, we present our work on creating a dataset of human motions paired with natural language descriptions. BABEL, the first dataset to provide per-frame natural language labels for human motion, is analyzed and extensively used in **Chapters 3** and **4**.

Chapter 2

Background

In this chapter, we start by introducing the main building blocks used in this work in Section 2.1. Specifically, we briefly describe generative models, and transformers, which are the main technical components used in our work. Then, we focus on the different representations of the human body and different features, commonly used in 3D human motion generation. Next, in Section 2.2 we present an overview of prior work on 3D human motion generation, focusing on motion generation conditioned on text and on works aiming for detailed control. Section 2.3 describes the challenges and progress on evaluating human motion generation. Finally, we provide an overview of the datasets that contain motion sequences along with language labels in Section 2.4.

2.1 Preliminaries

2.1.1 Transformers

Transformers [Vaswani et al., 2017] have revolutionized the field of deep learning, because of their ability to scale well and easily, as the data grows bigger. This was initially shown in natural language processing (NLP) [Brown et al., 2020] and more recently across different domains of the computer vision community [Dosovitskiy et al., 2021; Liu et al., 2021; Carion et al., 2020; Yuan et al., 2021; Zhang et al., 2021a]. Transformer's success has also enabled significant advances in multimodal tasks such as text-to-image synthesis [Ramesh et al., 2021; Saharia et al., 2022; Wei et al., 2021]. Unlike traditional sequence models such as recurrent

NNs (RNNs) [Rumelhart et al., 1986] and long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997] networks, transformers can process entire sequences simultaneously, which allows them to capture long-term dependencies between sequential elements more effectively than RNNs. A key element that enables this is the attention mechanism, which is briefly described next.

Attention mechanism The attention mechanism—first introduced by Bahdanau et al. [2014]—computes for each token a weighted sum of all token representations. Its computation, given a sequence of tokens, involves three key components: the Query (Q), which represents the current token or element we are focusing on; the Key (K), which represents all tokens or elements in the sequence; and the Value (V), which is the information from all tokens that should be aggregated based on the attention weights. Given input embeddings, we project them into Queries Q, Keys K, and Values V. The raw scores are computed as $QK^T/\sqrt{d_k}$. Then, softmax is applied to produce the attention weights that sum to one. Those weights are used to average the Values V. Here, d_k is the dimension of the key vectors to prevent them from exploding (becoming too big). The softmax function ensures that the attention weights sum to 1, which produces a weighted average that naturally emphasizes the most relevant tokens over the sequence. Because softmax is smooth and differentiable, those probabilistic weights can be optimized safely with gradient descent, avoiding vanishing-gradient or scale issues. Softmax also ensures that the elements of the sequence are weighted appropriately, and without over-relying in certain elements of the sequence. This lets the model pull in information from the most relevant parts of the sequence. The scaled dot-product attention computes relationships between tokens by comparing queries, keys, and values. The core of the transformer lies in its ability to capture dependencies in data using attention. When Q, K, and V come from the same sequence it is called self-attention; when they come from different sequences it is called cross-attention.

Transformer architecture The Transformer [Vaswani et al., 2017] arranges these attention modules into an encoder–decoder structure. The transformer architecture consists of two main components: the encoder and the decoder. The encoder extracts features from the input sequence, while the decoder generates the output sequence. The encoder is a stack of identical layers, each containing multi-head self-attention followed by a position-wise feed-forward network, with residual connections and layer normalization around each sub-layer. Multi-head here refers to multiple self attention operations that are performed in parallel to capture different relationships between the features of the elements of the input sequence. The decoder has the same two sub-layers plus an additional masked self-attention step to prevent tokens from seeing future positions, and a cross-attention stage over the encoder's outputs. This setup supports both autoregressive and full-sequence decoding; in this thesis, we use full-sequence decoding for efficient motion synthesis.

2.1.2 Generative models

Generative models are a class of machine learning models designed to generate new data instances that match the data distribution. Unlike discriminative models, which classify input data, generative models learn the underlying distribution of data to produce new, plausible examples. Two prominent approaches to generative modeling – used in this thesis – are variational autoencoders (VAEs) [Kingma and Welling, 2014] and diffusion models [Sohl-Dickstein et al., 2015; Ho et al., 2020]. Alternative approaches include Normalizing Flows [Rezende and Mohamed, 2015; Papamakarios et al., 2021], and Generative Adversarial Networks (GANs) [Goodfellow et al., 2014]. Normalizing Flows are generative models that use invertible transformations to map a simple distribution (e.g., Gaussian) into a complex target distribution (e.g., the data distribution). The key benefit is that these transformations are invertible, allowing both efficient generation and likelihood estimation. The model learns to maximize the likelihood of the data under the transformed distribution. GANs consist of two NNs: a generator and a

discriminator, which engage in a competition. The generator creates fake data to mimic real data, while the discriminator tries to distinguish real from fake data. Its adversarial training process pushes the generator to produce more realistic outputs. However, GANs can be difficult to train due to issues like mode collapse [Goodfellow et al., 2014; Salimans et al., 2016]– i.e., the generation of a limited set of examples which can trick the discriminator but do not capture well the full data distribution. GANs approximate the data distribution and do not usually provide with an exact estimate of the likelihood, as opposed to Normalizing Flows.

Variational autoencoders

Kingma and Welling [2014] introduced VAEs, which are probabilistic generative models based on the concept of autoencoders. In a traditional autoencoder, the model compresses data into a lower-dimensional latent space (encoding) and then reconstructs the input from this compressed representation (decoding). The key distinction of VAEs lies in their variational approach, where the latent space is modeled as a probability distribution, typically Gaussian. This allows VAEs to generate new data by sampling from this latent distribution.

The VAE framework consists of two main components, the encoder, and the decoder. This encoder network maps input data to a probability distribution over the latent space. The encoder takes the input data and processes it through several layers, typically consisting of convolutional or fully connected layers, to extract meaningful features. The output of the encoder is parameterized as two vectors: the mean (μ) and the variance (σ^2) of the latent distribution. By expressing the encoder's output as a mean (μ) and variance (σ^2) the model can smoothly learn the shape of the latent distribution and draw samples that produce diverse, data-driven outputs.

The decoder is a NN that reconstructs the original data from the latent variable sampled from the learned distribution. It takes the sampled latent vector as input and processes it through several layers, which can also be convolutional or fully connected, depending on the data type. The decoder aims to generate data that closely resembles the original input.

The output of the decoder is typically a probability distribution over the data space, allowing for the generation of new data instances. The decoder's architecture is designed to capture the complex relationships between the latent space and the data space, enabling it to produce high-quality reconstructions.

The loss function of VAEs combines a reconstruction loss and a regularization term based on the Kullback-Leibler (KL) divergence. The KL divergence penalizes the difference between the learned latent distribution and a prior (usually a standard Gaussian), ensuring that the latent space remains continuous and well-structured.

Diffusion models

Diffusion models [Sohl-Dickstein et al., 2015; Ho et al., 2020] are probabilistic generative models that have achieved state-of-the-art performance in image generation, both unconditionally [Ho et al., 2020] and when conditioned on text [Nichol et al., 2021; Rombach et al., 2022]. The main idea behind diffusion models involves a Markov process, where data is gradually transformed into noise through a forward diffusion process, and a learned reverse process is trained to iteratively denoise the data back into realistic samples.

The forward diffusion process systematically adds small increments of Gaussian noise to clean data over multiple time steps, following a predefined noise schedule, until the data is indistinguishable from pure noise. Formally, given data points x_0, a sequence $x_1, \ldots, x_T$ is generated by repeatedly adding Gaussian noise, resulting in a tractable noise distribution, $q(x_T) \approx \mathcal{N}(0, I)$..

The reverse denoising process is then modeled by training a neural network to invert this noising procedure. The model learns conditional distributions, $p_\theta(x_{t-1} \mid x_t)$,, aiming to recover the original data from noisy observations at each step. Training leverages variational inference to minimize a simplified evidence lower bound (ELBO), effectively aligning the model's predicted denoising steps with the true underlying data distribution [Ho et al., 2020]. This process has clear connections to

denoising score matching and stochastic differential equations, offering strong theoretical grounding for these models [Song et al., 2021b].

Diffusion models can be viewed as probabilistic generalizations of denoising autoencoders, chaining multiple denoising steps into a coherent generative framework [Vincent, 2011]. Several subsequent studies improved their theoretical understanding [Kingma et al., 2021], developed accelerated sampling methods such as DDIM [Song et al., 2021a], and introduced latent diffusion techniques to efficiently scale to high-resolution images [Rombach et al., 2022]. Due to their robustness against common GAN pitfalls like mode collapse and their ability to model complex data distributions precisely, diffusion models have become foundational tools across diverse generative tasks.

2.1.3 Representations of 3D motion

3D human motion representation is a critical component of motion systems. Representing humans has evolved from early photographic techniques to advanced computational models. The first methods, attempted to represent the human body using 2D overlapping rectangles [Hinton, 1976] or hierarchical cylinders [Marr and Nishihara, 1978]. This laid the groundwork for analyzing motion. Surface models like SCAPE [Anguelov et al., 2005] introduced realistic body shapes but lacked compatibility with modern systems. SCAPE does not have an underlying skeleton, and optimization is needed to get the vertices for each new pose and shape. The SMPL model [Loper et al., 2015] used LBS to get the vertices, a shape-based skeleton and a rich learned shape space, becoming the standard 3D human representation. Regarding motion features, different rotation and velocity features, based on local or global coordinate frames, have been used. This thesis builds on SMPL-based representations of human motion. A short overview of the different attempts to represent humans, motion and different features used to train generative models for human motion is provided in this Section.

Early attempts to represent 3D human motion

The representation and recognition of human body motion have been extensively studied through various models and approaches. In this thesis, we focus on representations that are capable of representing the full human body in a human-like form. Various representations have been proposed motivated by human anatomy using skeletal representations. In the 1800s, Eadweard Muybridge captured human motion as sequences of photographs, experimenting with 'atomic' actions by stacking rapid-fire images. These sequential images laid the groundwork for both cinematic motion studies and biomechanical analysis, eventually inspiring the first articulated human models that captured skeleton-driven movement. The Graphical Marionette [Maxwell, 1983] introduced a marionette-like humanoid inspired by the anatomy of the human skeleton. Hinton [1976] introduced a method to interpret visual data, which iteratively refines possible shapes until they are consistent. This method demonstrated its application in human pose with overlapping rectangles as puppets and influenced later advancements in machine learning and NNs. Marr and Nishihara [1978] proposed a model for recognizing 3D objects based on their spatial structure, using a hierarchical representation of cylinders. Hogg [1983] introduced a vision system to recognize human walking patterns using a hierarchical model to represent the human body. This work was one of the first to analyze human motion from a learning/computational perspective. Similarly, Rohr [1994] presented a model-based method to recognize human movements, particularly walking, in image sequences. They used a 3D model made of cylinders and matched the model's projected contours to image data. Gavrila and Davis [1996] were the first to use multi-camera systems. They developed a system to track human movements in 3D using multiple camera views, employing a graphical human model to match real human appearances in images. This approach eliminated the need for markers, enabling natural, unconstrained motion tracking and advancing multi-view human tracking in 3D computer vision. Ju et al. [2002] introduced a method to track human limb motion using a "cardboard person model," where limbs

are represented as planar patches. Sidenbladh et al. [2000] introduced a probabilistic method for tracking 3D human figures using 2D image sequences, advancing articulated motion tracking for human figures, while Ormoneit et al. [2000] introduced methods to learn and track periodic human motion from video, using 3D motion data. These works laid the ground for realistic and detailed human motion representations.

3D Human motion surface models

Plänkers and Fua [2001] presented a 3D model-based approach to recover both body shape and motion from video sequences, enabling realistic 3D whole-body modeling and tracking. Allen et al. [2003] introduced a method for reconstructing and parameterizing human body shapes from 3D range scans, capturing a wide range of body shapes for applications in body shape analysis, model synthesis, and editing. The SCAPE model captured both articulated and non-rigid deformations, rather than prior work, allowing for realistic 3D human models in various poses and shapes [Anguelov et al., 2005]. Although used later on to reconstruct humans from a single image [Balan et al., 2007], it was incompatible with modern game engines.

SMPL model

More recently, the SMPL model [Loper et al., 2015] has been proposed which solved the problems previous models had, such as compatibility with graphics engines, full disentanglement of pose and shape, easy and realistic mesh animation via a small set of pose and shape parameters. Next, SMPL+H puts together SMPL and MANO hand model [Romero et al., 2017] to extend SMPL with full hand articulation. This brings us to its latest variant, SMPL-X which can capture the full human body pose (hands, face expression and hand articulation) [Pavlakos et al., 2019]. The family of SMPL models has been the community's standard way to represent human bodies across different applications such as human body motion [Petrovich et al., 2022; Tevet et al., 2023], pose, shape and motion reconstruction from video [Kocabas et al., 2020], humans under

clothing. Although different models were proposed after it [Xu et al., 2020; Osman et al., 2020], it remains the most widely used across researchers. Since all the works in this thesis use the SMPL model to capture and represent the human motion, some terms that are shared among the works in Chapters 3, 4, 5 and Annex A.

The SMPL (Skinned Multi-Person Linear) model is a parametric 3D model for representing human bodies. It uses a skeleton with 24 joints to control a surface mesh consisting of 6,890 vertices. The model separates human body representation into two components: shape and pose. Shape variations are captured by a set of shape parameters β, which define the body shape. The pose parameters θ describe joint rotations in the body, Those parameters are used to create the final mesh. First, the shape is formed by the betas and the template mesh. Then, the pose parameters are used to pose the skeleton appropriately. Finally, SMPL uses LBS to smoothly deform the body [Kavan et al., 2008]. The final mesh vertices are calculated based on the posed mesh and the learned skinning weights. This approach allows the SMPL model to generate realistic body movements by controlling shape and pose independently, making it useful for applications such as motion capture and animation. More recently, some issues such as the lack of sparsity in the pose correctives — which leads to spurious correlations between vertices and their association with the rotations of joints — or feet articulation has been addressed in follow-up works such STAR [Osman et al., 2020] or SUPR [Osman et al., 2022].

In this thesis, we use SMPL compatible rotations and we choose to set the shape parameters to be zero. We do that since our main focus is the motion control from text, and assuming a single shape lets us simplify the problem.

Features used to represent 3D motions

Similarly to human body representations, there are multiple ways prior work represents human motion. This is crucial and still an open reasearch question when training networks to generate 3D motion. Different types

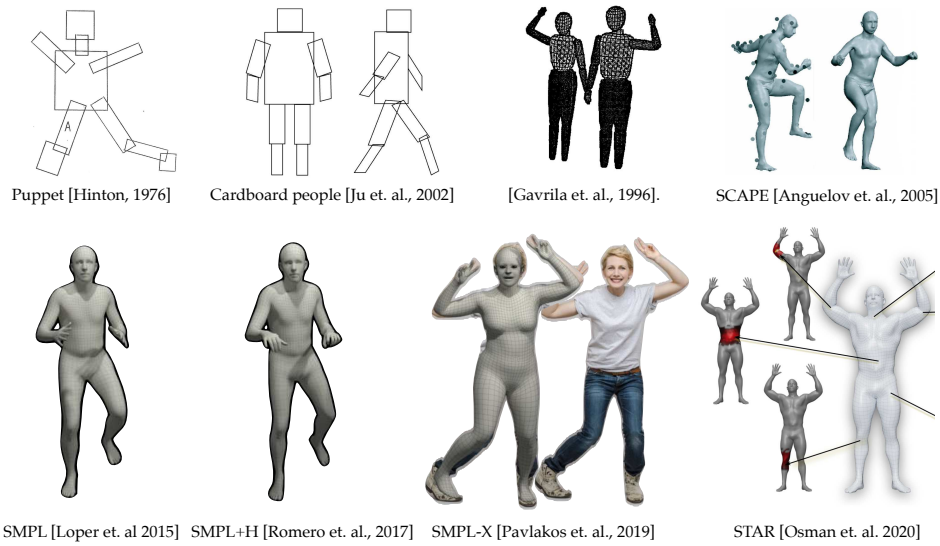


FIGURE 2.1: **Human body representations:** A brief history of the different representations of the human body. Starting from cardboard shapes [Hinton, 1976] and ellipsoids [Gavrila and Davis, 1996], moving towards more realistic representations using surfaces that are not compatible with 3D engines [Anguelov et al., 2005]. Recently, the vast majority represents realistic human bodies using the family of SMPL models [Loper et al., 2015; Romero et al., 2017; Pavlakos et al., 2019]. The most recent alternatives, such as STAR [Osman et al., 2020], build on and improve the surface representation using sparse pose correctives.

of rotation representations and features have been proposed [Holden et al., 2016; Guo et al., 2022a; Petrovich et al., 2022]. Mostly, the human body is represented by a skeleton, in which rotations are relative to the root joint following the kinematic tree. Such rotations can be SMPL rotations, or based on other skeletons such as MMM [Plappert et al., 2016]. Some prior work also represents rotations using only positional and angular velocities [Diomataris et al., 2024]. However, those representations mainly work for autoregressive models and require careful training. This happens as small error can accumulate easily and have significant artifacts after integration to get the positions from the velocities. For that reason, mainly 6D rotations [Zhou et al., 2019] have been widely adopted, which are suitable when training NNs due to their continuous nature over conventional rotations. 6D rotations avoid the discontinuities typically associated with 3D rotation formats, like Euler angles or quaternions. Holden et al. [2016] propose a pre-processing strategy which is widely adopted in recent literature, and commonly referred to as motion canonicalization. Their work greatly influenced how motion is represented to be fed as input to a NN. In detail, for each motion in the dataset, they identify the frontal direction by the cross product between the shoulder-to-shoulder and hip-to-hip vector. Then, they orient each motion to face the same initial direction. Their method also uses velocities for the root translation and represents joint positions in the local coordinate frame and call this representation RIFKE, which stands for Rotation-Invariant Forward Kinematics. This representation represents translation and global orientation with invariant features, essentially putting the different motions in the data in the same coordinate space. RIFKE has been used by the vast majority of works using HumanML3D dataset [Guo et al., 2022a] as a benchmark, and has also inspired representations which use joint rotations instead of positions. However, it has been applied to joint positions which cannot directly recover the 3D human body, and IK is being used as a post-processing step. Regarding the body's translation, most prior work represents it as velocities to induce the desired translation invariance in the input motion representation [Holden et al., 2016]. Additional features are often used to enforce smoother generations and

eliminate uncanny effects such as foot sliding. Such features comprise joint positions (either local or global) [Athanasiou et al., 2024; Holden et al., 2016], contacts [Guo et al., 2022a; Holden et al., 2016], separate components for the global orientation — e.g., separation to gravity and directional components [Petrovich et al., 2024].

In our work, we always use SMPL compatible representations, which allow for game engine compatibility and eliminate the need of optimizing the skeleton to fit to a body surface or any post-processing steps. Concurrent work mainly used HumanML3D motion representation, [Tevet et al., 2023; Guo et al., 2022b; Guo et al., 2022a; Zhang et al., 2023b] which requires optimization based on the predicted 3D joints using SMPLify [Bogo et al., 2016] to recover the 3D bodies. This process is prone to optimization artifacts and is significantly slower than regressing SMPL rotations directly.

In Chapters 3 and 4, we use a representation similar to RIFKE. We canonicalize the motion similar to RIFKE, representing the root translation with velocities and the SMPL body rotations using 6D rotations [Zhou et al., 2019]. In Chapter 5, the motion representation is inspired from STMC [Petrovich et al., 2024]. Motion is represented using SMPL pose parameters that are encoded using 6D rotations, and the gravity and xy orientations are disentangled. In addition to these features, we use local joint positions to ensure motions are smooth. A detailed outline of the representation can be found in Section 5.4.1 of Chapter 5.

Finally, it is common for motion features to be normalized using the mean and variance from the training set, to facilitate NN learning.

2.2 3D human motion generation

The field of 3D human motion generation focuses on creating and predicting realistic human movements from some initial motion. Early approaches use statistical models to predict motion sequences or rely on techniques like IK to ensure physically plausible movements [Sidenbladh et al., 2000]. Along with different fields of computer vision, NNs became popular in motion generation. Unconditional motion generation using

NNs, focuses initially on motion prediction [Martinez et al., 2017], and extends to generation using joint positions [Yuan and Kitani, 2020] or points on the surface of the SMPL model, as in MOJO [Zhang et al., 2021b]. Different tasks were established as the field was progressing, such as filling the "gaps" between motions (motion in-filling) [Harvey et al., 2020], conditioning on action labels [Guo et al., 2020], environmental constraints [Starke et al., 2019], or even music and speech [Kucherenko et al., 2023; Pu and Shan, 2022]. More recently, free-form text descriptions have become a prominent control signal [Ghosh et al., 2021; Petrovich et al., 2022; Guo et al., 2022a]. In addition to generating single motions, follow-up works explore how to combine multiple actions over time (temporal compositionality) or execute simultaneous movements (spatial compositionality), detailed in Section 2.2.4. Motion editing techniques have also emerged, allowing users to modify specific parts of a motion, like adjusting arm movements, while keeping the rest of the body intact, detailed in Section 2.2.5. This section provides an overview of these related works, laying the groundwork for understanding the advancements and applications in 3D human motion generation.

2.2.1 Unconditional motion generation & prediction

Motion prediction. Early work relies on statistical models (e.g., PCA) paired with Markov chains to predict future frames of human motion, conditioned on past motion [Bowden, 2000; Galata et al., 2001], or synthesize cyclic motions and locomotion using Gaussian processes [Urtasun et al., 2007] and PCA in the Fourier domain of the joint trajectories [Ormoneit et al., 2005]. More recent work relies on RNN models [Fragkiadaki et al., 2015; Gopalakrishnan et al., 2019; Pavllo et al., 2020], which can process variable sequence lengths. At the same time, other methods generate entire sequences at once through convolution [Holden et al., 2016; Yan et al., 2019], being more computationally efficient but less controllable. The majority of the aforementioned approaches are deterministic, while human motion is inherently stochastic and diverse. To address this, probabilistic models

have been proposed [Barsoum et al., 2017; Zhang et al., 2021b; Yuan and Kitani, 2020; Zhao et al., 2020], and approaches like DLow focus on increasing sample diversity [Yuan and Kitani, 2020].

Motion in-filling. An alternative to motion prediction is to generate individual actions or poses and then fill in the transitions between them, sometimes referred to as in-betweening. This is useful for animators as it is often needed to generate a motion which accurately transitions through different pre-specified poses. Early work on motion in-betweening relied on IK and space-time constraints to ensure physically plausible motion [Rose et al., 1996; Witkin and Kass, 1988]. Recent approaches have learned more expressive transitions from data [Duan et al., 2021; Harvey and Pal, 2018; Harvey et al., 2020; Hernandez et al., 2019; Zhang and Panne, 2018; Zhou et al., 2020]. Recurrent Transition Networks [Harvey and Pal, 2018] were proposed for fixed-length transitions, and were later extended into a stochastic model with a benchmark, based on LaFan1 dataset, for motion in-betweening [Harvey et al., 2020]. Convolutional approaches [Zhou et al., 2020] also demonstrate success by interpolating over long periods.

2.2.2 Motion generation conditioned on different modalities

Conditioned on objects & environment Several methods generate motion conditioned on scene constraints like the environment and objects [Starke et al., 2019; Hassan et al., 2021]. MotionVAE [Ling et al., 2020] used a VAE and deep reinforcement learning for goal-directed motion to navigate a character through mazes. For example, Zhang and Panne [2018] explored a mixture-of-experts model for dynamically computing architecture weights. Similarly, Starke et al. [2019] used a mixture-of-experts NN model to handle scene constraints and object interactions. Recent work on human-object interaction synthesis explores interactions with static and dynamic objects. For static objects, many approaches have explored basic actions in scenes with static objects such as navigating, sitting and lying on a couch [Hassan et al., 2021; Hassan

et al., 2019; Wang et al., 2022d; Zhang et al., 2022]. These methods include regression models [Mir et al., 2024; Araújo et al., 2023], diffusion-based models [Wang et al., 2024b], and reinforcement learning techniques [Xiao et al., 2023], enabling actions like sitting or lying down in context-aware ways. Recently, there has been increasing attention on synthesizing interactions with dynamic objects [Diller and Dai, 2024; Peng et al., 2023]. OMOMO [Li et al., 2023a] introduced a dataset of interaction with dynamic objects and proposed a framework to generate human motion from object motion. CHOIS [Li et al., 2024b] synthesizes actions in scenes from text and waypoints. Recently, Wu et al. [2024] synthesize coordinated full-body, object, and hand motions. We briefly mention these works here for completeness, but this thesis does not focus on generation in scenes or using objects. However, LLM-based planning and putting agents in real-world environments is an important future step.

Music-conditioned motion generation. Another popular condition for generating motion is music. Li et al. [2021] proposed a method for music-conditioned motion generation and introduced a dataset, AIST++, based on SMPL fitting on Mixamo motion sequences. Moltisanti et al. [2022] focused on generating more expressive dance sequences which are synchronized to different music styles. Music-based generation often relies on the temporal structure of both music and motion to produce synchronized movements. This synchronization was explicitly enforced on follow-up works based on acoustic and music features has been used in [Alexanderson et al., 2023], using bigger datasets [Valle-Pérez et al., 2021]. Concurrent work, used diffusion models in a sequence [Tseng et al., 2023] or autoregressive manner [Zhang et al., 2024].

Audio-conditioned motion generation. Audio and human speech has been another prominent control. Audio-conditioned motion generation focuses mainly on communicating speech cues with gestures. Talking-with-Hands dataset [Lee et al., 2019] has been a first step to enable this by providing a large-scale dataset with skeletons paired with conversational data. Yazdian et al. [2022] used machine translation to

map text to gesture chunks. Ao et al. [2023] uses text features, while diffusion-based models have been explored to generate mainly upper body motions from speech [Yang et al., 2023; Chhatre et al., 2024]. More recent works aim for detailed SMPL-X motions, that include hands and full-body gestures. Yi et al. [2023] introduced a new dataset, TalkSHOW with SMPL-X fits of TV footage, along with a VQ-VAE model that generates upper body face expressions and hand motions. BEAT Liu et al., 2022 introduced a new benchmark on gesture generation for full bodies. Its follow-up work, EMAGE [Liu et al., 2024] introduced the BEAT2 dataset and an approach to generate full-body SMPL-X motions with global motion that are not only focused on upper bodies.

2.2.3 Text-driven motion generation (single text)

Motion generation conditioned on action labels. Action2Motion [Guo et al., 2020] and ACTOR [Petrovich et al., 2021] are two recent approaches that address the problem of motion generation conditioned on action labels. Action2Motion uses a GRU-VAE architecture that operates on a per-frame basis, while ACTOR [Petrovich et al., 2021] employs a transformer-VAE model to encode and decode motion sequences in a single step. These methods focus on using a predefined set of action labels, while newer approaches focus on free-form language inputs. Such methods were the initial approaches which then led to the extension from action categories to free-form language.

Language-based human motion generation. Language-based motion generation extends the control capabilities of motion generation systems by using free-form language input. For example, Text2Action [Ahn et al., 2018] relies on an encoder-decoder RNN to learn the mapping between language and pose. Initial efforts include Language2Pose [Ahuja and Morency, 2019], which learns a joint embedding space for language and poses, and Ghosh et al. [2021], which generates 3D skeletons from textual input. Recently, ACTOR [Petrovich et al., 2021] and TEMOS [Petrovich et al., 2022] have demonstrated progress in handling free-form language,

building on transformer and VAE models to enable generation from a broader range of text inputs. Following the success of diffusion models [Ho et al., 2020] in other domains, several works have recently applied them to text-conditioned motion generation. MDM [Tevet et al., 2023], FLAME [Kim et al., 2022a], and MotionDiffuse [Zhang et al., 2023b] demonstrate the capabilities of diffusion-based models for motion synthesis, achieving state-of-the-art results on text-conditioned motion generation tasks. Diffusion models have been shown to handle finegrained local control, such as joint trajectories and keyframes [Huang et al., 2024; Karunratanakul et al., 2023] and offer significant improvements in generating diverse and realistic motions from textual input.

2.2.4 Finegrained text control (Spatial; Temporal)

Compositionality in motion generation. Human motion is a composition in space and time —with actions happening one after another and different body parts performing different actions. Most research focuses on temporal compositionality, such as sequencing actions over time [Wang et al., 2022a]. Early work by Arikan et al. [2003] used dynamic programming to compose motion sequences from action labels in a motion database. Recent approaches such as MultiAct [Lee et al., 2022] aim to produce smooth transitions between generated actions.

Temporal compositionality in motion generation. Some recent works address motion generation beyond a single textual instruction, focusing on temporal compositionality. These approaches aim to generate a sequence of motions from a series of textual inputs. For instance, TEACH [Athanasiou et al., 2022], presented in Chapter 3 builds on TEMOS [Petrovich et al., 2022] by incorporating an action-level recursive design that generates the next action conditioned on past motion. MultiAct [Lee et al., 2022] similarly targets the generation of continuous transitions between actions, producing temporally coherent motion sequences from text. PriorMDM [Shafir et al., 2024] builds on diffusion models and extends them by noising and denoising the transition

space between actions to produce smooth actions. More recently, DiffCollage [Zhang et al., 2023d] also builds on the diffusion model and generates smooth action compositions from text via parallel generation of the motions and their transitions first, while denoising and merging the results later.

Spatial compositionality (simultaneous actions). While temporal compositionality focuses on sequences of actions, spatial deals with simultaneous actions described in a single text instruction. Recent work, such as MotionCLIP [Tevet et al., 2022] and MDM [Tevet et al., 2023], has explored the compositional capabilities of their methods, testing their ability to generate simultaneous actions in response to single textual inputs. MotionDiffuse [Zhang et al., 2023b] further extends this by injecting manually labeled body-part information to achieve spatial compositionality, enabling more complex and realistic simultaneous motions from textual descriptions. Concurrent to these works, SINC, presented in Chapter 4, exploits the knowledge of LLMs about human body motion and automatically creates synthetic data for simultaneous actions. This extends beyond the training data, which is limited, and allows for the compositions of actions from single-action models. In addition, a trained model on this synthetic data shows superior performance for generating simultaneous actions.

2.2.5 Motion editing

Part-based & trajectory-based motion editing. Recent approaches to motion editing have focused on controlling individual parts of the body during the editing process. MDM [Tevet et al., 2023] shows a part-based editing application of diffusion models, allowing the user to adjust specific body regions, such as the upper or lower body, based on textual descriptions. Similarly, MotionDiffuse [Zhang et al., 2023b] enables finegrained control over body parts through text-based conditioning, selectively editing a body motion while keeping other parts unchanged. This approach is particularly useful for tasks like refining

motion, retargeting or creating new variations of a motion sequence based on user-specified instructions. Similarly, FLAME [Kim et al., 2022b] allows editing of body-part-specific motion through diffusion-based methods similarly. Diffusion-based models were shown to be suitable for finegrained control using either joint trajectories[Karunratanakul et al., 2023; Xie et al., 2024] or keyframes [Cohan et al., 2024] as conditions. These methods allow part-based editing, which is useful in tasks like retargeting or adjusting only specific portions of a motion sequence. However, they are limited to a specific edit type based only on coordinates and relying on manual body-part selection.

Style-based motion editing. Style-based editing or motion style transfer exploits datasets that contain a small set of style labels such as 'angry' and 'old' using clips of 2D or 3D data [Aberman et al., 2020]. This line of work, focuses mostly on copying the style of one motion onto another, typically performing the same action. Holden et al. [2017] with Phase-Functioned NNs (PFNN), use a one-hot style representation to learn a few motion styles, but their approach requires the total number of styles to be known before training. Learned motion matching [Holden et al., 2020] can efficiently model stylized motion but learns no explicit style representation, rather using a general feature matching approach that increases in cost with the dataset size. Mason et al. [2022] introduce a dataset 100STYLES, shown in Figure 2.3, which contains 100 styles of some cyclic motion such as walking, sidestepping and running. They also introduce a model based on PFNN that mixes such styles. However, all these works mostly focus on a small set of actions and styles.

Motion editing from language instructions. There are several recent approaches to editing human motion with different levels of control. However, all such works rely on editing specific manually selected body parts or focus on specific types of edits. CoMo [Huang et al., 2024] and FineMoGen [Zhang et al., 2023c] use large language models (LLMs) to generate motion edits based on natural language descriptions, enabling finegrained control over individual body parts. Among *heuristic*-based

approaches [Fieraru et al., 2021; Goel et al., 2024], AIFit [Fieraru et al., 2021] can edit domain-specific exercise poses from a pre-defined grammar and is focused on a limited set of cyclic motions from their Fit3D dataset. Iterative Motion Editing [Goel et al., 2024] relies on captioned source motions, which are passed through an LLM along with a pre-defined set of 'Motion Editing Operators' (MEOs), to detect which joints and frames should be edited. A pre-trained diffusion model is then used to infill these locations.

In contrast, our work in Chapter 5 does not focus on a specific type of motion editing or any heuristics. Our MotionFix contains diverse edits and our model TMED can generate edited animation based on free-form language and an initial animation to which we refer to as source motion.

2.3 Evaluation of generative motion models

Evaluating generative models poses a significant challenge due to the complex and often subjective nature of the generated outputs (e.g., images, text, or motion). Comparing the generated results with the groundtruth samples is often not enough. The purpose of generative models is not to memorize a particular image or motion but to be able to generate realistic data points, either unconditionally or given a condition such as text [Chen et al., 2024b], class or any other type (e.g., music, speech [Kucherenko et al., 2023] etc.), within the data distribution. Especially when text is supposed to guide the generation, there are multiple "correct" answers and measuring error against the ground truth is limiting. It is an open research question how to evaluate conditional generative models and, in most cases, different metrics are used to show different aspects of the results.

In text-conditioned motion generation, there are similar challenges. Imagine, for example, someone walking in a circle. There are multiple correct ways to perform this action; e.g., in a narrower or broader circle, starting with the left or right foot, with faster/slower or bigger/smaller steps. This challenge becomes more apparent as the text control becomes more finegrained. The existing text annotations are coarse, and text is not

the most suitable finegrained control for motion, as it cannot precisely and exhaustively describe it. There are multiple approaches to evaluate such conditional or unconditional generative models. Further evaluation methods used today for text-conditioned motion generation methods can be broadly categorized into four main approaches: coordinate-based metrics, feature-based metrics, human studies, and retrieval-based scores. Each method comes with its set of trade-offs, and it is common to use multiple metrics when evaluating a conditional motion model to pinpoint each model's strengths and weaknesses. Below is an overview of these approaches and the corresponding difficulties.

2.3.1 Coordinate-based metrics

Initial unconditional motion generation methods [Zhang et al., 2021b] and motion prediction methods [Yuan and Kitani, 2020] use coordinate-based metrics to evaluate their accuracy and diversity compared against the groundtruth. All of these initial metrics focus on how the global joint positions resemble the ones in the groundtruth motions. Joint positions are commonly chosen as the input to these metrics, as they provide a significantly better representation which aligns well with human judgment.

A first such a set of metrics is used in motion generation tasks. The **Average Pairwise Distance (APD)** measures diversity by evaluating how different the motion samples are from each other [Aliakbarian et al., 2020]. The **Average Displacement Error (ADE)** evaluates accuracy by measuring how close the generated motions are to the ground truth overall time steps. The **Final Displacement Error (FDE)** assesses accuracy by focusing on the closeness of the final poses of the generated and ground truth motions [Alahi et al., 2016]. The **Multi-Modal ADE (MMADE)** extends ADE to account for multiple possible ground truth motions by grouping similar past motions, while the **Multi-Modal FDE (MMFDE)** extends FDE to multi-modal settings by considering multiple possible future motions. These metrics collectively assess a method's ability to generate diverse and accurate predictions, where APD focuses on diversity, and ADE,

FDE, MMADE, and MMFDE evaluate accuracy in single and multi-modal contexts [Yuan and Kitani, 2019]. Concurrently to those, the Normalized Power Spectrum Similarity (NPSS) [Gopalakrishnan et al., 2019] has been introduced. NPSS focuses on comparing the periodicity and distribution of joint angle variations in motion sequences, aligning more closely with human perception of motion quality. It is computed using the power spectrum in the frequency domain of two sequences using the Earth Mover's Distance (EMD).

Inspired by these metrics, the first coordinate-based metrics for text-conditioned motion generation were proposed. Language2Pose [Ahuja and Morency, 2019] introduce **Average Position Error (APE)**, which measures the accuracy of generated motions by calculating the average difference in joint positions between the generated and ground-truth poses. It is split into local joint positions and the global root trajectory to separately account for pose and motion direction accuracy. Later, Ghosh et al. [2021] introduce **Average Variance Error (AVE)**. AVE evaluates the variance consistency of joint positions between generated and ground-truth motions. This metric captures how well the variability of generated poses aligns with the distribution of ground-truth poses. These metrics are computed for the local and global joints, the global trajectory (x-y trajectory) and the full root joint translation (including the z-axis/gravity component). We use APE, AVE-based metrics in Chapters 3, 4 along with feature-based scores and additional metrics to provide sufficient evaluation for our models. Although such metrics are useful for generation and prediction tasks, they do not factor in the inherent diversity of human motion and the alignment with text descriptions. In addition, to the existing coordinate-based metrics, we introduce Transition Distance (TD) in Chapter 3 to measure the distance between the neighboring frames of the motion pairs which were to be temporally composed. Our metric addresses the lack of quantitative evaluation measures for computing the transition error between motions.

2.3.2 Feature-based metrics

Metrics like FID [Heusel et al., 2017] and IS [Salimans et al., 2016] are popular tools for quantifying the quality and diversity of generative models' outputs in the image generation domain. These methods typically compare the statistical properties of real and generated data using large pre-trained models (e.g., InceptionV3 for image data). Similarly, in the text-to-motion domain Ghosh et al. [2021] proposed **Content Encoding Error (CEE)**, and **Style Encoding Error (SEE)**. CEE assesses the alignment of generated motions with input textual descriptions by comparing the latent embeddings of poses and sentences in a shared representation space. It evaluates the semantic consistency between input and output. SEE evaluates the consistency of style between generated motions and their corresponding textual descriptions by comparing the Gram matrices of sentence and pose embeddings. This metric captures the overall stylistic alignment of motions.

Such evaluation of generative models faces several challenges. One significant issue is the bias from pre-trained models, as these evaluation scores depend heavily on pre-trained networks and feature extractors that may not fully capture the specific nuances of generated motions, particularly when the domain diverges from the model's training data and might focus on common data classes. Another challenge is the inability to capture finegrained qualities; metrics like FID and IS aim to show whether the feature distributions match. However, they have little to do with how much the motions align with a text. Lastly, there is the lack of interpretability in these scores, making them difficult to use in a practical sense and making it difficult to assess improvements in the generated motions. Hence, they are useful, but they have to be used in addition to other metrics. We also introduce TEMOS Score, in Chapter 4, to better measure the alignment of the text with the generated motion, and we show how it solves some problems with the coordinate-based metrics. To alleviate their biases, training classifiers and contrastive models, of text and motion, need to grow bigger. Furthermore, instead of focusing on matching distributions in isolation, evaluation methods should focus on

how the one modality (text) can retrieve the other one (motion) and vice versa.

2.3.3 Human evaluation

Perceptual studies, where human participants evaluate the quality of generated content, are essential for assessing generative models. These studies often require participants to rate or rank outputs based on attributes such as realism, quality, diversity, or alignment with specific goals. Human evaluation is considered the "gold standard" because it captures subjective qualities that automated metrics cannot. However, these studies face several challenges. Subjectivity in ratings arises due to individual biases, preferences, and cultural differences, making consistent conclusions difficult. Cost and scalability are significant barriers, as recruiting participants, designing experiments, and analyzing results require considerable resources, especially for large-scale studies. Inconsistency and reproducibility also pose challenges, as participants may interpret evaluation criteria differently or apply them inconsistently. This complicates the replication of results.

Finally, training and standardization of evaluators is a complex task, requiring significant effort to ensure consistent assessments across participants. Strategies to address these issues include providing clear guidelines, using diverse participant pools, leveraging pairwise comparisons to reduce cognitive load, employing crowdsourcing platforms to scale evaluations cost-effectively, and applying advanced statistical methods to analyze results. Despite the challenges, perceptual studies are indispensable, as they ensure that generative models align with human expectations and produce outputs suitable for real-world applications. Perceptual studies have been used in Chapter 5 to compare TMED (our motion editing model) against the baselines. Details about our efforts to mitigate subjectivity and ensure the credibility of the results can be found in Section 5.5.1 of Chapter 5.

2.3.4 Retrieval-based metrics

One evaluation approach that has gained popularity is retrieval-based methods. In retrieval-based approaches, a classifier or encoder is trained on real data, and retrieval metrics (such as precision, recall, and accuracy) paired with a distance measure, are used to evaluate how well the generated data aligns with real-world samples. This method is often used when assessing how closely the generated data resembles true data distributions or how well can a text retrieve the target motion.

The evaluation using retrieval-based scores is widely used for cross-modal evaluation. One major issue is the dependence on classifier quality, as the accuracy of these metrics relies heavily on the quality of the classifier used to evaluate the generated samples, potentially introducing bias. Additionally, diversity is often undervalued by these metrics, which tend to favor similarity to training data, thereby overlooking the model's ability to produce novel, diverse, or creative outputs. Finally, there is the challenge of limited generalization; retrieval-based metrics are effective for tasks involving the reproduction of known classes but struggle with open-ended generation tasks that require novelty.

Regardless of these challenges, they have been proven empirically that they can provide valuable insights into the performance of generative models, especially when combined with other evaluation methods [Voas et al., 2023]. A first such evaluation benchmark has been proposed as a part of the HumanML3D dataset [Guo et al., 2022a]. The text and motion feature extractor were trained in a contrastive manner. The text encoder maps raw text into feature vectors. Motion sequences are processed into motion features using a bidirectional GRU. The contrastive loss aligns matched text-motion pairs while ensuring mismatched pairs are separated. Then, recall scores are computed based on these motion and text features. Each generated motion feature is compared against a batch of 32 elements of text features. Those 32 elements contain the actual text and 31 randomly sampled texts from the the test set. This process is repeated for the groundtruth and the generated motions. Such scores evaluate the ranking quality of generated motions when compared to

their corresponding textual inputs. High R-precision indicates that the model generates motions closely aligned with the provided text. Such metrics have been widely adopted in the literature and later improved based on better contrastive methods, such as TMR [Petrovich et al., 2023], to compare text-motion alignment [Petrovich et al., 2024; Lin et al., 2023].

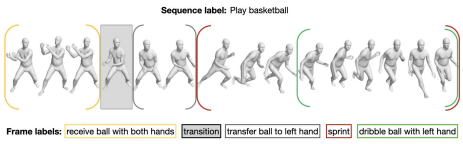
In Chapter 5, we introduce several motion-to-motion retrieval scores to evaluate our model on the newly introduced dataset MotionFix. We introduce two motion-to-motion retrieval scores to measure how well the generated motions follow the instructions and resemble the initial motions given for editing.

Evaluating generative models is inherently challenging due to the subjectivity of outputs and the limitations of current evaluation methods. Coordinate-based scores ignore the text alignment and focus on matching specific motion patterns. Hence, they are rarely used on their own as they poorly measure diversity, and capture very little about the text-motion alignment. Feature-based scores (e.g., FID) are limited by pre-trained models and their lack of finegrained assessment capabilities. Human evaluation, though often seen as the most reliable, is expensive, subjective, and inconsistent. Retrieval-based metrics are more automated but struggle with novelty and diversity, focusing instead on matching the generated data with known distributions. As a result, a combination of these methods is often used to achieve a more comprehensive evaluation of generative models.

2.4 Motion & language datasets

2.4.1 Text-to-motion datasets

Recently, datasets related to human motion have become critical for advancing fields like computer vision, robotics, and motion generation. These datasets enable a wide range of applications such as human pose estimation, action recognition, and multimodal learning. Below is a brief summary of several notable datasets, detailing their size, structure,

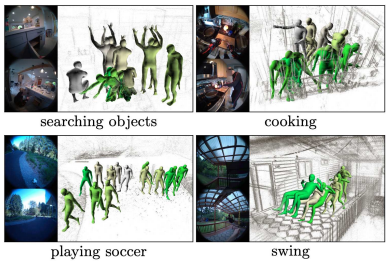


BABEL [Punnakal et. al, 2021]

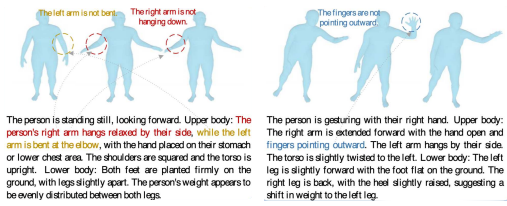
HumanML3D [Guo et. al, 2022]

KIT Motion-Language [Plappert et. al, 2016]

Motion-X [Guo et. al, 2023]



Nymeria [Ma et. al, 2024]



MotionBase [Wang et. al, 2024]

FIGURE 2.2: **Text-to-motion datasets:** Different motion datasets have been proposed in the literature that contain different types of labels and ways to represent humans. BABEL [Punnakkal et al., 2021] contains finegrained description for all frames in a motion sequence, HumanML3D [Guo et al., 2022a] contains multiple annotations for the same motions, while KIT [Plappert et al., 2016] was the first and smaller such dataset containing some limited everyday actions. The first two datasets annotate parts of the AMASS dataset. More recently, datasets like Motion-X [Lin et al., 2023] and MotionBase [Wang et al., 2024a] exploit in-the-wild internet videos and reconstruct SMPL-X bodies from them. Finally, Nymeria [Ma et al., 2024], contains a much more rich feature set (depth maps, egocentric videos, scene & gaze information) about people doing everyday actions.

and specific features, see Figure 2.2. These datasets contain action labels, free-form language, or both, paired with 3D motion data (mostly SMPL-based) and, optionally, corresponding RGB videos. Our criterion to including a dataset in our analysis is the existence of 3D motion data with some text labels.

The Human3.6M (H3.6M) [Ionescu et al., 2014] dataset is a large-scale collection of 3D human poses, capturing over 3.6 million poses using a motion capture system. It includes 11 actors performing various daily activities (e.g., "walking," "sitting,", "phone conversations") from multiple viewpoints. The dataset contains synchronized 2D/3D data, depth information, making it valuable for tasks like human pose estimation, action recognition, and modeling. It has mainly been used for pose estimation and motion prediction, given its limited action categories and unavailability of SMPL-based groundtruth motion. The NTU RGB+D dataset [Shahroudy et al., 2016] is a large-scale dataset for 3D human activity analysis. It contains over 56,000 video samples (approximately 120 hours of footage) collected from 40 different subjects performing 60 different actions. These actions are divided into three categories: 40 daily actions (e.g., "drinking," "reading"), 9 health-related actions (e.g., "falling down," "staggering"), and 11 mutual actions (e.g., "punching," "hugging"). Data is collected from multiple camera views and includes RGB, depth maps, 3D skeletal data, and infrared frames. A follow-up of this work resulted in a later version of the dataset NTU RGB+D 120 [Liu et al., 2020]. This large-scale benchmark for 3D human activity recognition, containing over 114,000 video samples (approximately 120 hours) captured from 106 subjects performing 120 distinct action classes. These actions are categorized into three groups: 82 daily activities, 12 health-related actions, and 26 mutual activities. Data modalities include RGB, depth, 3D skeletal data, and infrared frames, captured from 155 different camera views, providing rich variation in terms of subjects, environments, and viewpoints. It has been mainly used for action-based motion generation [Petrovich et al., 2021; Guo et al., 2020], and action recognition as more recent datasets annotating AMASS [Mahmood et al., 2019] rose in popularity. The HumanAct12 dataset [Guo et al.,

2020] is a 3D human motion dataset, comprising 1,191 motion clips and over 90,000 frames. It is organized into 12 high-level action categories, including activities like "warm-up" and "lifting dumbbells," and further divided into 34 subcategories. The dataset includes 24 body joints and offers high-quality, accurate 3D pose annotations, making it suitable for tasks like action recognition and 3D motion generation [Guo et al., 2020; Petrovich et al., 2021]. The BABEL dataset [Punnakkal et al., 2021] provides dense action annotations for over 43.5 hours of motion capture (MoCap) data from the AMASS dataset. It includes 9,421 unique language labels organized into 252 action categories, covering everyday actions like "walking," "dancing," "playing instruments," and more. A total of 66,289 action segments are annotated, averaging 5.01 segments per sequence. BABEL's frame-level labeling makes it a valuable resource for tasks like action recognition, motion synthesis, and temporal localization in human movement data. For further details on BABEL, please refer to the Annex A. BABEL was the first dataset with action annotations of AMASS. Concurrently, the HumanML3D dataset [Guo et al., 2022a] is one of the largest collections of 3D human motions paired with text annotations. It includes 14,616 motion sequences with 44,970 corresponding textual descriptions, comprising over 28.59 hours of motion data. These descriptions span a wide range of action types, such as "daily activities," "sports," and "artistic movements," with each motion sequence annotated by at least three distinct text descriptions. Given its free-form text description, although not as detailed as in BABEL, it became prominent and widely used for the task of text-to-motion generation. The LaFAN1 dataset [Harvey et al., 2020] is a high-quality motion capture dataset designed for motion prediction and transition generation. It includes 496 motion sequences from 5 actors, totaling over 60 minutes of motion data. The dataset focuses on realistic transitions between keyframes and provides benchmarks for evaluating in-betweening tasks. LaFAN1 includes human locomotion, diverse actions, and cyclic motions, making it particularly suitable for research in motion synthesis, especially in tasks requiring robust motion transitions.

Recently, datasets attempting to exploit Internet videos and capture

data in in-the-wild scenarios have been proposed. The Motion-X dataset [Lin et al., 2023] is a large-scale 3D human motion dataset containing 81,084 motion sequences, spanning 15.6 million frames, and annotated with both sequence-level and frame-level text descriptions. It captures expressive whole-body motions, including body, facial expressions, and hand gestures. The dataset covers both indoor and outdoor scenes, featuring diverse activities such as "daily life," "sports," and "professional performances." Motion-X aims to enhance research in 3D whole-body motion generation and human mesh recovery. The Nymeria dataset [Ma et al., 2024] is a large-scale collection of multimodal egocentric human motion data, capturing 300 hours of daily activities across 1,200 recordings from 264 participants in 50 locations. It includes 260 million body poses, 201 million images, 11.7 billion IMU samples, and 10.8 million gaze points. The dataset spans a diverse range of indoor and outdoor activities and is annotated with 310.5K sentences in 8.64 million words, providing a rich resource for research in egocentric human-object interactions, AR, and human motion analysis. Wang et al. [2024a] introduces MotionBase, a large-scale motion generation benchmark designed to advance research in motion generation. MotionBase contains over 80,000 motion sequences, covering approximately 300 hours of motion data. The dataset also features a vocabulary of over 1,000 natural language instructions, making it one of the most comprehensive motion datasets to date. This study presents techniques such as a 2D lookup-free motion tokenization method that improves representation capacity and explores the use of synthetic data and pseudo labels for training. The authors emphasize that scaling both data and models results in significant improvements, especially for generating unseen motions. Additionally, they note limitations in current evaluation metrics, particularly for out-of-domain instructions.

An illustration of all these datasets can be seen in Figure 2.2, from which we can observe the different types of annotations. For example, there are more or less detailed text annotations, e.g., per frame vs. per sequence (BABEL vs. HumanML3D), or different body models SMPL vs. SMPL-X bodies (initial works vs. Motion-X [Lin et al., 2023],

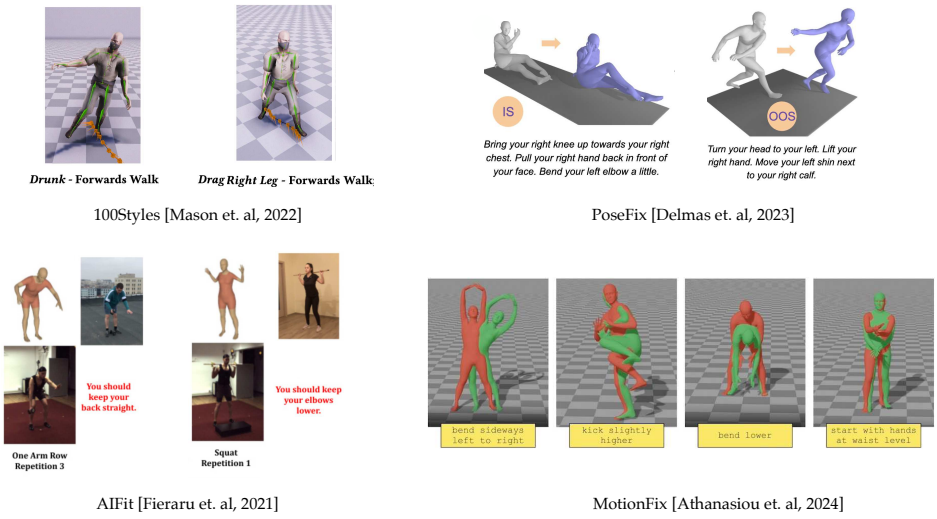


FIGURE 2.3: **Motion editing datasets:** We show some datasets from the literature such as 100Styles [Mason et al., 2022], which blends some basic actions such as walk and run with different styles, AIFit [Fieraru et al., 2021] contains exercise repetition performed by a trainer and a trainee containing annotation segmentation. PoseFix [Delmas et al., 2023] which is closer to the task we are addressing in this thesis, contains pairs of pose and edit texts. Finally, MotionFix (Chapter 5) contains about 6.7K pairs of motions along with edit texts, being the first dataset which contains motions and free-form language descriptions.

MotionBase [Wang et al., 2024a]), and some datasets include different modalities and scene information (Nymeria).

2.4.2 Motion editing datasets

There are numerous traditional methods for editing 3D humans to generate movies and small animations [Catmull, 1972], imposing space or time constraints [Cohen, 1992], interactively generating motions using procedural animation [Lee and Shin, 1999; Perlin, 1995] or physics-based approaches [Popović and Witkin, 1999]. Although these approaches attempt to solve this problem, very little amount of editing ground truth existed until recently. Recent work using *language* can be grouped into pose [Kim et al., 2021; Delmas et al., 2023] or motion [Fieraru et al., 2021; Goel et al., 2024] editing. In FixMyPose [Kim et al., 2022b], the

focus is on editing athletic human poses in synthetic images. In [Delmas et al., 2023], a text-based 3D human pose editing method is developed, enabled through the collection of the PoseFix dataset containing language descriptions of differences between pairs of poses. PoseFix builds on the previous work of PoseScript [Delmas et al., 2022], where a dataset of pose descriptions is automatically collected through a rule-based approach. The PoseFix dataset consists of 135,000 3D human pose pairs annotated with corresponding textual feedback that describes how to adjust the source pose to match the target pose. This dataset is designed for tasks such as text-based pose editing and correctional text generation. A subset of 6,157 pairs is annotated by human annotators, with the rest automatically generated. PoseFix focuses on finegrained pose correction in a wide range of human motions, providing a valuable resource for multi-modal learning. Although this dataset refers to static poses, it is the closest to our work for dynamic bodies and useful to be mentioned here.

In terms of dynamic bodies, current motion editing approaches mainly involve datasets that are *style*-based editing or motion style transfer exploits datasets that contain a limited number of style labels such as 'angry' and 'old' [Aberman et al., 2020] and even up to 100 different styles [Mason et al., 2022]. As mentioned in Section 2.2.5, most of the part-based editing considers selecting a subset of the body parts, manually [Tevet et al., 2023; Zhang et al., 2023b]. *Heuristic*-based approaches, such as AIFit [Fieraru et al., 2021] can edit domain-specific exercise poses from a pre-defined grammar and is focused on a limited set of cyclic motions from their Fit3D dataset. Fit3D contains exercises performed by a trainer and a trainee and repetition segmentation, as the exercises performed in the dataset are cyclic.

In contrast to prior works, MotionFix does not focus on a specific type of motion editing or any heuristics. Unlike PoseFix that concentrates on static poses, our MotionFix dataset involves dynamic motions where the space of possible edits are much larger, necessitating a different approach to data collection. Our MotionFix contains diverse edits, as can be seen in Figure 2.3.

Finally, we compare the scale of all the aforementioned datasets in Table 2.1. It is clear, that initially there were small datasets with very few action labels [Ionescu et al., 2014], which gradually increased but the groundtruth motions were noisy [Zou et al., 2022; Shahroudy et al., 2016]. The next generation of dataset [Guo et al., 2022a; Punnakkal et al., 2021], significantly improved the quality and amount and detail of labels, enabling free-from text-to-motion generation. Recently, datasets grew further including more expressive features of human-motion [Lin et al., 2023; Wang et al., 2024a] and different environment features [Ma et al., 2024] pushing their scale further. In parallel, the aim towards finegrained control in both pose [Delmas et al., 2023], and motion space (see Chapter 5) has brought a surge in popularity, stimulating further research on detailed motion captioning [Chen et al., 2024a; Ashutosh et al., 2025] and new motion editing datasets [Jiang et al., 2025].

Dataset	GT?	Act.	Hours	Vocab.	Motions	Label Type
Human3.6M [81]	✓	17	18	-	187	action labels
NTU RGB+D 60 [174]	✗	60	37	-	56,880	action labels
KIT-ML [152]	✓	-	-	1,263	3,911	free-from text
NTU RGB+D 120 [114]	✗	120	74	-	114,480	action labels
LaFan1 [65]	✓	12	4.6	-	77	action labels
HumanAct12 [63]	✗	12	6	-	1,191	action labels
BABEL [156]	✓	252	43.5	1,628	44,682	action labels, free-from text
HumanML3D [61]	✓	–	28.6	6,557	14,616	free-from text
PoseFix [39]	✓	-	-	1,068	-	free-from text, predefined free-from text
Motion-X [109]	✗	-	144.2	-	81,100	predefined free-from text, free-from text
Nymeria [118]	✓	-	300	6,545	311,000	free-from text, action labels
MotionFix [14]	✗		6.9 × 6.9	1,479	6,730 × 2	free-from text
MotionBase[1] [202]	✗	-	-	-	930,000	free-from text

TABLE 2.1: Existing datasets with language labels for human motion. `GT motion` indicates whether the human movements are accurate (MoCap) or noisy estimates (e.g., via tracking). `# Actions` indicates the total count of action categories in each dataset. `# Hours` indicates the total duration of all sequences in the dataset. Vocabulary size, # Motions, and Label Type columns have been added, with placeholders for the corresponding data. BABEL uniquely provides large-scale dense (per-frame) action labels for natural, continuous, ground-truth human movement data.

In this thesis, we introduce a method to create synthetic data, detailed in Chapter 4, and a text-based 3D motion editing dataset, MotionFix, detailed in Chapter 5. Finally, we describe BABEL — to which the author of this thesis contributed — in Annex A. Due to its frame-level annotations, BABEL serves as a benchmark for temporal and simultaneous action generation in the following two chapters.

[1]Estimated from the original preprint.

Chapter 3

Temporal Compositions of 3D Human Motions

Given a series of natural language descriptions, our task is to generate 3D human motions that correspond semantically to the text, and follow the temporal order of the instructions. In particular, our goal is to enable the synthesis of a series of actions, which we refer to as temporal action composition. The state of the art in text-conditioned motion synthesis only takes a single action or a single sentence as input. This is partially due to lack of suitable training data containing action sequences, but also due to the computational complexity of their non-autoregressive model formulation, which does not scale well to long sequences. In this Chapter, we address both issues. First, we exploit the recent BABEL motion-text collection, which has a wide range of labeled actions, many of which occur in a sequence with transitions between them. Next, we design a Transformer-based approach that operates non-autoregressively within an action, but autoregressively within the sequence of actions. This hierarchical formulation proves effective in our experiments when compared with multiple baselines. Our approach, called TEACH for "TEmporal Action Compositions for Human motions", produces realistic human motions for a wide variety of actions and temporal compositions from language descriptions. To encourage work on this new task, we make our code available for research purposes at `teach.is.tue.mpg.de`.

3.1 Introduction

The generation of realistic 3D human motions has applications in virtual reality, the games industry, and any applications that require motion capture data. Recently, controlling 3D human motion synthesis with semantics has received increasing attention [Guo et al., 2020; Petrovich et al., 2021; Ghosh et al., 2021; Petrovich et al., 2022]. The task concerns inputting semantics in the form of categorical actions, or free-form natural language descriptions, and outputting a series of 3D body poses. In this Chapter, we address the latter, i.e., text-conditioned motion generation, which is more flexible compared to pre-defining a set of categories. More specifically, our goal in this Chapter, is to animate a sequence of actions given a sequence of textual prompts. TEACH was the first to systematically explore generating such *temporal action compositions*. Our initial work stimulated further research in the area in terms of both modeling techniques (e.g., diffusion – DiffCollage, FlowMDM [Barquero et al., 2024; Zhang et al., 2023d] and transformers – MultiAct Lee et al., 2022) and new datasets such as Sequential Texts Described Motion (STDM) Li et al., 2023b.

Humans move in complex ways that involve different simultaneous and/or sequential actions. Hence, *compositionality* in time must be modeled to generate everyday motions that contain a series of different actions and that last longer than a few seconds. Compositionality in space, i.e., simultaneous actions, is another interesting direction, addressed in Chapter 4. Generative models are popular for synthesizing images conditioned on textual descriptions [Ramesh et al., 2021; Saharia et al., 2022]. Their success can be largely attributed to massive training datasets. The same breakthrough has not happened for 3D human motion generation due to lack of motion-text data. Standard benchmarks for the text-conditioned motion synthesis task (e.g., the KIT Motion-Language dataset [Plappert et al., 2016]) are limited in the vocabulary of actions and the number of motion sequences. In our experiments, we use textual annotations of the BABEL dataset [Punnakkal et al., 2021], providing English descriptions for the AMASS motion capture collection [Mahmood

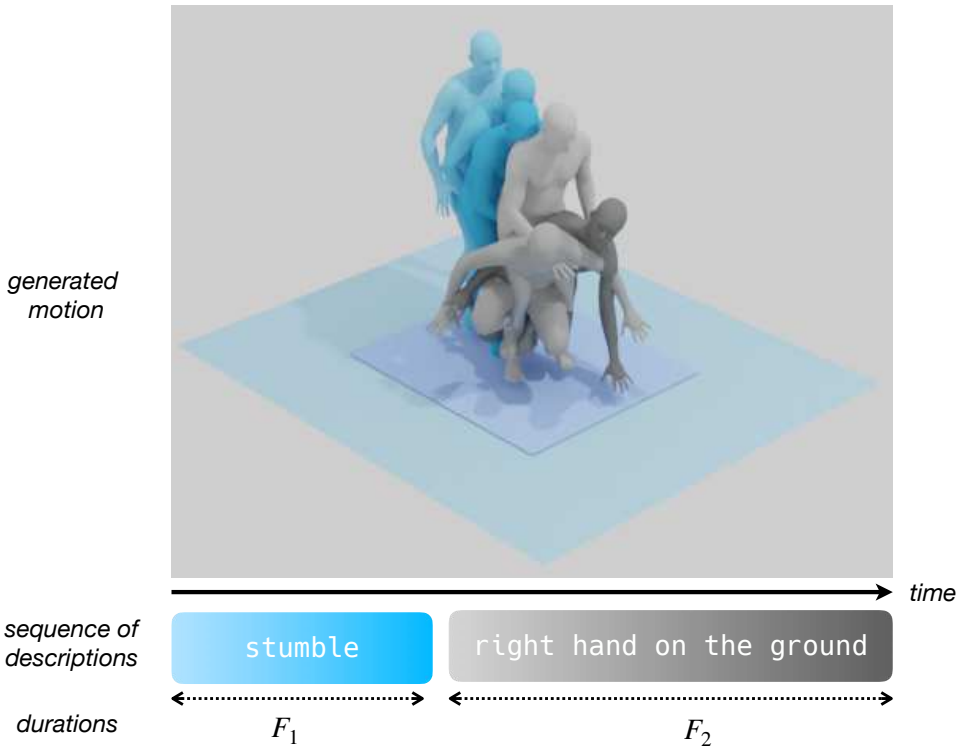


FIGURE 3.1: **Goal:** Given a sequence of descriptions and durations as input, our goal is to generate a 3D human motion respecting the instruction and achieving temporal action compositionality. We design a recursive approach, TEACH, that can produce a variable number of actions given a stream of textual prompts. Note that the color saturation is aligned with the progress of each action.

et al., 2019]. BABELis described in more detail in Appendix A. This dataset is both larger and more diverse than previous datasets. While previous work uses BABEL for its categorical action annotations (60/120 classes) focusing mostly on classification settings [Tevet et al., 2022] or motion generation [Xu et al., 2023], we directly train with its free-form language descriptions, which have not been used before. Thus, our generated motions cover a significantly wider variety of actions compared to the state of the art [Petrovich et al., 2022].

TEMOS [Petrovich et al., 2022] established a new baseline in 3D motion synthesis conditioned on text descriptions, using Transformer-based

VAEs [Petrovich et al., 2021] and pretrained language models [Sanh et al., 2019] to sample realistic motions corresponding to a text input. TEMOS is trained on the KIT dataset [Plappert et al., 2016] which contains only single-action data. Besides being trained on the KIT data, TEMOS is not directly applicable to our task of generating a *sequence* of actions. While its non-autoregressive formulation generates high-quality motions, the approach does not readily scale to long sequences of multiple motions due to the quadratic time complexity of Transformers. Moreover, to embed complex sequences of actions in the latent space would require seeing a combinatorial number of action combinations during training. With existing training data, generalization to new sequences would be challenging. In this Chapter, we combine the best of both worlds, by designing an iterative model that generates one motion per action at a time, by conditioning on the previous motion. Within each iteration, we keep the non-autoregressive action generation approach, which probabilistically generates diverse and high-quality motions (see Figure 3.1). We experimentally show that our iterative method compares favorably against baselines that jointly or independently generate pairs of actions in a single shot (Figure 3.5).

One of the key challenges in synthesizing long action sequences given a stream of textual prompts is how to ensure *continuity* within the transitions between actions. Independently generating one motion per action would not guarantee temporal smoothness. In our framework, we find that encoding the next action conditioned on the last few frames of the previous action is a simple and effective solution. To account for any remaining discontinuities still present with this solution, we apply spherical linear interpolation (Slerp) over a short time window. Note that our approach treats transitions significantly better than a baseline that uses Slerp to interpolate between independently generated actions.

Our contributions are the following: (i) We introduce and establish a new benchmark for temporal action composition of 3D motions on the BABEL dataset; (ii) We design a new hybrid NN model, TEACH (TEmporal Action Composition for 3D Humans), that addresses the limitations of previous state of the art methods by iteratively generating

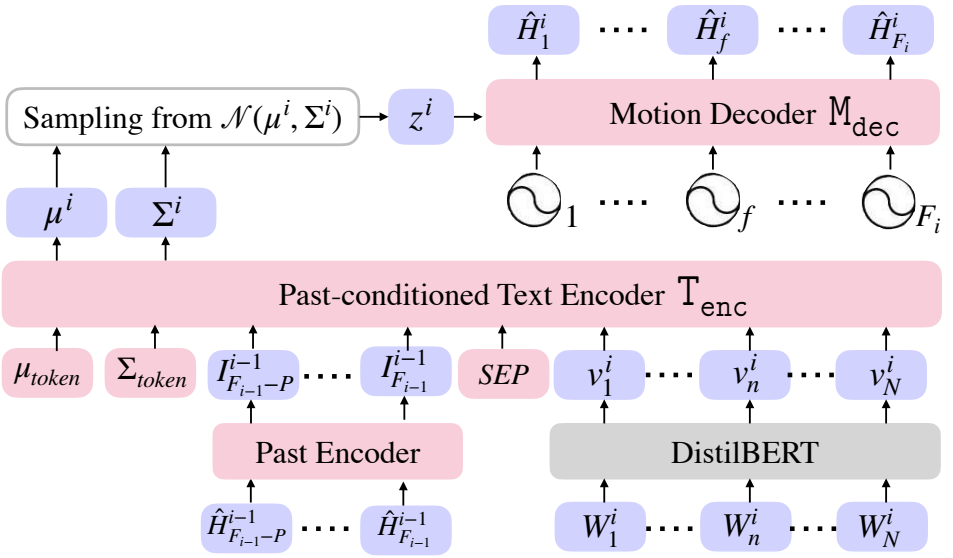


FIGURE 3.2: **Method overview:** Our TEACH model is a variational encoder-decoder NN. The current text instruction and the past frames are encoded by the corresponding encoders and are fed to T_{enc} along with the additional tokens. T_{enc} produces the distribution parameters from which the latent vector is sampled and given to the decoder to generate a sequence of 3D human poses. In this figure, we omit the motion encoder for simplicity.

infinitely many actions with smooth transitions; (iii) We obtain promising results for text-to-motion synthesis from a large-vocabulary of actions.

3.2 Motion synthesis with TEACH

First, we define the task (Section 3.2.1) of temporal composition. Next, we present the TEACH architecture and explain the different baselines and architectural components of our method (Section 3.2.2). Finally, we describe the training procedure and the losses (Section 3.2.3).

3.2.1 Task definition

Starting from a sequence of instructions in natural language (e.g., in English), our goal is to generate smooth and realistic human motions that correspond to the instructions. Here, we demonstrate results for

pairs or triplets of actions but our model can autoregressively generate an arbitrary sequence of actions given the respective action descriptions. During training, TEACH takes as input: a sequence of English prompts $S_1, \ldots, S_K$, where each phrase corresponds to an action description and consists of a sequence of words $S_i = W_1^i, \ldots, W_N^i$, e.g., "turn to the right", "walk forward", "sit down", etc. and a 3D human motion sequence. Each sequence consists of poses, $H_1, \ldots, H_F$, parametrized by the SMPL body model [Loper et al., 2015]. In Chapter, we follow the representation described in [Petrovich et al., 2022] that converts SMPL parameters to a $6D$ rotation representation [Zhou et al., 2019] together with root translation. Moreover, we use the same normalization and canonicalization process as in [Petrovich et al., 2022].

3.2.2 Architecture

Our architecture is inspired by TEMOS [Petrovich et al., 2022]. We use the same language encoder (DistillBERT) and motion encoder. We omit the details of the motion encoder as it is the same used in TEMOS. However, TEMOS is constrained to output a sampled motion given a language description, without being able to handle sequences of actions. Thus, we design a new text encoder architecture, which includes a Past Encoder (PC) that provides our method with the context of the previous action when generating the second action in each pair. For the first motion in each pair, we disable the Past Encoder and only use the learnable tokens and the encoded text. A separation token is used to facilitate disambiguation of motion and text modalities in the model [Devlin et al., 2019]. As illustrated in Figure 3.2, we encode the current text instructions $W_1^i, \ldots, W_N^i$ using a pre-trained, frozen text model (DistilBERT) into text features $\mathrm{v}_1^i, \ldots, \mathrm{v}_N^i$. Moreover, the last P frames of the previous generated motion, $\widehat{H}^{i-1}_{F_{i-1}-P:F_{i-1}}$, are encoded into motion features $I_{F_{i-1}-P:F_{i-1}}$ (Past Encoder). Then, we combine the features from the previous action, $I^{i-1}_{F_{i-1}-j}, j \in \mathbb{N}$, and the current text features along with learnable tokens (μ_{token}, Σ_{token} and SEP), and pass them as inputs to the Past-conditioned Text-Encoder, which generates the distribution parameters μ^i and Σ^i. μ^i and Σ^i are treated as

parameters of a Gaussian distribution, from which we sample and decode the final motion. Next we explain each module separately.

Past-conditioned text encoder. We first encode the natural language descriptions with a frozen DistilBERT [Sanh et al., 2019] which takes as input the current text instruction $S_i = W_1^i, \ldots, W_N^i$ and outputs text features $\mathrm{v}_1^i, \ldots, \mathrm{v}_N^i$. We use a Transformer encoder architecture to encode the past motion corresponding to the last P frames of the previous action. The past motion $\hat{H}^{i-1}_{F_{i-1}-P}, \ldots, \hat{H}^{i-1}_{F_{i-1}}$ is transformed into pose features $I^{i-1}_{F_{i-1}-P}, \ldots, I^{i-1}_{F_{i-1}}$. Finally, we use a transformer encoder as the Past-conditioned Text Encoder module T_{enc} to jointly encode the past motion features and the current text features into μ^i and Σ^i, parameters of a Gaussian distribution. This network takes as extra inputs, the μ_{token} and Σ_{token} as in ACTOR [Petrovich et al., 2021], and a special token (*SEP*) to separate both modalities. From the Gaussian distribution $\mathcal{N}(\mu^i, \Sigma^i)$, we sample a latent vector z^i. For the first motion we disable PC since there is no previous motion.

Motion decoder. We use the same decoder architecture, generating a sequence of poses from a single embedding, following TEMOS [Petrovich et al., 2022]. This Transformer-based motion decoder takes the current latent vector z^i and F_i positional encodings (in the form of sinusoidal functions) as input, and generates the sequence of human motions.

Baselines. As there is no prior work—except for interpolation-based ones, like Slerp— that explicitly benchmarks and learns a model for sequences of actions, we design several baselines using TEMOS, Slerp [Shoemake, 1985], and geometric transformations. Note that TEACH is different from the action-driven motion prediction proposed concurrently in [Mao et al., 2022], as we deal with full motion generation and free-form language descriptions and do not explicitly use only pairs formed by transitions. Specifically (Section 3.5), we employ two baselines: "Independent", which is based on TEMOS and is trained on single action segments, and "Joint", which is also based on TEMOS, but takes as input both the motions (i.e., concatenation of the respective segments) and the corresponding language labels separated with a comma. For the case of Independent, the generated motions are fused into a pair of actions by: (1) aligning the last

generated frame of the first action with the first frame of the second action by orientation and translation and (2) applying spherical interpolation to fill in the remaining transition between the two actions.

3.2.3 Training

Data handling. BABEL consists of language descriptions and action categories for the majority of sequences in AMASS [Mahmood et al., 2019]. Each sequence is separated into segments that can overlap without any constraint, except that all the frames of the sequence must be labeled. To train the Independent baseline, we use the training set segments from BABEL. Furthermore, we extract pairs of actions to train the remaining models. To achieve this, we process each sequence and, for each segment, $s_i = [t_s^i, t_e^i]$, we calculate all the segments $s_j : s_j \cap s_i \neq \varnothing, s_j \nsubseteq s_i, s_j \ngeq s_i$, except if a segment is a "transition". We think of transitions happening *between* actions so a transition and an action cannot form a pair. Simply, we use all the segments that are not a superset or subset of each other and have an overlap. Next, if a segment is connected to another segment via a transition (i.e., the same transition overlaps with both), that triple is considered a pair of actions. For the rest of the cases, the pairs are formed by the segment overlaps and not transitions.

A training iteration consists of two forward passes. The first action, corresponding to sentence description S_1, and the given length F_1, will produce μ^1, Σ^1 and the generated motion $\hat{H}_1^1, \dots \hat{H}_{F_1}^1$. Then, given the second instruction S_2, the given length F_2, and the P last frames of the previous generated motion, we produce μ^2, Σ^2 and the generated motion of the second segment $\hat{H}_1^2, \dots \hat{H}_{F_2}^2$. We do one backward pass, which optimizes the reconstruction loss and the KL loss on the two segments jointly.

Reconstruction loss. From the two forward passes, we generate the motions $\widehat{H}_{1:F_1}^1$ and $\widehat{H}_{1:F_2}^2$. We enforce them to be close to the corresponding ground truth motions $H_{1:F_1}^1$ and $H_{1:F_2}^2$ via the following loss terms:

$$\mathcal{L}_R = \mathscr{L}(H_{1:F_1}^1, \widehat{H}_{1:F_1}^1) + \mathscr{L}(H_{1:F_2}^2, \widehat{H}_{1:F_2}^2), \tag{3.1}$$

where $\mathscr{L}$ is the smooth L1 loss.

KL loss. By using the notation $\phi^i = \mathcal{N}(\mu^i, \Sigma^i)$, and $\psi = \mathcal{N}(0, I)$, this loss regularizes the two Gaussian distributions ϕ^1 and ϕ^2 to be close to ψ as in the VAE formulation. We minimize the KL divergences

$$\mathcal{L}_{KL} = KL(\phi^1, \psi) + KL(\phi^2, \psi). \tag{3.2}$$

We also use the same additional KL losses as TEMOS, to enforce the latent vectors to follow the same distributions and the same L_1 loss to keep them as close as possible. We omit them from the description for simplicity and to highlight our technical contributions. The total loss is a weighted sum of the two terms: $\mathcal{L} = \mathcal{L}_R + \lambda_{KL}\mathcal{L}_{KL}$. In practice, we use $\lambda_{KL} = 10^{-5}$ as in TEMOS [Petrovich et al., 2022] and ACTOR [Petrovich et al., 2021].

Implementation details. For both the Past-Conditioned Text Encoder and the Past Encoder, we use a Transformer encoder model with 6 layers and 6 heads, a dropout of 0.1 and a feed-forward size of 1024. The latent vector dimension is 256. The whole model is trained with the AdamW optimizer [Loshchilov and Hutter, 2019] with a fixed learning rate of 10^{-4} with a batch size of 32 or 16. Both during training and test time, we use ground truth durations (F_i). We also use Slerp [Shoemake, 1985] and alignment between the first and the second action for TEACH as well as for the Independent baseline. We apply Slerp to interpolate for 8 frames at the beginning of the second motion which includes the transition.

Runtime. As described in this Section 3.2, we train 3 models: Independent, Joint and TEACH. Our computational resources are various types of GPUs, mainly "Tesla V100-PCIE-16GB", "Tesla V100-PCIE-32GB", "Tesla V100-SXM2-32GB". We trained each model with a single GPU. Independent training time is 2 days approximately, Joint is 5 days and TEACH is 2 days to reach 600 epochs. The Joint baseline due to the quadratic increase in time complexity for transformers is significantly slower to train and difficult to scale to more than two actions.

3.3 Experiments

We first describe the dataset (Section 3.3.1) and evaluation metrics (Section 3.3.3) used in our experiments. We then report our main results by comparing our method with multiple baselines (Section 3.3.4). Next, we present an ablation study to investigate the contribution of motion interpolation (Section 3.3.6) and different numbers of past frames used by PC (Section 3.3.7). Finally, we provide qualitative results (Section 3.3.8) and a discussion on limitations (Section 3.3.9).

3.3.1 BABEL dataset

We train and evaluate on BABEL [Punnakkal et al., 2021], which provides textual descriptions for the motions in the AMASS collection [Mahmood et al., 2019]. In particular, we use the processed text version (lemmatized etc. as opposed to the raw version which is also provided). We do not use the categorical action labels. In total there are 10881 motion sequences, with 65926 textual labels and the corresponding segments. The unique property of BABEL is that it has annotated segments that overlap in each sequence, which allows us to investigate generation of a *sequence of actions*. In contrast, a textual label in KIT [Plappert et al., 2016] covers the entire sequence. Moreover, KIT is smaller both in terms of the number of motion sequences and the number of actions. Figure 3.3 shows the distribution of verbs according to the most frequent verbs in BABEL. Refer to Section 3.3.2 for additional analysis of KIT's most frequent verbs compared with BABEL and also analysis of other part-of-speech categories. There are approximately 5.7k and 23.4k pairs in the validation and training sets, respectively. We consider pairs of actions for simplicity, but TEACH is applicable to sequences of actions of arbitrary length. Note that, we do not use "t-pose" or "a-pose" actions during training. We use t ransitions only to identify possible pairs of actions. During training, in the case of segment overlap, we uniformly distribute the overlapping frames across the two segments that constitute the pair. Furthermore, note that the majority of the pair data ($\sim$ 70%) is created by overlapping segments and not by

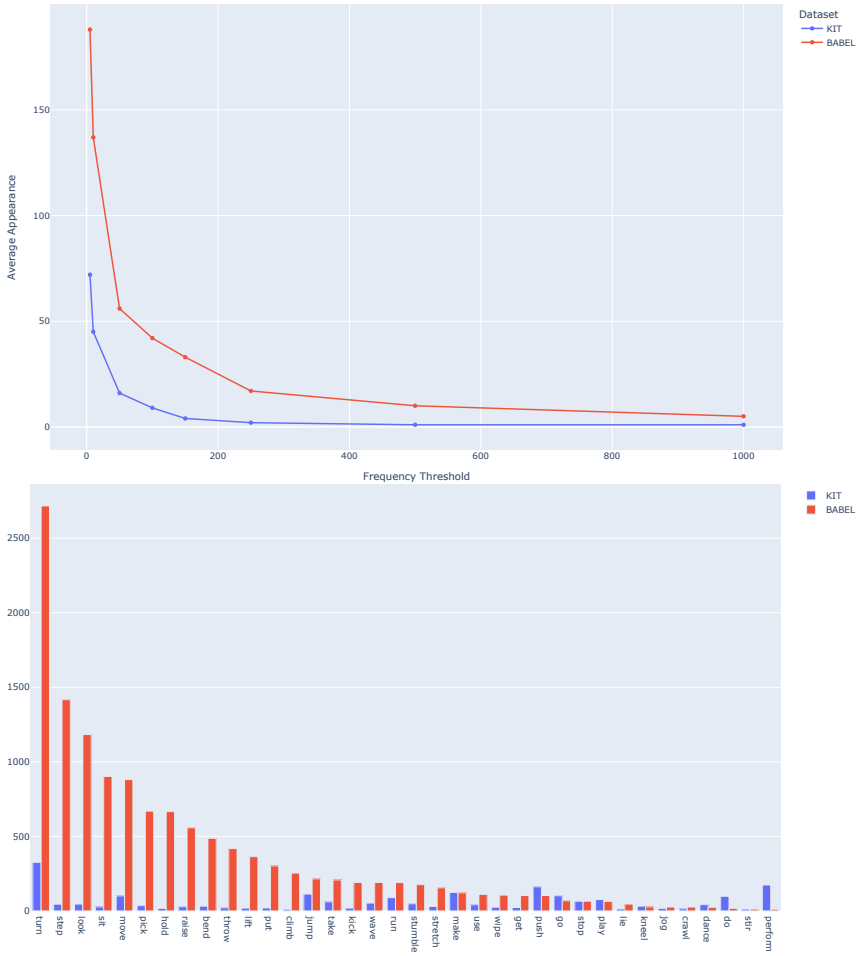


FIGURE 3.3: **BABEL vs KIT:** We provide a comparative analysis of the amount of data and the vocabulary of verbs. On the top, the number of different words (y-axis) that appear more than n times is plotted against various frequency thresholds (y-axis). This represents how many words occur at least n times. We see that BABEL consistently has at least twice as many tokens as KIT. On the bottom, the verb histogram shows that BABEL has more samples across a wide range of actions. Note that there are differences in how the datasets label actions with generic words like "do" and "perform" being common in KIT and rare in BABEL, which is more specific.

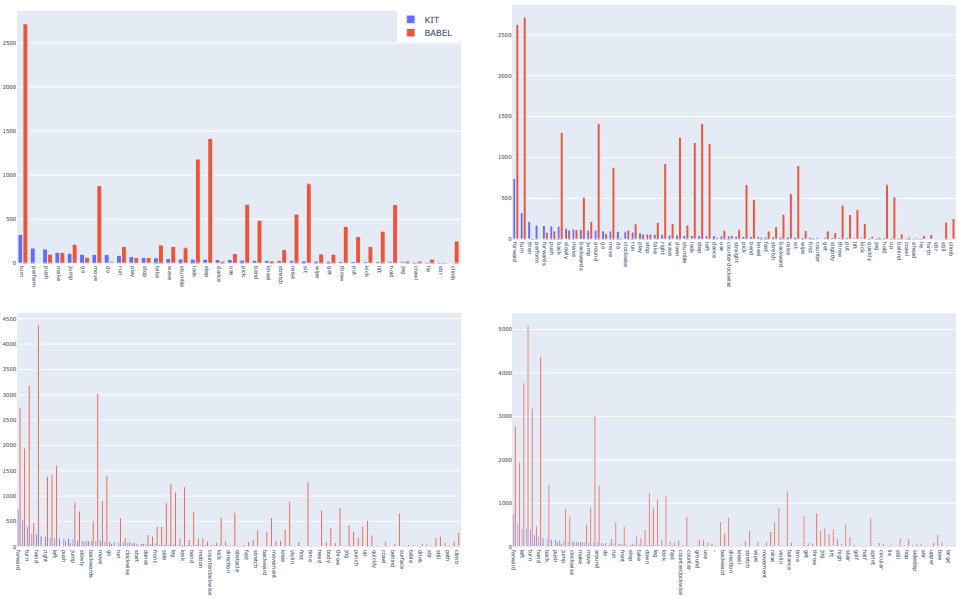


FIGURE 3.4: **BABEL vs KIT:** Here, we show additional plots regarding the language statistics of BABEL and KIT. We show the token frequency of the two different datasets for different POS tag groups. Verbs (top), verbs and adverbs (second), verbs, adverbs, and nouns (third) and verbs, adverbs, nouns, adjectives (fourth).

transitions. In the case of a transition, we concatenate the transition with the second segment.

3.3.2 Extended dataset statistics

Similar to Section 3.3.1, we provide further statistics on the BABEL dataset [Punnakkal et al., 2021].

Language statistics

In Figure 3.4, we demonstrate the frequency of different tokens for the two datasets, but this time sorted according to the most frequent words in the vocabulary of the KIT dataset [Plappert et al., 2016]. As we have shown in Figure 3.3, BABEL is at least twice as rich in terms of language. Here, we show that, even for the most frequent tokens of KIT and for all the different POS (part of speech) tag combinations, the same tokens appear much more

Datatype	Mean(s)	Std(s)	Median(s)	Samples(#)
Segments	1.97	3.73	1.1	61639
Segments*	2.49	4.46	1.38	40395
Sequences	6.61	6.85	4.23	4287
Seg* + Seq	2.88	4.89	1.52	44682
Pairs	4.98	5.98	3.4	49820

TABLE 3.1: **Datatype statistics:** We show the statistics from different BABEL label types. Sequences are the AMASS [Mahmood et al., 2019] motions with a single action label, Segments are the smaller motions that are extracted from longer AMASS sequences. Pairs are the two-action motions that we extract from consecutive segments. We denote the exclusion of "transition", "a-pose", and "t-pose" labels with *.

often in BABEL. Note that we show 4 different plots for the 4 cases of POS tags which are highly probable to involve an action.

Duration statistics

In Tab. 3.1, we show the statistics of different data types in BABEL. To be clear, both Segments and Sequences datatypes are included in BABEL. Using the consecutive Segments, we build the pairs, as the Sequences contain only a single action. However, Sequences are included in the independent model training. We observe from Tab. 3.1, that the mean length of the Segments is 3 times smaller than those of the Sequences. Moreover, the median is consistently smaller than the mean which implies that the distribution of the durations are long-tailed.

Data processing

We use BABEL's original splits for the training and validation set. We report our final results in the validation set, for easier reproduction, since the BABEL test set is not publicly available. We use AMASS motions subsampled at 30 fps. When training with pairs, we remove motion pairs that have a duration shorter than 0.3 seconds or longer than 25 seconds.

Similarly, we remove motion segments or sequences that are shorter than 0.3 seconds when training the Independent model and whenever a motion is longer than 5 seconds, we take a random 5-second subset to feed as input. We train the independent baseline using both Segments and Sequences.

For the canonicalization of input motions, we follow the same setup as in [Petrovich et al., 2022] by rotating the bodies to face the forward direction. We canonicalize each action separately for the independent baseline. For TEACH and the joint baseline, we canonicalize the entire sequence according to the first frame. We also standardize the data (similar to [Petrovich et al., 2022]) for each of the different cases (pairs, single-action).

3.3.3 Evaluation metrics

We follow the evaluation metrics employed by [Ghosh et al., 2021; Petrovich et al., 2022], namely Average Positional Error (APE) and Average Variational Error (AVE), measured on the root joint and the rest of the body joints separately. Mean *local* and *global* refer to the joint position in the local (with respect to the root) or global coordinate systems, respectively. As in [Petrovich et al., 2022], we sample one random motion generation from our variational model and compare it against the ground truth motion corresponding to the test description. While we quantitatively evaluate on pairs of actions, we qualitatively show the ability of our model to generate two or more actions in Figures 3.7, 3.8 and the project website [1].

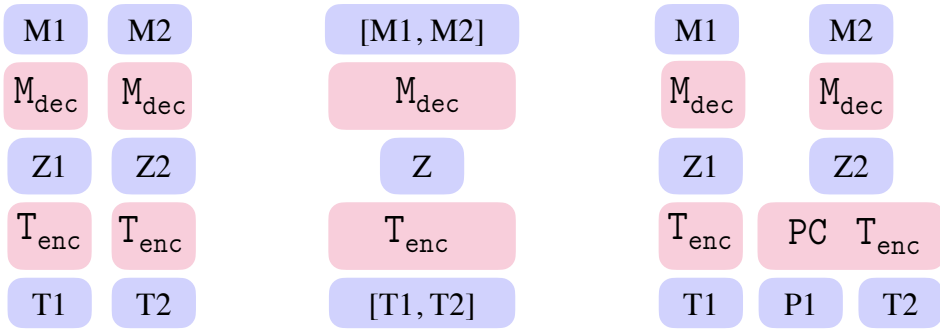
3.3.4 Comparison with baselines

Here, we first describe the baselines we created by adapting TEMOS to the action sequence synthesis task without any architectural changes [Petrovich et al., 2022]. Figure 3.5 summarizes the two variants (a) independent and (b) joint training.

[1] https://teach.is.tue.mpg.de

Methods	Average Positional Error ↓				Average Variance Error ↓			
	root	gl. traj.	mean loc.	mean gl.	root	gl. traj.	mean loc.	mean gl.
(a) Independent	0.729	0.707	0.169	0.770	0.255	0.253	0.016	0.267
(b) Joint	0.790	0.773	0.163	0.832	0.306	0.305	**0.014**	0.317
(c) TEACH	**0.674**	**0.654**	**0.159**	**0.717**	**0.222**	**0.220**	**0.014**	**0.234**

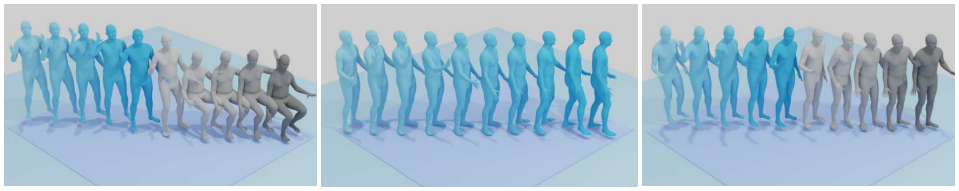
TABLE 3.2: **Comparison against baselines on pairs of actions:** We benchmark the 3 different approaches on action pairs of BABEL [Punnakkal et al., 2021]. As we can see TEACH outperforms Joint and Independent baselines on all the metrics.



(a) Independent (b) Joint (c) Past-Conditioned

FIGURE 3.5: **Variants:** We illustrate the baselines for independent single-action training (a) and joint two-action training (b). Our method on the other hand is recursive, and is conditioned on the past motion (c). T1 and T2 denote the sequence of two textual descriptions. M stands for motion, and Z stands for the latent vector.

Independent training, in Figure 3.5 (a), refers to inputting a single text and outputting a single motion, as is the case for TEMOS. To adapt this model to a sequence of actions, we generate the two actions and perform an interpolation operation (i.e., Slerp) to obtain smooth transitions between the independently generated motions. However, a naive interpolation results in poor transitions since the second motion may start at a different global location, with a different global rotation. To account for this mismatch, we translate the root of the second motion to have the same x,y coordinates as the first motion. Then, we rotate it to match the first motion's global orientation. The advantage of this



(a) Independent (b) Joint (c) TEACH

FIGURE 3.6: **Qualitative comparison:** We show an illustrative example for (a) Independent, (b) Joint and (c) TEACH for the sequence of actions [`wave the right hand`, `raise the left hand`]. While the individual waving and raising hand actions are correctly generated, the single-action independent baseline (a) transitions from standing to sitting incoherently as the next action is not conditioned on the past. Joint baseline (b) on the other hand, waves with the right hand but does not raise the left one, probably because such an action combination was not present in the training set. On the other hand, TEACH learns about both single action variation and autoregressive transitions between actions, and thus completes both actions naturally. Note that, while these motions are performed in place, we artificially translate each pose to show the motion frame-by-frame such that the transition and action details are easier to see.

model is its ability to scale up to any number of action compositions. However, despite the interpolation, we observe unnatural motions due to large changes between body poses during transitions. For example, in Figure 3.6, the model generates two motions that are incompatible in terms of pose, creating an unrealistic transition. This is expected, as the independent baseline has no notion of the previous motions.

The joint training, in Figure 3.5 (b), is another alternative to extend TEMOS to multiple descriptions without further modifications. We simply combine the sequence of descriptions into a single text with a comma in between, and train the model with pairs of motions corresponding to consecutive actions. The advantage of this model is the ability to produce smooth motions, including the transitions. However, the major disadvantage concerns scalability. Due to quadratic complexity with respect to the motion duration, the joint training does not scale well to a large number of actions. Moreover, it would require many action combinations, i.e., a concatenation of more than pairs of actions, at training to produce a variable number of actions. Such data is difficult to capture, making it challenging to train such a model. Finally, it may be difficult to

generalize to unseen action combinations. In our experiments, we train this model with 2-action pairs (which is a relatively easy setting compared to more actions).

In contrast to independent and joint training, our model (Figure 3.5 (c)) is recursive and the future action is conditioned on the previous action. In Tab. 3.2, we summarize the performance of these three variants on the BABEL validation set. Our past-conditioned TEACH, which uses the last 5 frames from the previous action, outperforms the baselines. Due to the difficulty of quantitative evaluation of generative models, we also rely on qualitative comparisons, provided in the video[2]. An illustration of our arguments can be seen in Figure 3.6.

Methods	Transition Dist.		Average Positional Error ↓				Average Variance Error ↓			
	w/ al..	w/out al..	root	gl. traj.	mean loc.	mean gl.	root	gl. traj.	mean loc.	mean gl.
Indep. (no Slerp)	0.151	0.177	0.762	0.740	0.170	0.805	0.255	0.253	0.016	0.267
TEACH (no Slerp)	**0.107**	**0.122**	0.677	0.658	**0.159**	0.722	0.227	0.225	0.015	0.239
Indep.	n/a	n/a	0.729	0.707	0.169	0.770	0.255	0.253	0.016	0.267
TEACH	n/a	n/a	**0.674**	**0.654**	**0.159**	**0.717**	**0.222**	**0.220**	**0.014**	**0.234**

TABLE 3.3: **Effect of Slerp:** We measure transition distance for generated samples given all the test set pairs. We define transition distance as the Euclidean distance between the last frame of the first action and the first frame of the second action, calculated on joint positions, when the last pose of the first action is aligned with the first pose of the next action and when it is not. TEACH better captures the transition between the two actions compared to the previous-action-agnostic TEMOS. Moreover, the Independent baseline cannot be benchmarked without orienting and aligning the poses as it is trained on single actions that are canonicalized to face in the forward direction.

3.3.5 Alignment & interpolation

To ensure that the applied Slerp interpolation is both effective and realistic, we first align the action pairs before performing the interpolation. To this end, we translate and rotate the second motion such that the last frame of the first action and the first frame of the second action match. Then, we interpolate between those aligned poses via Slerp. We insert 8 frames

[2] https://teach.is.tue.mpg.de

P	Average Positional Error ↓				Average Variance Error ↓			
	root joint	global traj.	mean local	mean global	root joint	global traj.	mean local	mean global
1	0.725	0.704	0.160	0.766	**0.222**	**0.220**	**0.015**	**0.234**
5	**0.674**	**0.654**	0.159	**0.717**	**0.222**	**0.220**	**0.015**	**0.234**
10	0.718	0.698	**0.157**	0.759	0.238	0.237	**0.015**	0.250
15	0.719	0.699	0.163	0.761	0.238	0.236	0.014	0.250

TABLE 3.4: **Ablation on the number of past frames:** Here, we change the number of past frame, while keeping the other training settings identical and report the different metrics. We observe the best performance when using 5 past frames.

between the two actions for interpolation. For the independent baseline, the alignment step is crucial since the single actions are independently generated without ensuring continuity within transition poses.

3.3.6 Effect of interpolating the action transitions

As explained in Sections 3.2 and 3.3.4, we use Slerp interpolation between actions both for the independent training baseline, and our method. We justify the use of such interpolation with the experiment in Tab. 3.3. Removing Slerp causes discontinuities which are easier to see visually from our video[3]. However, the discontinuity is higher for the independent generations than in TEACH. To measure the degree of discontinuity, we report the average transition distance, i.e., the Euclidean distance between the two body poses corresponding to the last frame of the previous action, and the first frame of the next action. We see a clear decrease in discontinuity with TEACH (0.107 vs 0.151 m), even when the bodies are aligned. Moreover, we measure the same metric in the absence of alignment for the global orientation of the second motion (see Section 3.3.4). We see that this alignment step is crucial for the independent baseline, as the transition distance compared to TEACH is even worse if we do not apply any alignment at all (0.122 to 0.177), demonstrating that TEACH models generate smoother transitions than the baseline.

[3] https://teach.is.tue.mpg.de

3.3.7 Past conditioning duration

In Tab. 3.4, we investigate the influence of the hyperparameter P, the number of frames from the past motion to input to the past-conditioned text encoder. While the performance is similar across 1, 5, 10, or 15 frames, we observe a slight improvement when using 5 frames as opposed to 1 frame, potentially because a single frame does not capture enough past information. However, further increasing the number of past frames does not improve the results.

3.3.8 Qualitative analysis

We present qualitative motion generation results in Figures 3.7, 3.8. In contrast to previous work that trains models on the KIT dataset [Plappert et al., 2016], our model can go beyond locomotive motions, and covers a wider variety of actions, such as `right hand on the ground`. Finally, we show examples of more than 2 actions in the last row of Figure 3.7 and examples for multiple actions along with their transitions in a single image in Figure 3.8. We refer to our website [4] for viewing the motions as a video, providing analyses of the effect of interpolation and motions beyond pairs of actions.

3.3.9 Limitations

Our attempt to tackle temporal compositions has limitations. TEACH is susceptible to acceleration peaks when transitioning from the first action to the second one. There is still the need to apply Slerp to smooth out these discontinuities but, as we see in Tab. 3.3, the starting/ending poses of the two actions are not far away. This behavior may also be attributed to the variational nature of the model, which makes it difficult to precisely match the previous motion without any explicit pose-level autoregressive constraints. Moreover, BABEL has a lot of overlapping actions which

[4] https://teach.is.tue.mpg.de

makes it difficult sometimes to have a visually "clear" sequence of actions, as some actions might mix with others.

3.4 Conclusion

We presented the new task of motion generation from a sequence of textual prompts, which we refer to as action compositions in time. We established a new benchmark on the BABEL dataset for this task, and explored various strong baselines, including independently or jointly training pairs of actions. Our recursive approach, TEACH , improves over the baselines quantitatively, while addressing the past limitations by allowing variable numbers of actions and producing fewer discontinuities at transitions. While we obtain promising results within this new direction, there is still room for improvement. Motion realism can be improved and contact with the world could be explicitly modeled. Here, we assume that the character does not know what it will do in the future; that is, it only looks backwards in time. In contrast, humans have goals and know what they will do next. This knowledge about the future can affect the present. Future work can explore such "looking ahead" to better generate realistic sequences of actions. We hope that TEACH will encourage further research on combining language and 3D motion, much like the field has done with language and 2D images [Ramesh et al., 2021; Saharia et al., 2022].

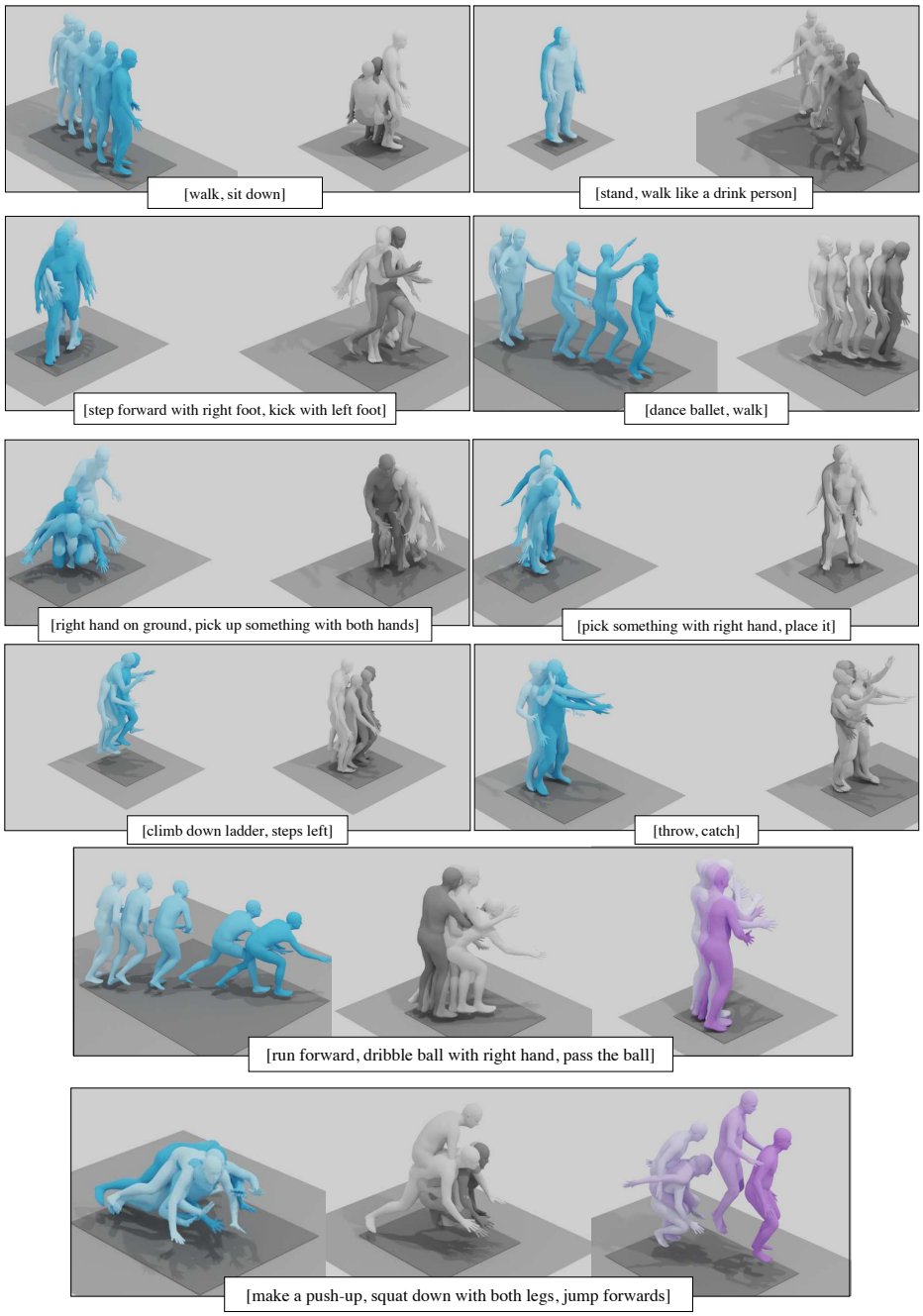


FIGURE 3.7: **TEACH qualitative results:** In the first 4 rows, we visualize TEACH results for pairs of actions. We see how TEACH generates, in all the cases, the two actions. Even finegrained sequences of action like 'step forward with the right foot' and 'kick with the left foot' are generated accurately. In the final two rows, we show triplets of actions. We use a separate image for each action in the sequence to make the performed action clearly visible. We denote the ending of the first action with the most saturated version of cyan, while the starting of the second is the less saturated version of gray.

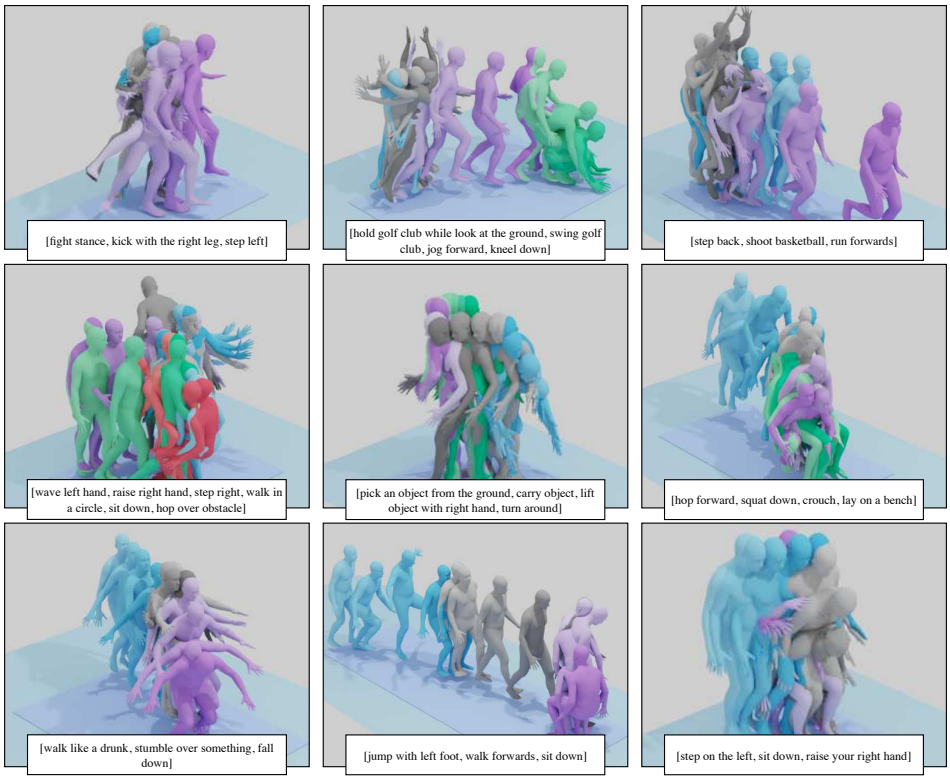


FIGURE 3.8: **TEACH qualitative results in a single sequence for multiple actions:** We visualize TEACH results for multiple actions in the same image. We show all the actions in the same image for completeness and to demonstrate the smoothness of TEACH transitions between actions. We see how TEACH generates accurately, in all the cases, multiple actions. Given its autoregressive generation sequence-level TEACH generates multiple actions that are finegrained, i.e., from 'wave' or 'raise' the left/right hand to high-level motion such as 'walk like a drunk' or sports related actions e.g., 'shoot basketball'. We denote the ending of the first action with the most saturated version of cyan, while the starting of the second is the less saturated version of gray. Similarly, we use purple for the third, green for the fourth and red for the fifth action if they exist.

Chapter 4

Spatial Compositions of 3D Human Motions

Building on our exploration of temporal action compositions, our second contribution investigates the spatial hierarchies inherent in human motion — for example, 'waving hand' while 'walking' at the same time. While temporal compositions address *when* actions occur in sequence, spatial compositions focus on *how* multiple actions are performed simultaneously across different body parts. Hence, a main problem of spatial compositions is identifying and composing motions from different body parts, as opposed to composing whole-body motions by synthesizing transitions. To solve this, motivated by the observation that the correspondence between actions and body parts is encoded in powerful language models, we extract this knowledge by prompting GPT-3 with text such as "What are the body parts involved in the action <action name>?", while also providing the parts list and few-shot examples. Given this action-part mapping, we combine body parts from two motions together and establish the first automated method to spatially compose two actions. However, training data with compositional actions is always limited by the combinatorics. Hence, we further create synthetic data with this approach, and use it to train a new state-of-the-art text-to-motion model SINC ("SImultaneous actioN Compositions for 3D human motions"). SINC synthesizes 3D human motions from textual descriptions involving concurrent actions, such as "waving hand while walking." Our contributions in this chapter complement

our temporal composition work from the previous chapter, targeting a previously unsolved problem of automating spatial compositions. In our experiments, we find that training with such GPT-guided synthetic data improves spatial composition generation over baselines. Our code is publicly available at sinc.is.tue.mpg.de.

4.1 Introduction

Text-conditioned 3D human motion generation has attracted increasing interest in the research community [Petrovich et al., 2022; Guo et al., 2022a; Athanasiou et al., 2022], where the task is to input natural language descriptions of actions and to output motion sequences that semantically correspond to the text. Such controlled motion synthesis has a variety of applications in fields that rely on motion capture data, such as special effects, games, and virtual reality. While there have been promising results in this direction, *finegrained* descriptions remain out of reach. Consider the scenario in which a movie production needs a particular motion of someone jumping down from a building. One may generate an initial motion with one description, and then gradually refine it until the desired motion is obtained, e.g., {'jumping down', 'with arms behind the back', 'while bending the knees'}. State-of-the-art methods [Petrovich et al., 2022; Chen et al., 2023] often fail to produce reasonable motions when conditioned on finegrained text describing multiple actions. In this Chapter, we take a step towards this goal by focusing on the *spatial composition* of motions. In other words, we aim to generate one motion depicting multiple simultaneous actions; see Figure 4.1. This paves the way for further research on finegrained human motion generation.

Previous work [Lin et al., 2018; Ahuja and Morency, 2019; Ghosh et al., 2021; Petrovich et al., 2022] initially explored the text-conditioned motion synthesis problem on the small-scale KIT Motion-Language dataset [Plappert et al., 2016]. Recent works [Guo et al., 2022a; Athanasiou et al., 2022] has shifted to the large-scale motion capture collection AMASS [Mahmood et al., 2019], and its language labels from BABEL [Punnakkal et al., 2021] or HumanML3D [Guo et al., 2022a].

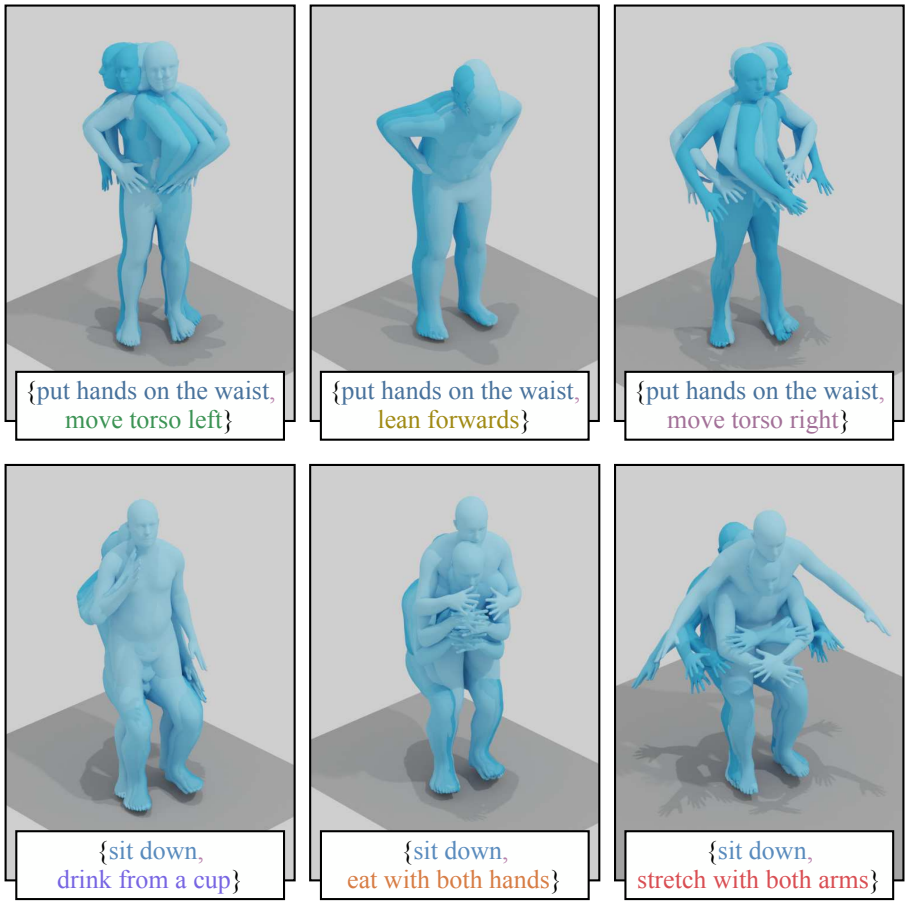


FIGURE 4.1: **Goal:** We demonstrate the task of *spatial compositions* in human motion synthesis. We generate 3D motions for a pair of actions, defined by a pair of textual descriptions. Here, we provide six sample input-output illustrations from our model. For example, we input the set of actions {'put hands on the waist', 'move torso left'} and generate one motion that simultaneously performs both.

In particular, similar to this work, our work in Chapter 3 focuses on finegrained descriptions by addressing temporal compositionality, that is, generating a sequence of actions, one *after* the other. We argue that composition in time is simpler for a model to learn, since the main challenge is to smoothly transition between actions. This does not necessarily require action-specific knowledge. Moreover, Slerp [Shoemake, 1985] may yield acceptable results in simple cases and when ensuring that

start and end frames are close-by — although it lacks the expressiveness needed for more complex or realistic transitions. However, there is no such easy solution for spatial compositions. Animators have to manually define and post-process the actions to compose them[4]. On the other hand, there is no such trivial solution for compositions in *space*, since one needs to know action-specific body parts to combine two motions. If one knows that 'waving' involves the hand and 'walking' involves the legs, then compositing the two actions can be performed by cutting and pasting the hand motion into the walking motion. This is often done manually in the animation industry.

To automate this process, we observe that pretrained language models such as GPT-3 [Brown et al., 2020] encode knowledge about which body parts are involved in different actions. This allows us to first establish a spatial composition baseline (analogous to the Slerp baseline for temporal compositions); i.e., independently generating actions and then combining with heuristics. Not surprisingly, we find that this is suboptimal. Instead, we use the synthesized compositions of actions as additional training data for a text-to-motion network. This enriched dataset enables our model, called SINC ("SImultaneous actioN Compositions for 3D human motions"), to outperform the baseline. Our GPT-based approach is similar in spirit to work that incorporates external linguistic knowledge into visual tasks [Yu et al., 2017; Wang et al., 2022c; Brooks et al., 2023].

While BABEL [Punnakkal et al., 2021] and HumanML3D [Guo et al., 2022a] have relatively large vocabularies of actions, they contain a limited number of *simultaneous* actions. A single temporal segment is rarely annotated with multiple texts. For example, BABEL contains only roughly 2.5K segments with simultaneous actions, while it has ∼25K segments with only one action. This highlights the difficulty of obtaining compositional data at scale. Moreover, for any reasonably large set of actions, it is impractical to collect data for all possible pairwise, or greater, combinations of actions such that there exists no unseen combination at test time [Yang et al., 2018; Yu et al., 2017]. With existing datasets,

[4] https://golaem.com/content/doc/golaem-crowd-documentation/mixing-motions-different-body-parts

it is easy to learn spurious correlations. For example, if waving is only ever observed by someone standing, a model will learn that waving involves moving the arm with straight legs. Thus generating waving and sitting would be highly unlikely. In our work, we address this challenge by artificially creating compositional data for training using GPT-3. By introducing more variety, our generative model is better able to understand what is essential to an action like 'waving'.

Our method, SINC, extends the text-to-motion model TEMOS [Petrovich et al., 2022] such that it becomes robust to input text describing more than one action, thanks to our synthetic training. We intentionally build on existing models to focus the analysis on our proposed synthetic data. Given a mix of real single actions, real pairs of actions, and synthetic pairs of actions, we train a probabilistic text-conditioned motion generation model. We introduce several baselines to measure sensitivity to the model design, as well as to check whether our learned motion decoder outperforms a simpler compositing technique (i.e., simply using our GPT-guided data creation approach, along with a single-action generation model). We observe limited realism when compositing different body parts together, and need to incorporate several heuristics, for example when merging motions whose body parts overlap. While such synthetic data is imperfect, it helps the model disentangle the body parts that are relevant for an action and avoid learning spurious correlations. Moreover, since our motion decoder has also access to real motions, it learns to generate realistic motions, greatly improving the realism of the generated motions compared to the synthetic composition baseline.

Our contributions are the following: (i) We establish a new benchmark on the problem of spatial compositions for 3D human motions, compare a number of baseline models on this new problem, and introduce a new evaluation metric that is based on a motion encoder that has been trained with text supervision. (ii) To address the data scarcity problem, we propose a GPT-guided synthetic data generation scheme by combining action-relevant body parts from two motions. (iii) We provide an extensive set of experiments on the BABEL dataset, including ablations that demonstrate the advantages of our synthetic training, as well as an

analysis quantifying the ability of GPT-3 to assign part labels to actions. Our code is at https://sinc.is.tue.mpg.de for research purposes.

4.2 External linguistic knowledge & synthetic data

External linguistic knowledge. Large language models have been exploited for many visual tasks such as instruction-conditioned image editing [Brooks et al., 2023], visual relationship detection [Yu et al., 2017], and human-object reconstruction [Wang et al., 2022c], among others. Similar to us, Wang et al. [2022c] incorporate GPT by asking what body part is in contact with a given object, which in turn is used for image-based 3D human-object reconstruction. In contrast, we exploit GPT to extract knowledge about body parts that are involved in an action.

Our work in this Chapter is the first to systematically model such body part associations from textual descriptions. Following our approach, recent approaches use LLM-prompting to extract knowledge about contact between people [Subramanian et al., 2024] and use those contact labels in the loss functions to improve contact generation. In 3D human motion generation, FineMoGen [Zhang et al., 2023c] uses LLMs to produce edit texts and demonstrates promising results for part editing. CoMo [Huang et al., 2024] extracts the body parts involved in an action from its textual description — similar to our approach — but uses them as input to a diffusion transformer in order to focus on generating the motion of the body parts involved in the action. Iterative Motion Editing [Goel et al., 2024] relies on captioned motions, which are passed through an LLM along with a pre-defined set of "Motion Editing Operators" (MEOs) to detect which joints and frames should be edited. A pre-trained diffusion model is then used to inpaint these regions.

All the aforementioned works use LLMs as knowledge extractors to derive labels or guide their models. Beyond label extraction, LLMs also serve as language-based interfaces to image, video, and 3D generation tools, significantly improving accessibility. ChatPose [Feng et al., 2024] is one of the first works in this direction, unifying pose generation, pose estimation, and general LLM-based reasoning into a single framework.

More recently, ChatHuman [Lin et al., 2024] integrates multiple 3D human-related tasks into an LLM-centric model by invoking appropriate tools and adding another layer of language-driven understanding that effectively leverages the tool outputs. Such works demonstrate the potential of LLMs to be used in conjunction with current models for 3D pose and motion estimation or generation — a promising direction for the future of research in 3D human understanding.

Training with synthetic data. Using synthetic data to train machine learning models is a standard approach for solving many visual recognition tasks, such as 3D body pose estimation [Chen et al., 2016; Patel et al., 2021; Black et al., 2023], 2D body part segmentation [Varol et al., 2017], 3D hand pose estimation [Hasson et al., 2019], video action recognition [Varol et al., 2021], 2D body pose estimation [Rogez and Schmid, 2016] pedestrian detection [Pishchulin et al., 2011], and optical flow estimation [Ilg et al., 2017]. Furthermore, Tome et al. [2019] offer large-scale synthetic data for egocentric camera wearer pose estimation relying on MoCap data, and recently EgoGen closes the gap in egocentric synthetic data generation by generating synthetic data and improving on the task of ego-centric 3D human pose estimation [Li et al., 2024a]. In a similar spirit to us, HUMANISE [Wang et al., 2022b] creates a synthetic dataset of human-scene interactions by combining 4 action categories from BABEL [Punnakkal et al., 2021] with 3D scenes, and pairing them with language descriptions. In this Chapter, we generate synthetic training data by combining existing 3D motion assets and language labels to overcome the data scarcity problem for compositional learning, helping our method avoid learning spurious correlations.

4.3 Spatial composition of motions from textual descriptions

Given a set of action descriptions in the form of text, such as {“walk in a circle”, “wave with the right hand”}, and a desired motion duration F, the goal is to probabilistically generate realistic 3D human motions such

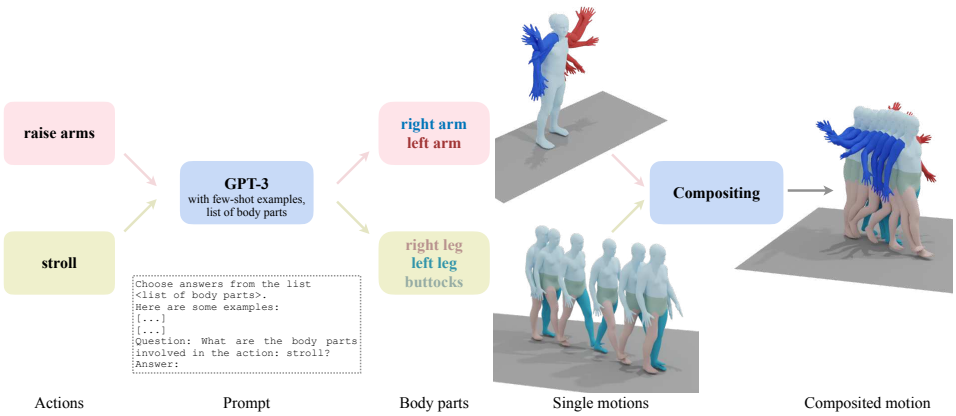


FIGURE 4.2: **GPT-guided synthetic training data creation:** We illustrate our procedure to generate Synth-Pairs. Here, we combine two motion sequences from the training set with the corresponding labels 'stroll' and 'raise arms'. We first prompt GPT-3 with the instructions, few-shot examples containing question-answer pairs, and "ask" for which body parts are included in an action –e.g., for the action stroll (bottom left). We minimally post-process the output of GPT-3 to assign this action to a set of body parts. The relevant body parts from each motion are then stitched together to form a new synthetically composited motion.

Body part labeling	Global	Torso	L. arm	R. arm	L. leg	R. leg	Mean
Part velocity magnitude	0.72	0.68	0.60	0.55	0.58	0.67	0.65
GPT-based (a) free-form	0.72	0.70	0.85	0.86	0.80	0.83	0.79
GPT-based (b) choose from list	0.79	0.68	0.89	0.90	0.88	0.89	0.84
GPT-based (c) list + few-shot	**0.84**	**0.72**	**0.89**	**0.89**	**0.89**	**0.90**	**0.85**

TABLE 4.1: **GPT body part labeling performance:** We report the part-labeling accuracy of GPT-3, as well as a simpler baseline based on part velocity magnitudes. For GPT-3, we experiment with various types of prompts on 100 manually annotated actions. (a) Asking which body parts are involved in an action, and post-processing free-form language outputs to associate to part labels. (b) Asking to choose from a given list of body parts, and (c) additionally also providing few-shot examples. See Section 4.3.1 for more details on these prompts.

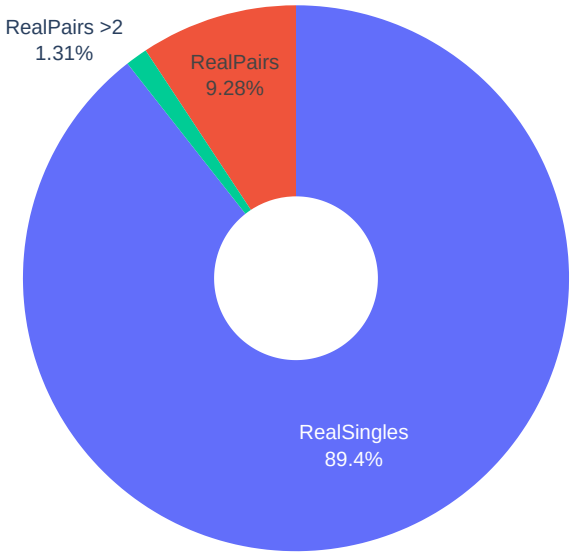


FIGURE 4.3: **Distribution of the training set:** The simultaneous Real-Pairs (the simultaneous actions, i.e., overlapping action segments, in BABEL) are the vast minority of the data, highlighting the importance of automatically enriching training data through our synthetic spatial compositions. Note that > 2 refers to triplets of simultaneous actions which, as expected, are a small minority.

that all the given actions are performed simultaneously in each generated sequence. We refer to this problem as spatial composition. Note that as a proof of concept, we perform our experiments mainly with pairs of actions, but the framework is applicable beyond pairs.

In the following, we first introduce our framework to generate synthetic training data by extracting correspondence between actions and body parts from large language models (Section 4.3.1). Then, we describe our model training with synthetically augmented data (Section 4.4.1), and finally present implementation details (Section 4.4.2). Note that for the rest of this Chapter, we denote the simultaneous actions from BABEL with **Real-Pairs**, the single-motion segments from BABEL with **Real-Singles**, and our synthetic data created by using body-part labels from GPT with **Synth-Pairs**.

4.3.1 GPT-guided synthetic training data creation

As explained in Section 4.1, we leverage a large language model, GPT-3 [Brown et al., 2020], to automatically assign a given action description to a set of body parts. After the extraction of body parts associated with an action, we synthetically combine existing motions together to create compositional training data. This process is illustrated in Figure 4.2 and explained further below.

Body part label extraction from GPT-3. We process the entire set of motion descriptions in the dataset to associate each action description to a set of body parts. We use the Text-Completion tool from OpenAI's API of GPT-3 [Brown et al., 2020] to extract the body part correspondence for a given language description. Specifically, for each individual action description in the dataset, we construct a prompt consisting of three parts. (i) We specify the instruction in the form of "choose answers from the list <list of body parts>', where the list is ['left arm', 'right arm', 'left leg', 'right leg', 'torso', 'neck', 'buttocks', 'waist']". (ii) We provide few-shot examples as question-answer pairs, where the question is 'What are the body parts involved in the action: <action>?', and the answer is the list of manually labeled body parts. (iii) The last part has the same form as the question, but we do not give the answer.

With this approach, GPT-3 outputs require minimal processing, i.e., the responses are words that correspond almost always to the provided list in (i). We post-process GPT-3's responses by removing punctuation, lowercasing, and mapping to a list of SMPL [Loper et al., 2015] body parts that we define separately, and use in the subsequent steps of our approach to generate synthetic data. We take a subset of SMPL body parts: ['left arm', 'right arm', 'left leg', 'right leg', 'torso', 'global orientation']. We coarsely define these six different body parts, but dealing with more finegrained body parts is certainly possible.

From the first list, 'neck' is mapped to 'torso', and ['waist', buttocks'] are mapped to 'global orientation'. This is because, when prompting for free-form outputs without providing a list (i) or few-shot examples (ii),

we qualitatively observe that GPT-3 refers to changes in global orientation of the body using words such as 'waist' or 'buttocks'. GPT-3 responses included 'global orientation' even in cases when it was not necessary, e.g., 'lift arm', 'raise leg'. Hence, we replace 'global orientation' with these two words instead. Finally, we include the label 'neck' in addition to 'torso', since GPT-3 tends to include 'neck' in its responses, especially when we prompt for the actions: 'look left' / 'look right'.

To evaluate our choices for the prompt, in Table 4.1 we measure the contribution of providing (i) the list, and (ii) few-shot examples in the prompt. For this, we manually label 100 action descriptions from BABEL. For each action, we annotate each body part as Yes/No/Sometimes to mark whether that body part is involved with that action. Note that we use 'Sometimes' for ambiguous cases, where it is acceptable to include, but not necessarily mandatory. For example 'hands' may or may not be involved in 'walking'. We then check the accuracy of GPT-3 body part labeling, by counting Yes/No as 1/0, ignoring optional body parts to not bias our evaluation.

A prompt asking for a free-form answer (i.e., "List the body parts involved in this action: <action>") complicates the required post-processing as one needs to handle over-detailed answers such as 'deltoids', 'triceps', or different ways of referring to the same body part. We manually built a lookup table to map from GPT-3 outputs to SMPL body parts but obtained suboptimal results. As can be seen from Table 4.1, providing the list (rows a vs b) significantly boosts the labeling accuracy, especially for picking the correct left/right arm/leg, which is further improved by providing few-shot examples (row c). We provide examples from GPT-3's responses for various prompts in Section 4.3.2.

4.3.2 Body part labeling with GPT-3

BABEL includes 6518 unique language labels for training and validation. We use these raw labels as input in the GPT-3 query. We prompt the public API https://openai.com/api/ for each of the BABEL action labels and automatically retrieve the body parts that are involved in the motion.

We experimented with various prompts before deciding on our final prompt template. We observed that GPT-3 outputs are easier to parse and map to our predefined list of body parts if we provide this list, as well as few-shot examples consisting of question-answer pairs. We use the following prompt, to extract the body part annotations for our synthetic data creation, as described in Section 4.3.1:

```
The instructions for this task are to choose
your answers from the list below:

left arm
right arm
left leg
buttocks
waist
right leg
torso
neck

Here are some examples of the question and answer
pairs for this task:

Question: What are the body parts involved in the
action of: walk forwards?
Answer: right leg
left leg
buttocks

Question: What are the body parts involved in the
action of: face to the left?
Answer: torso
neck

Question: What are the body parts involved in the
action of: put headphones over ears?
```

```
Answer: right arm
left arm
neck

Question: What are the body parts involved in the
action of: sit down?
Answer: right leg
left leg
buttocks
waist

Question: What are the body parts involved in the
action of: [ACTION]?
```

This shows the full prompt used to extract the annotations using GPT-3 for composing actions spatially. In Table 4.1, we quantitatively evaluated the body part labeling performance of this prompt, along with alternative prompts. Here, in Table 4.2, we provide qualitative examples to illustrate the behavior of GPT-3 to each of the prompt types. (a) "Free-form" prompt type contains only the final question. (b) "Choosing from a list" contains both the initial list to chose from and the final lines which only contain the question. (c) "Choosing from a list + Few-shot examples" refers to the full prompt. As shown in Table 4.2, using "Free-form" prompting requires a tedious post-processing of GPT-3 responses, since one needs a comprehensive mapping from all possible body part namings to our list. Moreover, the level of detail is not consistent across actions (e.g., 'left leg and hips' versus 'deltoid and triceps muscles'). We extract the associated body parts by detecting keywords from a manually constructed lookup table; however, the labeling accuracy based on Table 4.1 is still lower than instructing GPT-3 to choose from a list. We obtain further gains by including few-shot examples in the prompt. This is demonstrated qualitatively in Table 4.2 for the label 'rotate shoulders' which GPT-3 includes neck in addition to torso or 'walk backwards with arms attach to the waist' for which arms are mistakenly omitted for the "Choose

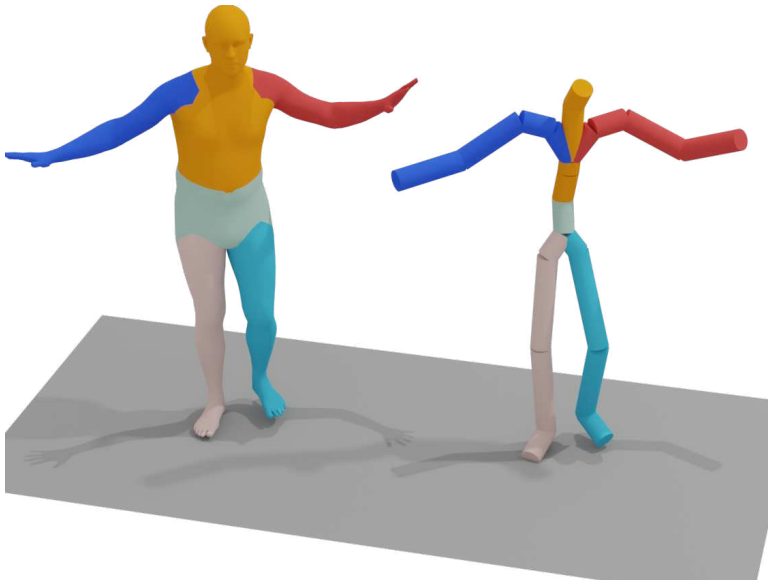


FIGURE 4.4: **Body parts:** Each color indicates a different body part. Vertices (left) and the skeleton (right) are extracted from the SMPL body model.

from a list" prompt. Our final prompt that provides both the list and few-shot examples perform best, while also requiring significantly less post-processing.

We use 6 body part labels based on common body segmentation maps [Loper et al., 2015], which we show in Figure 4.4. Since most of the AMASS dataset does not contain finegrained hand motions, we do not include hands as separate body parts. Furthermore, we observe that GPT-3 behavior may become inconsistent if we provide a long list of finegrained parts instead of few coarse labels. The main body parts include the right/left extremities, the torso-neck part, and the pelvis denoted as global. Global, except for the pelvis rotation, includes global translation of the body and it is used when either the waist or buttocks are included in GPT-3's response.

Could we extract body part labels without GPT-3? To test the effectiveness of our GPT-based body part labeling, we also implement an alternative body-part labeling approach based on part velocity magnitude. The assumption is that we have action-motion pairs, and if a

Action	Prompt Type	GPT-3 Response
move right arm in circular motion	Free-form	The person's right arm, shoulder, and possibly the upper part of their body.
	Choosing from a list	right arm
	Choosing from a list + Few-shot examples	right arm
make large circles with left leg in front of body	Free-form	The left leg and the hips.
	Choosing from a list	left leg
	Choosing from a list + Few-shot examples	left leg
overhead throw	Free-form	The deltoid muscle in the shoulder and the triceps muscle in the arm are moving when someone is doing an overhead throw.
	Choosing from a list	left arm right arm
	Choosing from a list + Few-shot examples	left arm right arm torso
walk backwards with arms attached to the waist	Free-form	The body parts involved in the action of walking backwards with arms attached to the waist are the legs, arms, back, and abdomen.
	Choosing from a list	right leg left leg buttocks
	Choosing from a list + Few-shot examples	left arm right arm left leg right leg waist
put down bottle with left hand	Free-form	Left arm Left hand Fingers
	Choosing from a list	left arm
	Choosing from a list + Few-shot examples	left arm torso
rotate shoulders	Free-form	The body parts involved in the action of rotating the shoulders are the neck, shoulders, arms, and back.
	Choosing from a list	left arm right arm arm torso neck
	Choosing from a list + Few-shot examples	left arm right arm arm torso

TABLE 4.2: **GPT response examples for different prompt types:** We show the responses of GPT-3 on some examples that demonstrate the differences between different prompt types. The output of the free-form prompt is non-trivial to parse and map to our list of body parts. On the other hand, providing the list and few-shot examples encourages GPT-3 to follow a more strict format and to describe the body parts with the same words as in our list.

body part movement is above a threshold, that part should be involved with the associated action. Specifically, we compute average positional velocities across frames for each body part, standardize (subtracting the mean, dividing by the standard deviation over frames), and determine a threshold (by visual inspection) to decide if a body part is involved in a given motion. This heuristic baseline has the disadvantage that it may suffer from spurious correlations (e.g., if we only see waving while walking, we will think that leg motion is critical to waving). From the first row of Table 4.1, we observe that the accuracy of this approach is significantly lower than the GPT-based approaches.

Body part composition to create new motions. Given a set of labeled motions to combine, and the extracted GPT-3 body parts involved, we first determine if the actions are compatible; i.e., whether a valid motion can be composited, based on the descriptions. For example, the actions ['walking', 'kicking with the right leg'] may not be performed at the same time as they both include the body part 'right leg'. For the synthetic training data, we only create compositions for valid pairs that are compatible in terms of their body part involvement, and use *real* motions from the database. Next, we detail the data creation procedure.

Given two motions A and B, along with the corresponding selected body parts extracted by GPT-3, we compose these motions into a new one by performing the following steps: (1) We trim the longer motion to match the length of the shorter one; (2) We order the motions A and B such that motion B always has fewer body parts than motion A; (3) If motion B involves at least one leg or the global orientation, we also select both legs, the global orientation, and translation from motion B (otherwise, we obtain these 4 values from motion A); (4) The remaining unselected body parts (if any) are taken from motion A; (5) The composited motion is obtained by combining selected body parts from motion A and B, along with the translation according to step 3. We perform step 3 to retain plausibility as much as possible, as the leg motions are highly correlated with changes in global translation and orientation. This procedure ensures

realism and accuracy of the compositions to some extent; but does not provide a guarantee.

Note that we also employ this approach as a baseline in our experiments, where we combine the motions under these assumptions using two *generated* motions from a single-action trained model. In this case, body part incompatibilities may occur ('walking' and 'kicking' both involve the leg), and body parts from motion B override the conflicting parts from motion A (see Section 4.4 for further details).

4.4 Synthetic data creation

We use GPT-3-guided spatial compositions in two parts of this Chapter. First, we use GPT-3 to benchmark how well a single-action baseline can perform, by applying composition as post-processing on independently generated motions (Figure 4.5 bottom). Secondly, we use GPT-3 to create synthetic data to train our model. In both cases, we employ the method described in Section 4.3.1. We use the heuristic of stitching the motion with fewer body parts (motion B) on top of the other motion (motion A), because the body parts of motion B are more likely to be local (as in "waving the right hand") and important for keeping the semantics of the motion. On the other hand, motion A is more likely to be a global motion (as in "walking" or "sitting") and grafting motion B onto motion A usually produces a realistic motion and preserves the semantics of both motions. Note that these heuristics were determined based on visual inspection over several examples, and may not be optimal.

The difference in the case of synthetic data creation is the compatibility test, which makes sure that no body part is involved in both of the motions being composited. Moreover, synthetic data combines existing real motions, and the single-action baseline combines generated motions.

We only apply the compatibility check for the synthetic data generation to avoid composing invalid motions, since a human physically cannot perform two actions with the same body part in most cases. This choice was made to ensure better synthetic data quality, as without it, the composition may be reduced down to one action (e.g., 'walking' would

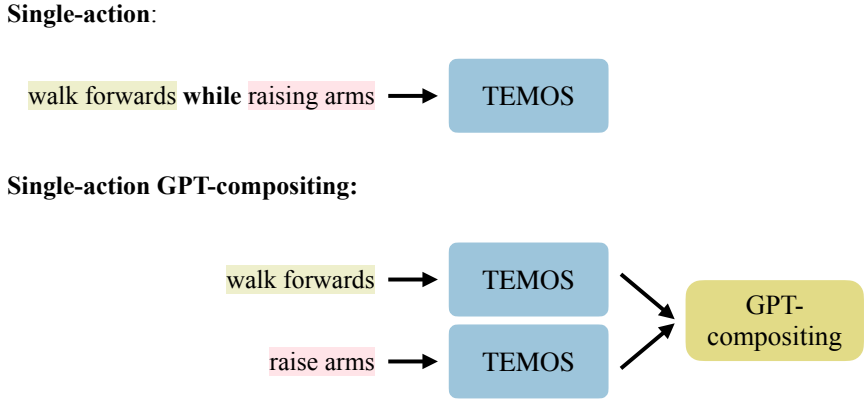


FIGURE 4.5: **Single-action baselines:** For both baselines, TEMOS is trained on Real-Singles (the single-motion segments) of BABEL. On the top, we concatenate the textual inputs by adding the word "while" in between actions. On the bottom, we generate the two actions independently and combine them with the body part guidance from GPT-3.

overwrite 'kicking' as the leg cannot do both). At test time, when we query 'walk' and 'kick with the right leg' with two different durations, SINC randomly generates one of the two actions, as seen in Figure 4.6.

4.4.1 Learning to generate spatial compositions

We employ the recent architecture TEMOS [Petrovich et al., 2022], which encodes the text into a distribution via a Transformer encoder (text encoder $\mathcal{T}_{enc}$), and produces motions by using a Transformer decoder (motion decoder $\mathcal{M}_{dec}$). Similar to Language2Pose [Ahuja and Morency, 2019], TEMOS contains a motion encoder ($\mathcal{M}_{enc}$) and encourages a cross-modal joint space between text and motion embeddings. A simplified overview of the architecture can be seen in Figure 4.7. At test time, the motion encoder is not used.

The motion encoder takes as input a body motion sequence $H \in \mathbb{R}^{l \times d_f}$ where d_f is the feature dimension and l the maximum motion length and outputs a single latent vector z^M and a distribution $\mathcal{N}(\mu^M, \Sigma^M)$. Similarly, the text encoder outputs z^T, which is sampled from the distribution $\mathcal{N}(\mu^T, \Sigma^T)$. These distribution parameters are obtained by appending two extra learnable tokens in the transformer encoder, and taking their

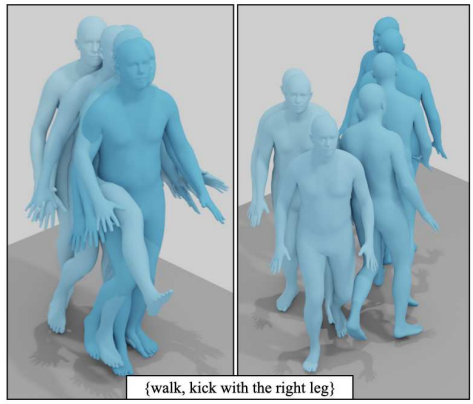


FIGURE 4.6: **Testing incompatible simultaneous actions:** We apply our model SINC on the input {'walk', 'kick with the right leg'} which represents an example of two incompatible actions due to involving the same body part 'right leg'. We display two random generations from our model, once with 2-second duration (left), and once with 4 seconds (right). We observe that SINC generates one of the two actions in each sample ('kick' on the left, 'walk' on the right).

corresponding outputs [Petrovich et al., 2021]. The latent vectors are sampled using the re-parametrization trick [Kingma and Welling, 2014]. The motion decoder then takes as input (a) the duration encoded by positional encodings $F \in \mathbb{R}^{l \times d}$, where l is the maximum motion length and d the latent dimension, (b) along with either the motion z^M or text z^T latent vector.

The model is supervised with the standard normal distribution losses:

$$\begin{aligned} \mathcal{L}_{KL}^{\mathcal{T}} &= \mathcal{KL}(\mathcal{N}(\mu^T, \Sigma^T), \mathcal{N}(0, I)) \\ \mathcal{L}_{KL}^{\mathcal{M}} &= \mathcal{KL}(\mathcal{N}(\mu^M, \Sigma^M), \mathcal{N}(0, I)) \end{aligned} \tag{4.1}$$

for the text and motion distributions, respectively. Moreover, $\mathcal{L}_{\mathcal{Z}} = \mathscr{L}(z^T, z^M)$ is used to force the text latent vectors to be close to the motion latent vector, where $\mathscr{L}$ is the smooth L1 loss. Finally, the distributions of different texts and the motion are supervised via $\mathcal{L}_{KL}^{\mathcal{M}\|\mathcal{T}} = \mathcal{KL}(\mathcal{N}(\mu^T, \Sigma^T),\ \mathcal{N}(\mu^M, \Sigma^M))$ and its symmetric version $\mathcal{L}_{KL}^{\mathcal{T}\|\mathcal{M}}$. The reconstruction losses for the generated motions, $\hat{H}^M$ and $\hat{H}^T$, from both the motion and the text branches, $\mathcal{L}_{\mathcal{R}} = \mathscr{L}(H, \hat{H}^T) + \mathscr{L}(H, \hat{H}^M)$, are

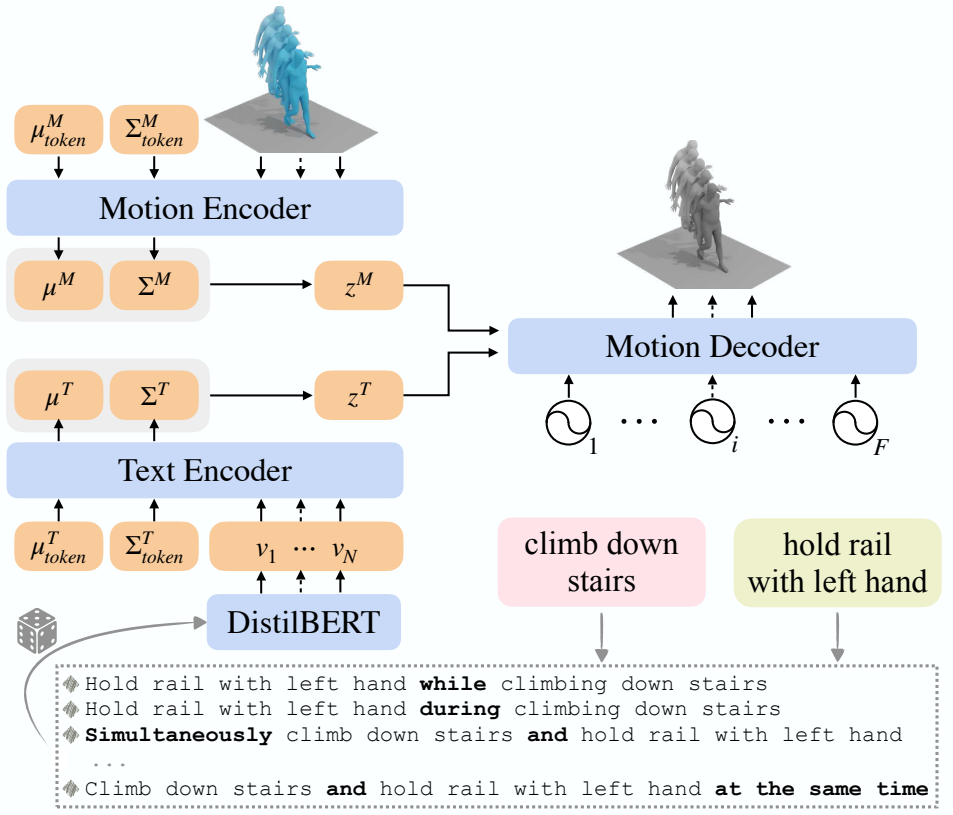


FIGURE 4.7: **Model architecture:** We extend TEMOS [Petrovich et al., 2022] such that it is trained with compositional actions. We build multiple descriptions given two action labels, by adding words such as 'while', 'during', etc. We then randomly sample one version during training as input to the text encoder.

added to the total loss:

$$\mathcal{L} = \mathcal{L}^{\mathcal{T}}_{\mathcal{KL}} + \mathcal{L}^{\mathcal{M}}_{\mathcal{KL}} + \mathcal{L}^{\mathcal{M}\|\mathcal{T}}_{\mathcal{KL}} + \mathcal{L}^{\mathcal{T}\|\mathcal{M}}_{\mathcal{KL}} + \mathcal{L}_{\mathcal{R}} + \mathcal{L}_{\mathcal{Z}}. \tag{4.2}$$

While our experiments use TEMOS [Petrovich et al., 2022], our synthetic data strategy is applicable to any text-to-motion generation model. We provide further evidence on the benefits of synthetic training on a diffusion-based approach (similar to MLD [Chen et al., 2023]) in Section 4.6.5.

Input text format and augmentations. Here, we describe how we provide the input to the text encoder. In case of a single motion that is described

by one action label, we simply input the original label as in [Petrovich et al., 2022]. In case of two or more descriptions, which is the focus of this Chapter, we combine multiple descriptions into a single text. Specifically, we use several keywords to describe simultaneous actions (e.g., 'while', 'at the same time', 'simultaneously', 'during', etc.), and randomly place them in the text description to form an input that imitates a free-form input. Moreover, we shuffle the order of the labels, and add inflections to verbs such as gerunds when grammatically applicable; e.g., when using 'while'. Figure 4.7 shows some examples. Such an input formation allows users to enter free-form language descriptions at test time, which is a natural interface for humans. During training, we pick a random text augmentation, and at test time, we evaluate all the models using the conjunction word 'while'. In Section 4.6.4, we provide results with more conjunction words both seen and unseen during training.

4.4.2 Implementation details

We define a 3D human motion as a sequence of human poses using the SMPL body model [Loper et al., 2015]. As in TEMOS [Petrovich et al., 2022; Holden et al., 2016], we represent the motion using the 6D rotations [Zhou et al., 2019] for body joints and the 2D-projection of the x, y trajectory along with the z translation. This results in $d_f = 135$ for each body pose in each motion sequence. All the motions are canonicalized to face the same forward direction and are standardized — the motion features are set to have mean zero and standard deviation one.

The input text is encoded with DistilBERT [Sanh et al., 2019] (whose parameters are frozen), followed by a learnable linear projection. The latent dimension is fixed to $d = 256$. We use 6 layers and heads in the transformers with a linear projection of size 1024. We set the batch size to 64 and the learning rate to $3 \cdot 10^{-4}$ for all our experiments.

Our model is applicable to arbitrary numbers of actions for a given motion. Therefore, we jointly train on single actions, and multiple actions. Single actions are from real data. Multiple actions can be (i) from synthetic pairs that are randomly generated 'on the fly' or (ii) from real data where

most such motions have two labels, but we also include those with more than two; see the video on our project page for more details[1]. For each sequence in a mini-batch, if it is a real single action, with probability p, we combine it randomly with another compatible action.

4.5 Experiments

We present data and evaluation metrics (Section 4.5.1), followed by the baselines we introduce (Section 4.5.3). We report quantitative experimental results with ablations (Sections 4.5.4 and 4.5.5). We conclude with a qualitative analysis (Section 4.6.6) and a discussion of limitations (Section 4.6.7).

4.5.1 Data and evaluation metrics

We use the **BABEL** dataset [Punnakkal et al., 2021], to exploit its unique potential to study simultaneous actions. Some BABEL motions come with multiple language descriptions, where annotations can overlap in time. We extract all such simultaneous action pairs for both training (2851 motions), and validation sets (1232 motions). We show the distribution of the simultaneous and single action segments in BABEL in Figure 4.3. We only consider the sequences that have a length between 600 (20 sec.) and 15 (0.5 sec.) frames. From the validation set, we exclude redundant pairs with the label 'stand', because this commonly occurs in the data while not representing challenging cases. We also remove pairs that are *seen* in the training set, and end up with 667 sequences that contain two simultaneous actions. The results on the full validation set are provided in Section 4.6.3. Besides the simultaneous pairs, we include the single-action data from BABEL in training. Specifically, there are 24066 and 8711 single-action motions for training and validation sets, respectively. As aforementioned, in our experiments, we denote the simultaneous actions from BABEL with **Real-Pairs**, the single-motion segments from BABEL with **Real-Singles**, and our synthetic data created by using body-part labels from GPT with

[1] https://sinc.is.tue.mpg.de

Synth-Pairs. We perform evaluation only on the real spatial pairs of the BABEL validation set to assess the quality of simultaneous action generation. We use the validation set as test set, given that BABEL test set is not publicly available. We train all of our models for 500 epochs.

We report evaluation metrics adopted by prior work [Ghosh et al., 2021; Petrovich et al., 2022], and work done in the previous Chapter 3: Average Positional Error (APE), and Average Variational Error (AVE). However, we observe that these metrics do not always correlate well with the visual quality of motions, nor their semantics. We introduce, and report results, using a new metric: **TEMOS score**. TEMOS score compares the cosine similarity between the generated motion and the ground truth after encoding them into the motion encoder of TEMOS [Petrovich et al., 2022], which is trained on BABEL Real-Singles (we do not observe significant changes when altering this model with TEMOS trained on different data, see Section 4.6.1).

This is similar in spirit to BERTScore [Devlin et al., 2019], which evaluates text generation quality by comparing to the ground truth in the text embedding space. More details can be found in Section 4.5.2. While this metric is also imperfect (e.g., it still assumes a single ground truth action), we observe that it better correlates with realism and motion semantics as it has been trained to encode motions controlled by text descriptions. An alternative performance measure is adopted by [Guo et al., 2022a] that reports motion-to-text retrieval metrics, randomly selecting for each motion 31 negative text descriptions along with the ground truth. Finally, we include diversity metrics in Section 4.6.2.

4.5.2 TEMOS Score

The position-based metrics typically used in this Chapter and prior work [Ghosh et al., 2021; Petrovich et al., 2022; Athanasiou et al., 2022] compare generated motions with the ground-truth motion in the coordinate space local to the body: they measure differences of positions and do not take into account semantics. Here are four types of examples where the metrics can fail: (1) with a cyclic motion such as “walking”,

the generation can be out of phase with the ground truth and still be semantically valid; (2) even for a non-cyclic motion such as "throwing an object", the timing can be different and can lead to bad scores on common metrics; (3) if the input text description is ambiguous such as "kick" (where the motion can be done from one leg or the other), the metrics may not reflect the quality of the generated motion; (4) if the motion demonstrates severe foot sliding or body translation artifacts, the error may be dominated by the translation error, effectively ignoring the overall implausibility of the limb motion e.g., feet not moving.

To avoid these issues, we introduce another performance measure called *TEMOS score*. We train a TEMOS model on BABEL Real-Singles for 1000 epochs, freeze its weights, and use its motion encoder component. Then, we extract features by feeding a motion B to the motion encoder, and use the mean of the distribution as the feature vector f. This feature captures the semantics of the motion as the motion space has been trained to explicitly model motion-text matching, i.e., cross-modal embedding space.

To calculate the TEMOS score, we feed the ground truth and the generated motions to the motion encoder, and extract the feature vectors f_{GT} and f_{motion}, respectively. Then we compute the score based on their cosine similarity as follows:

$$\text{TEMOS score}(f_{GT}, f_{motion}) = \frac{1}{2}\left(1 + \frac{f_{GT} \cdot f_{motion}}{||f_{GT}|| \cdot \ ||f_{motion}||}\right).$$

The range of this score is between 0 and 1, with a maximum at 1, which occurs when the two motions are identical.

4.5.3 Single-action baselines

In the following, we introduce and describe two baselines using a model trained with one description per motion: (i) A naive single-action baseline that relies on a text-to-motion synthesis model trained on single actions, tested on pairs of actions. (ii) Our proposed GPT-compositing applied on independent motion generations from a single-action model.

Model	Tr. Data		TEMOS ↑	Average Positional Error ↓				Average Variance Error ↓			
	R.-P.	R.-S.	score	root joint	glob. traj.	mean loc.	mean glob.	root joint	glob. traj.	mean loc.	mean glob.
Single-action	✗	✓	0.601	0.592	0.551	0.286	0.712	**0.076**	**0.075**	**0.013**	**0.083**
Single-action GPT-comp.	✗	✓	0.618	**0.546**	**0.507**	0.282	**0.666**	**0.076**	**0.075**	**0.013**	**0.082**
SINC-STE	✓	✗	0.614	0.636	0.615	0.275	0.743	0.082	0.081	0.014	0.090
SINC	✓	✗	0.631	0.703	0.682	0.269	0.815	0.107	0.106	**0.013**	0.114
SINC	✓	✓	**0.640**	0.601	0.573	**0.268**	0.724	0.093	0.092	**0.012**	0.100

TABLE 4.3: **Baseline comparison:** We train only with Real-Pairs of the BABEL dataset and report performance when compositing naively or with GPT-3 annotations. Furthermore, we ablate the model design for handling multiple textual inputs when extending TEMOS [Petrovich et al., 2022]. We observe better performance at handling action pairs with a single text encoder (SINC) that takes as input the two text labels as a single free-form description with various augmentations, as described in Section 4.3, compared to separate text encodings of the labels (SINC-STE). Moreover, we report the performance of SINC when adding Real-Singles, as well.

Single-action model. Our first baseline tests the ability of single-action models to synthesize compositions by only modifying the input text. We train with Real-Singles from BABEL. At test time, we concatenate the text descriptions using 'while' as a keyword and evaluate the generated motions.

Single-action GPT-compositing. Another single-action baseline generates two independent motions given two texts, which are then combined using our proposed GPT-guided composition, stitching body parts from two motions (as described in our synthetic data creation; see Section 4.3.1). Note that unlike the synthetic data, which combines real motions, this baseline combines generated motions. The disadvantage of this model is that it requires GPT at test time, and is based on heuristics that may be error-prone, such as trimming the motions to the same duration, and resolving common body part labels (see the video[2] on our project page for details[3]). In the presence of a model that is trained only on individual actions (Real-Singles), we observe that the GPT-based compositing of two

[2] https://www.youtube.com/watch?v=Si63Hlwru5c

[3] https://sinc.is.tue.mpg.de

independent generations improves the performance over the single-action baseline (as shown in Table 4.3 top). Based on qualitative observation (see Section 4.6.6), the single-action baseline often generates one out of the two actions. The GPT-compositing baseline better captures both actions; however, lacks realism due to composing actions with heuristics. SINC, which trains on compositional data, alleviates both issues. Although from Table 4.3 the GPT-compositing baseline outperforms SINC for the position-based metrics, i.e., APE, AVE, we observe that SINC outperforms all baselines for the TEMOS score. However, APE and AVE are much less reliable than TEMOS score for judging if a motion follows a fine-grained instruction, especially for our synthetically created data which are out-of-distribution for the validation set. Given the poor alignment of positional metrics with text descriptions, as explained in Section 4.5.2, we base our conclusion mostly on TEMOS score and include APE, AVE for consistency with prior work. Our synthetic data pipeline demonstrates its superiority further when being used from different models, as we show in Table 4.10. We refer to our video (5:02–5:55) for further comparisons between the different metrics[4]. Note that, an important advantage of our approach is that **we do not rely in body part labels during inference**, as the GPT-compositing baseline does, but learn from the synthetic data during training.

4.5.4 The effect of the input text format

To confirm whether our free-form input format sacrifices performance compared to a more controlled alternative of keeping the two action texts separate, we experiment with a variant of our SINC model by changing the text encoding. Instead of a single text combining two actions, we concatenate them together with a learnable separation token in between after independently encoding the actions with DistilBERT. We refer to this separate text encoding variant as SINC-STE. In Table 4.3, we compare SINC with SINC-STE when trained only with Real-Pairs, and observe a better TEMOS score with the free-form text augmentations, at the cost of

[4] https://www.youtube.com/watch?v=Si63Hlwru5c

Synthetic data	Training Data			TEMOS ↑	Average Positional Error ↓				Average Variance Error ↓			
	R.-P.	R.-S. %	Syn.-P. %	score	root joint	glob. traj.	mean loc.	mean glob.	root joint	glob. traj.	mean loc.	mean glob.
N/A	✓	0	0	0.631	0.703	0.682	0.269	0.815	0.107	0.106	0.013	0.114
	✓	100	0	0.640	0.601	0.573	0.268	0.724	0.093	0.092	0.012	0.100
Random composition	✗	0	100	0.539	0.489	0.434	0.291	0.595	0.075	0.074	0.012	0.082
	✗	50	50	0.540	0.587	0.535	0.288	0.687	0.077	0.076	0.012	0.083
	✓	0	100	0.619	0.485	0.438	0.272	0.602	0.074	0.073	0.011	0.081
	✓	50	50	0.617	0.454	0.394	0.272	0.560	0.069	0.068	0.011	0.075
GPT composition	✗	0	100	0.618	0.478	0.451	0.265	0.610	0.063	0.062	0.012	0.070
	✗	50	50	0.541	0.646	0.598	0.290	0.747	0.078	0.077	0.012	0.085
	✓	0	100	0.642	0.553	0.527	0.266	0.671	0.061	0.060	0.011	0.068
	✓	50	50	**0.644**	0.481	0.452	0.261	0.605	0.064	0.062	0.011	0.070

TABLE 4.4: **Contribution of the synthetic data:** We report performance when including two types of synthetic data created by body part combination, either determined by GPT or randomly. We further experiment (i) with different percentages of sampling ratios between the Real-Singles and Synth-Pairs, and (ii) with the inclusion of Real-Pairs.

worse positional errors. We observe that metrics based on joint positions may score high even in the absence of the second action, especially if it involves a finegrained motion (see video[5]). Besides quantitative performance, SINC has the advantage of allowing more flexible inputs.

4.5.5 Training with different sets of data

Contribution of Real-Singles and Real-Pairs. In Table 4.3, we report the performance of SINC when adding both Real-Pairs and Real-Singles to training. We see that training with the large number of single actions of BABEL, in addition to the small amount of action pairs, improves performance, and highlights the limited scale of the available pairs.

Contribution of GPT-guided Synth-Pairs. We experiment with different training sources in Table 4.4, mainly to assess the effect of adding synthetic training data. The percentages (0, 50, or 100) reflect the probability p that a real-single action is composited synthetically with another action (see Section 4.4.2). When using all training data (i.e., Real-P, Real-S 50%,

[5] https://www.youtube.com/watch?v=Si63Hlwru5c

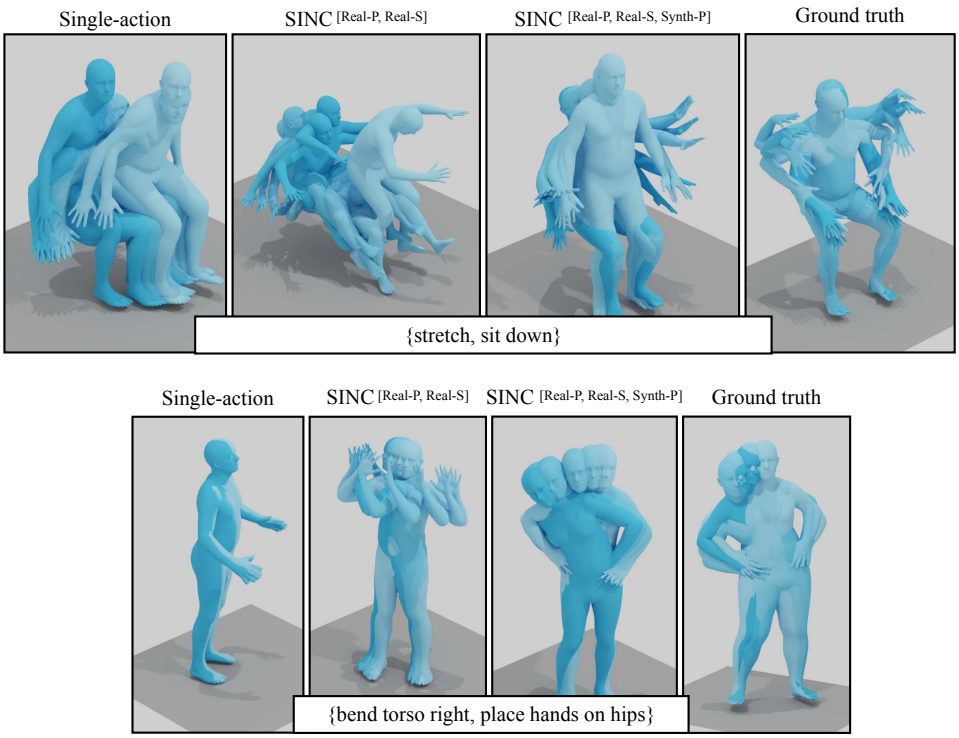


FIGURE 4.8: **Qualitative analysis:** We compare different models on two simultaneous action pairs. Both the Single-action model and the model not trained on synthetic data fail to generate those two compositions. Our model trained with the synthetic data successfully generates the composition in both cases. We include more comparisons in the video on our project page[7].

Synth-P 50%), we obtain the best TEMOS score, and more importantly observe better qualitative results (see Figure 4.8). In particular, the model trained with GPT-guided synthetic data demonstrates superior generalization capability to unseen combinations. In the video[6], we provide results with input combinations that are unseen both in the real training and validation sets.

Synthetic data without GPT guidance. We further test whether our GPT-guidance to generate synthetic data is better than just randomly mixing body parts (Random composition). In Table 4.4, GPT compositions outperform Random compositions, especially when training only on synthetic data (0.539 vs 0.618 TEMOS score).

[6] https://www.youtube.com/watch?v=Si63Hlwru5c

	Model used for TEMOS score	
	Single-action	**SINC**
Single-action	0.601	0.594
SINC	0.644	0.637

TABLE 4.5: **TEMOS score with various TEMOS models:** We report performance using different trained models to compute the TEMOS score. While the absolute score slightly differs when measured with a different model (e.g., 0.644 vs 0.637), the relative ranking of the models we compare remains the same.

4.6 Additional quantitative evaluation

We report quantitative results when evaluating with various conjunction words (Section 4.6.4), when using various TEMOS models to compute the TEMOS score (Section 4.6.1), when evaluating the diversity and multimodality metrics (Section 4.6.2), and, when evaluating on the full validation set for completeness (Section 4.6.3).

4.6.1 TEMOS score with various TEMOS models

As mentioned in Section 4.5.1, to report the TEMOS score, we use a TEMOS model trained on Real-Singles of BABEL. Since TEMOS score is a new metric, to ensure our results are consistent regardless of the training set of the evaluation model we justify our choice in Table 4.5. We choose to train it only on Real-Singles of BABEL without explicitly training with pairs. Hence, we analyze whether the choice of the TEMOS model has a large impact on the results when trained on pairs. In Table 4.5, we observe that the TEMOS score trend is similar when computed with TEMOS models trained on Real-Singles (Single-action) or on all real and synthetic data (SINC).

4.6.2 Diversity

Following Guo et al. [2022a], we report the overall diversity (for all action

	Div. →	Multimod. ↑
SINC	1.10	1.13
Real	1.34	-

TABLE 4.6: **Diversity evaluation:** We report the diversity and multimodality metrics of [Guo et al., 2022a] for our SINC model.

pairs), and multimodality (i.e., per-action-pair diversity) in Table 4.6. We report these numbers for completeness, as the setting of spatial compositions restrict the full body motion and provides much finer control than text-to-motion generation. We measure the L2 distance between the TEMOS embeddings of two sets of generations. For multimodality we sample 20 generations per description, and for diversity we generate 5 samples per description. Both metrics are computed for 300 random descriptions from the BABEL validation set. Real motions do not contain a sufficient number of motions for each action pair, thus the reason for omitting their multimodality. We observe that the diversity of SINC generated motions is close to the ones in the groundtruth set of pairs.

4.6.3 Full validation set

As explained in Section 4.5.1, we report all the results on a challenging subset of the validation set (i.e., without the action 'stand', and using only unseen examples). Here, we provide the results on the full validation set for completeness. In particular, we repeat the Tables 4.3 and 4.4, in Tables 4.7 and 4.8. When 'standing' actions are included in the full validation set, the TEMOS score for the baselines increase, but SINC still outperforms them in TEMOS score which is the most important metric. As expected, we observe slightly improved results overall on this 'easier' validation set and the conclusions remain similar to the comparisons in our original setting. We provide these results for completeness and to show that our conclusion on the original validation set remains consistent with the ones on our filtered validation set. The main change is that the differences in TEMOS score become more apparent.

Model	Tr. Data		TEMOS ↑	Average Positional Error ↓				Average Variance Error ↓			
	Real-P.	Real-S.	score	root joint	glob. traj.	mean loc.	mean glob.	root joint	glob. traj.	mean loc.	mean glob.
Single-action	✗	✓	0.607	0.516	0.483	0.262	0.626	0.067	0.066	0.012	0.073
Single-action GPT-comp.	✗	✓	0.626	0.458	0.431	0.244	0.569	0.068	0.067	0.011	0.074
SINC-STE	✓	✗	0.630	0.502	0.477	0.249	0.616	0.074	0.074	0.010	0.08
SINC	✓	✗	0.634	0.602	0.586	0.243	0.704	0.084	0.083	0.011	0.091
SINC	✓	✓	**0.645**	0.519	0.495	0.248	0.632	0.078	0.077	0.010	0.084

TABLE 4.7: **Baseline comparison on the full validation set of BABEL:** We observe similar trends with the filtered validation set reported in Table 4.3.

Synthetic data	Training Data			TEMOS ↑	Average Positional Error ↓				Average Variance Error ↓			
	Real-P.	Real-S. %	Syn.-P. %	score	root joint	glob. traj.	mean loc.	mean glob.	root joint	glob. traj.	mean loc.	mean glob.
N/A	✓	0	0	0.634	0.602	0.586	0.243	0.704	0.084	0.083	0.011	0.091
	✓	100	0	0.645	0.519	0.495	0.248	0.632	0.078	0.077	0.010	0.084
Random composition	✗	50	50	0.551	0.575	0.534	0.259	0.664	0.072	0.071	0.011	0.078
	✗	0	100	0.552	0.454	0.411	0.263	0.551	0.068	0.067	0.011	0.074
	✓	50	50	0.619	0.396	0.362	0.242	0.504	0.060	0.059	0.010	0.067
	✓	0	100	0.619	0.422	0.390	0.241	0.530	0.062	0.061	0.010	0.068
GPT composition	✗	50	50	0.554	0.641	0.604	0.262	0.731	0.074	0.073	0.011	0.081
	✗	0	100	0.632	0.424	0.405	0.237	0.543	0.055	0.054	0.011	0.062
	✓	50	50	**0.651**	0.418	0.397	0.234	0.533	0.055	0.054	0.010	0.062
	✓	0	100	**0.645**	0.472	0.453	0.237	0.581	0.053	0.053	0.010	0.060

TABLE 4.8: **Contribution of the synthetic data on the full validation set of BABEL:** We complement Table 4.4 of this Chapter, by reporting on the full validation set (without any filtering).

Conjunction Word	Seen in training	Model	TEMOS score ↑	Average Positional Error ↓				Average Variance Error ↓			
				root joint	glob. traj.	mean loc.	mean glob.	root joint	glob. traj.	mean loc.	mean glob.
`while`	✓	Single-action	0.601	0.592	0.551	0.286	0.712	0.076	0.075	0.013	0.083
		SINC	**0.644**	0.493	0.463	0.266	0.616	0.066	0.065	0.012	0.072
`during`	✓	Single-action	0.598	0.629	0.587	0.284	0.752	0.085	0.084	0.013	0.093
		SINC	**0.642**	0.497	0.471	0.261	0.622	0.065	0.063	0.012	0.071
`and ...at the same time`	✓	Single-action	0.599	0.607	0.568	0.283	0.722	0.084	0.083	0.014	0.092
		SINC	**0.643**	0.495	0.468	0.264	0.620	0.065	0.064	0.012	0.072
`in parallel`	✗	Single-action	0.600	0.611	0.570	0.294	0.736	0.081	0.081	0.012	0.089
		SINC	**0.643**	0.583	0.555	0.266	0.704	0.074	0.072	0.012	0.080
`whilst`	✗	Single-action	0.599	0.551	0.511	0.288	0.670	0.073	0.072	0.012	0.080
		SINC	**0.644**	0.491	0.461	0.262	0.614	0.066	0.065	0.012	0.072
`synchro-nously`	✗	Single-action	0.596	0.520	0.476	0.294	0.644	0.074	0.072	0.013	0.081
		SINC	**0.637**	0.520	0.492	0.261	0.644	0.0644	0.0632	0.011	0.070

TABLE 4.9: **Evaluation using different conjunction words:** In Table 4.3 of this Chapter, we evaluated the models with the conjunction word `while`. Here, we report performance when joining the two actions using other conjunction words, for both seen and unseen conjunction words during training. We observe similar trends for the TEMOS scores and the positional metrics as for using `while` to join the actions. Overall, performance of Single-action methods remains significantly inferior, especially for the TEMOS score. Note that SINC refers to our best model which is trained on both Real Singles, Real Pairs and Synthetic Pairs.

4.6.4 More conjunction words

In our previous experiments in this Chapter, we used `while` as our conjunction word. For completeness, in Table 4.9 we evaluate the Single-action method and our best model with other conjunction words at test time. We observe that the differences are minimal and the methods perform similarly across different conjunctions. This is true for all conjunctions both seen and unseen during training. The performance is similar, likely due to the text embeddings mapping the expressions to similar points.

Model	Synthetic training	TEMOS Score
MLD [Chen et al., 2023]	✗	0.612
MLD [Chen et al., 2023]	✓	**0.638**
TEMOS [Petrovich et al., 2022]	✗	0.640
TEMOS [Petrovich et al., 2022]	✓	**0.644**

TABLE 4.10: **Additional results with a diffusion model:** We report the performance of MLD [Chen et al., 2023] with and without adding the synthetic training data. We observe that synthetic data helps for both MLD and TEMOS.

4.6.5 Additional experiment with diffusion models

To complement our study with the TEMOS model [Petrovich et al., 2022], here, we provide an additional experiment by training a more recent state-of-the-art architecture for text-conditioned motion generation. Specifically, we implement Motion Latent Diffusion (MLD) [Chen et al., 2023] with the same text input pipeline as our method (see Section 4.4.1). Since MLD applies the diffusion on the latent space, we extract a single latent vector per motion (using the TEMOS model trained on Real-singles as a feature extractor). We train the diffusion model for 1000 epochs on 2 GPUs, with a batch size of 16, and learning rate of 1e-4. Instead of the coordinate-based representation of Guo et al. [Guo et al., 2022a], we directly train on 6D rotation representation (as is done for TEMOS, see Section 4.4.2).

Apart from those adaptations, we use the same architectural choices as in the original work [Chen et al., 2023]. In Table 4.10, we report the results with and without synthetic data, as we did for TEMOS with the rows 10 and 2 of Table 4.4, respectively. The same conclusion holds for MLD: the model trained on additional synthetic data demonstrates better performance than the one trained only on real data (Real-Pairs and Real-Singles).

4.6.6 Qualitative analysis

In Figure 4.9, we present simultaneous action generations using SINC for the validation set of BABEL. We show one random generation from our model for each description pair (left), along with the ground truth (right). Note that we display one sample due to space constraints, but the model can synthesize multiple diverse motions per input. We observe that, while being sometimes different from the ground-truth motion, our generations follow the semantics of *both* actions, achieving spatial compositionality. Moreover, we qualitatively compare different models trained with and without synthetic data in Figure 4.8, for the pair {'stretch', 'sit down'} and {'bend torso right', 'put hands on hips'}. This action pair combination is unseen in Real-Pairs, but is seen in the Synthetic-Pairs data. In both cases, the Single-action model and the model that has not been trained on Synthetic-Pairs (first two columns) fail to generate the motion in contrast to SINC which is trained on spatial compositions.

Finally, in Figure 4.10 we show failure cases of GPT-composition. Our baseline fails to generate a motion that corresponds to the instruction when the body parts are overlapping (top row). Another failure case happens when global orientation is important for the semantics of an action ('turn left') and is assigned to the walking action since it involves both feet (bottom row).

4.6.7 Limitations

Our framework relies on synthetic data creation by combining arbitrary motions together. Even if the body parts are compatible, in real life, not all actions appear simultaneously together. Future work should also explore the *semantic* compatibility between actions by extracting this knowledge from language models to construct semantically meaningful compositions. However, language models are also prone to mistakes. In particular, GPT-3 body part labels may be insufficient or ambiguous (e.g., 'walking' may or may not involve hands). Additionally, going beyond our 6 course parts to obtain finegrained body part label association is important. In particular, this could involve the fingers and even facial expressions.

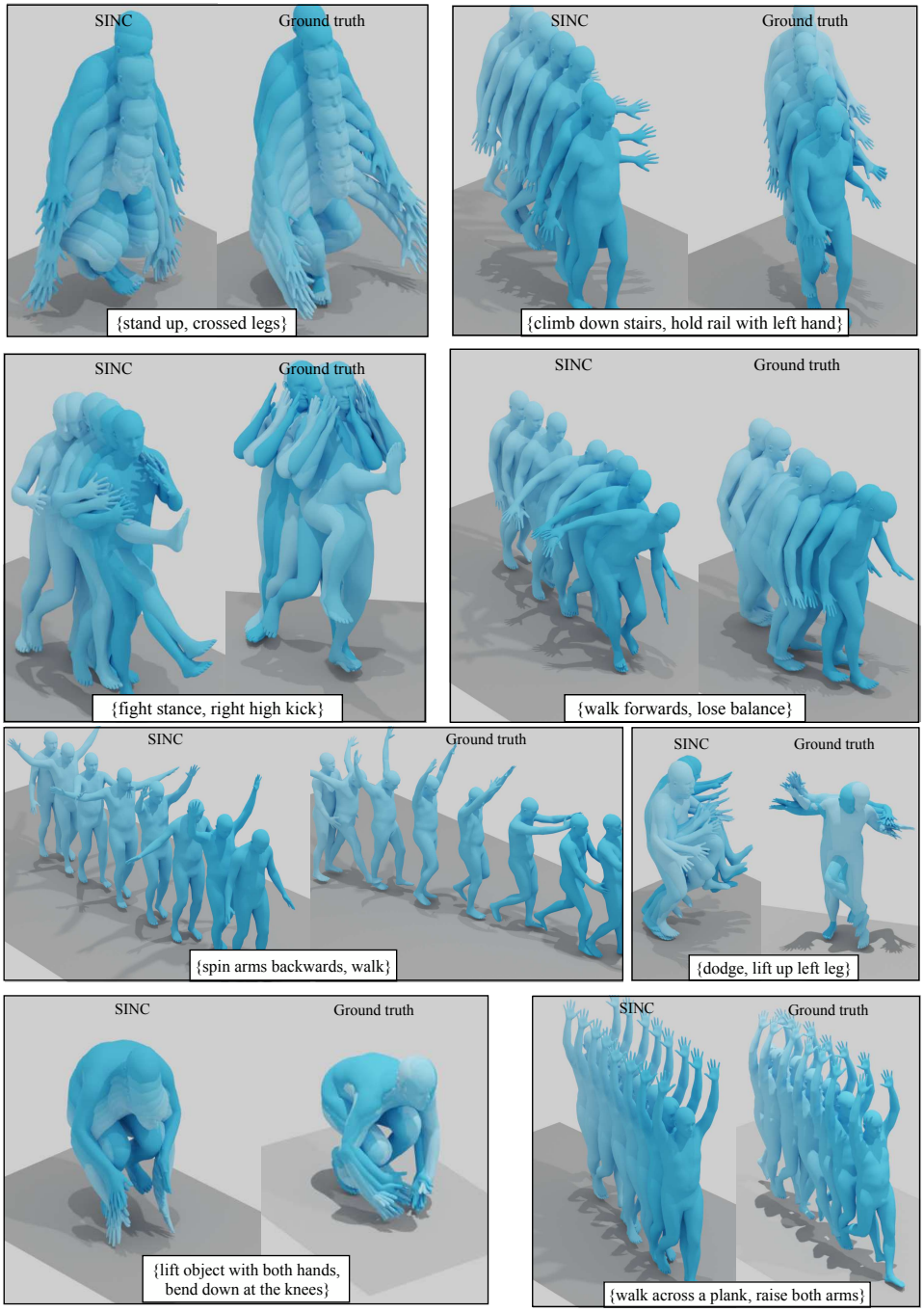


FIGURE 4.9: **Qualitative analysis:** (a) We present qualitative results for our final model, SINC, for various description pairs from the validation set. Our generations correctly correspond to the input semantics even when they are different from the ground truth, highlighting the challenge of coordinate-based (positional) performance measures. We display the ground truth (GT) for reference to define what the given actions mean.

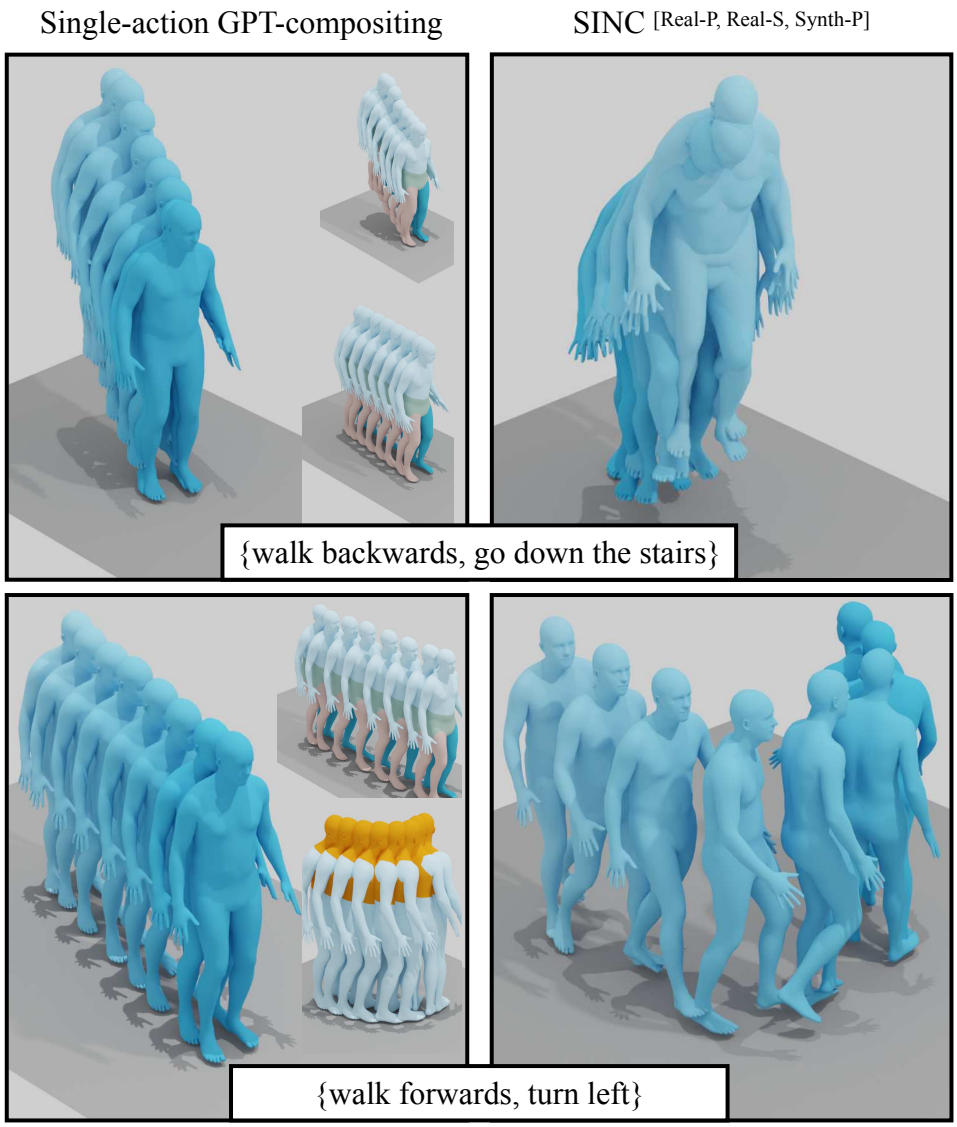


FIGURE 4.10: **Single-action GPT-compositing vs SINC:** We show two examples that highlight the advantage of our model compared to GPT compositions. Top: The detected body parts overlap causing the stitching to generate a forwards movement. Bottom: The global orientation is taken from the 'walk forwards' failing to generate a left turn.

Another limitation of the work done in this Chapter (and the whole field – see Section 2) concerns the evaluation metrics. Despite introducing a new TEMOS score, perceptually meaningful performance measures are still missing. Finally, our model is conceptually not limited to pairs, but since it is rare to simultaneously perform more than two actions, we mainly focus on pairs in this Chapter.

4.7 Conclusions

In this Chapter, we established a new method to create spatial compositions of 3D human motions. Given a set of textual descriptions, our SINC model is able to generate motions that simultaneously perform multiple actions presented as textual input. We make use of the GPT-3 language model to obtain a mapping between actions and body parts to automatically create synthetic combinations of compatible actions. We use these synthetic motions to enrich the training of our model and find that it helps it generalize to new, complex, motions. We introduce multiple baselines and experiment with different data sources for this new problem. Our findings will open up possibilities for further research in finegrained motion synthesis. While here we focus on spatial composition, future work should explore jointly modeling spatial and temporal action composition.

Chapter 5

Editing 3D Motions with Text

The focus of this chapter is 3D human motion *editing*. Given a 3D human motion and a textual description of the desired modification, our goal is to generate an edited motion as described by the text. The challenges include the lack of training data and the design of a model that faithfully edits the source motion. In this chapter, we address both these challenges. We build a methodology to semi-automatically collect a dataset of triplets in the form of (i) a source motion, (ii) a target motion, and (iii) an edit text, and create the new MotionFix dataset. Having access to such data allows us to train a conditional diffusion model, TMED, that takes both the source motion and the edit text as input. We further build various baselines trained only on text-motion pairs dataset, and show superior performance of our model trained on triplets. We introduce new retrieval-based metrics for motion editing, and establish a new benchmark using the evaluation set of MotionFix. Our results are encouraging, paving the way for further research on finegrained motion generation. Code, models, and data are available at https://motionfix.is.tue.mpg.de.

5.1 Introduction

Human motion control is an essential component of the animation pipeline, and involves creating a motion, as well as editing it until the motion matches the desired outcome. Text descriptions have emerged as one of the prominent ways to control motion generation [Petrovich et al., 2022; Tevet et al., 2023; Guo et al., 2022a; Zhang et al., 2023b]. However,

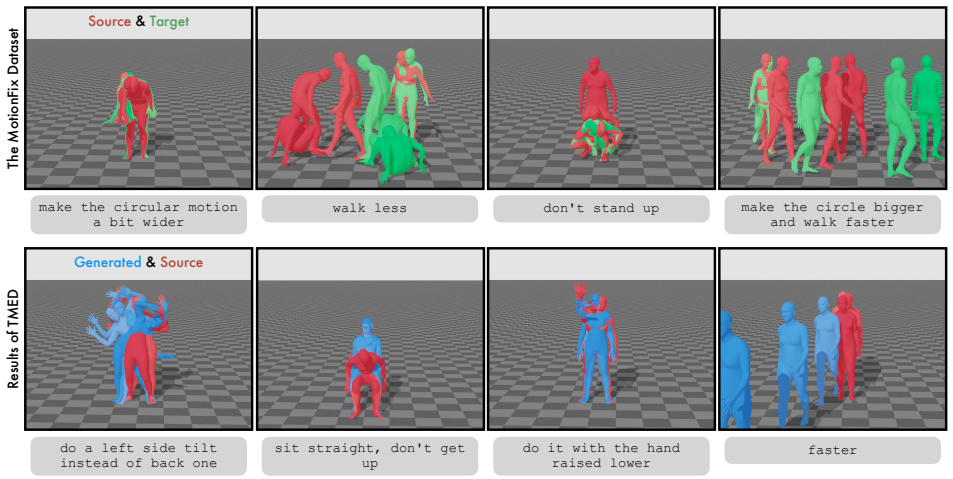


FIGURE 5.1: Our text-driven motion diffusion model (TMED) enables 3D human motion editing from natural language descriptions. To train this model, we introduce a semi-automatically collected dataset, MotionFix, that contains diverse types of edits such as modifying body parts, changing certain moments of the motion, or editing the speed or the style.

due to the inherent ambiguity in high-level language instructions, the resulting generation may not necessarily correspond to the motion one has in mind. An animator may then need to edit the motion further. Motion editing is non-trivial, arguably more complex than static pose editing, and may involve multiple types of instructions, such as changing the speed of a motion, modifying the repetitions for cyclic actions, adjusting the posture of a particular body part or modifying a certain temporal segment of a motion. In this Chapter, we aim, given an initial source motion and an edit description, to generate a new motion that follows the source motion and edits it according to the text instruction.

There are existing approaches that can modify body coordinates [Karunratanakul et al., 2023] and lower/upper limbs [Zhang et al., 2023b; Tevet et al., 2023]. However, they require manual selection of the body parts, which prevents making edits beyond local modifications such as the speed of the overall action. On the other hand, several methods have been proposed to provide more fine-grained control in text-to-motion generation. These include temporal control, such as Zhang et al. [2023a] and TEACH (in Chapter 3), as well as spatial control, shown in SINC

(Chapter 4) and its follow-ups [Zou et al., 2024; Huang et al., 2024]. These methods can be repurposed for editing, but would be limited to addition or subtraction of actions, as they can only edit the motion of a specific body part and are unable to edit the overall motion style and speed or edit a temporal segment of it. Given the above, all these works are limited to specific types of edits. Instead, our work considers unrestricted edits described by language instructions. To enable unrestricted motion editing, we collect a manually annotated dataset, MotionFix, that supports training generative models for the task of text-driven motion editing.

Constructing a dataset for 3D human motion edits is non-trivial. In contrast to text-based image editing, where methods such as InstructPix2Pix [Brooks et al., 2023] exploit large text-to-image generation models [Rombach et al., 2022] to automatically create training data, there exists no 3D motion generation model which generalizes faithfully to unrestricted text inputs. Besides, *dynamic* edits can be more complex than static image edits. In fact, PoseFix [Delmas et al., 2023] is a successful example of a 3D human body pose editing dataset; however, the differences between static poses are mapped to text in a rule-based manner using joint distances, and such a rule-based approach is not applicable to dynamic motions. To demonstrate this, we show examples of triplets from MotionFix dataset in Figure 5.1 (top row), illustrating the initial or source motion (in red) overlaid with the edited or target motion (in green) along with the edit text in the bottom. For example, as it can be seen in Figure 5.1 (top row), motion edits can refer to spatial arrangements of a motion (e.g., 'make the circular motion wider'), edits in a specific temporal segment of the motion (e.g., 'don't stand up') or edits referring to overall motion (e.g., 'make the circle bigger and walk faster'). Such generic edits cannot be deterministically described by rule-based approaches.

In this Chapter, we take a different route and mine existing motion capture (MoCap) datasets to find suitable motion pairs automatically, for which the differences are then manually described by typing text. By not relying on a generative model, we ensure motion quality; and by annotating the text (which is relatively fast), we achieve unrestricted edits.

The key challenge in our semi-automatic data curation pipeline is how

to find motion pairs that are similar enough for a meaningful and concise edit text to describe the difference. The difference between the source and target motions should not be too large to avoid the annotator typing a too-complex text or describing entirely the target motion, discarding the source. An overly complicated text describing every little subtle difference of joints and timing of motion would lead to very long text descriptions and would not be user friendly as it goes against the role of language as high level control. Furthermore, we want the text to implicitly refer to the source motion and not describe the target motion entirely, as this would make the source motion irrelevant. The motions should have similarities for the annotator to potentially make reference to the source. Our solution is to employ the recent TMR motion embedding space [Petrovich et al., 2023] that effectively captures semantics, as well as sufficient details for the body dynamics, thanks to its contrastively and generatively trained motion encoder. To form our candidate pairs for annotation, for each motion in a large MoCap collection [Mahmood et al., 2019], we retrieve the top-ranked motions according to their embedding similarity. Using crowdsourcing, we collect textual annotations for these pairs. The resulting dataset, MotionFix, is the first text-based motion editing dataset, which contains different types of edits as can be seen in Figure 5.1 (top). Some edits involve a specific body part (e.g., the hand should "make the circular motion a bit wider"), others alter the overall body dynamics (e.g., "make the circle bigger and walk faster" when walking in a circle).

MotionFix enables both training and benchmarking for this new task. We design and train a **T**ext-based **M**otion **E**diting **D**iffusion model, TMED, that is conditioned on both the source motion and the edit text. The results of our TMED model are encouraging as shown in Figure 5.1 (bottom), generating different types of edits. For example, the model can edit the overall spatial coordinates of a motion ("do a left side tilt instead of back one"), the way a motion is performed ("sit straight, don't get up"), parts of the body ("do it with the hand raised lower"), or the speed of a motion ("faster").

To benchmark our model and compare against baselines, we introduce new metrics on the evaluation set of MotionFix. Following the commonly

adopted retrieval-based metrics in text-to-motion generation benchmarks [Guo et al., 2022a], we perform motion-to-motion retrieval and check how often the ground-truth target motion is in the top ranks. We also report the ranking of the source motion to evaluate the proximity to the source. While this metric should not be too high – otherwise there would be no edit — it gives intuition about whether the generated motion deviates too much from the source. Our experiments demonstrate that our conditional model trained on triplets generates motions that are closer to the target, compared to strong baselines we build on top of state-of-the-art text-to-motion generation methods, which have only access to text-motion pairs for training.

Our contributions are the following: (i) We introduce MotionFix, the first language-based motion editing dataset. MotionFix provides motion-motion-text triplets annotated through our semi-automatic data collection methodology. This dataset enables both training and benchmarking for this new task. (ii) We introduce several baselines based on text-to-motion generation, which generate the motion of body parts that are supposed to be edited based on language models (similar to what we have done in Chapter 4). While our baselines achieve promising results, we show that models trained on text-motion pairs fall behind those trained on our triplets. (iii) We propose TMED, a diffusion-based model for motion editing given language instructions. We demonstrate both qualitatively and quantitatively that TMED outperforms all the baselines.

5.2 The new MotionFix dataset

The progress in controlling 3D humans with text has been driven by new datasets that pair 3D humans with language descriptions. In Table 5.1, we summarize the three most popular motion description datasets. KIT [Plappert et al., 2016] is the first source of such data While KIT, BABEL, HumanML3D enable text-to-motion generation training, they do not support editing. On the other hand, as can be next seen in Table 5.1, PoseFix [Delmas et al., 2023] provides pose editing triplets, but does not support *motion* editing. Our MotionFix dataset supports motion editing

Dataset	#motions	vocab.	label type
KIT-ML [Plappert et al., 2016]	3911	1623	motion description
BABEL [Punnakkal et al., 2021]	10881	1347	motion description, action
HumanML3D [Guo et al., 2022a]	14616	5371	motion description
PoseFix [Delmas et al., 2023]	6157 $\times$ 2	1068	pose editing
MotionFix (ours)	6730 $\times$ 2	1479	motion editing

TABLE 5.1: **Comparison with existing datasets:** MotionFix is the first dataset supporting the task of text-based motion editing.

training, while being at a similar scale to PoseFix in terms of the number of triplets and the vocabulary of edit texts.

Appropriate training data for text-based motion editing are in the form of triplets: source motions, target motions and edit texts. As discussed in Section 5.1, a big challenge in motion editing from language instructions is the lack of training data. To overcome this challenge, we design a semi-automatic data creation methodology. We first automatically construct candidate motion pairs that are similar (and different) enough, so the edit can potentially be described by language in simple words. We then ask annotators to manually type the edit text. In the following, we detail our procedure.

We make use of a motion embedding space to find motion pairs that are similar. Specifically, we employ a recent text-to-motion retrieval model TMR [Petrovich et al., 2023]. TMR is trained with a contrastive loss on the latent space of motions and texts, and reports state-of-the-art results for text-motion retrieval. We observe that such a model, by design, has the ability to produce latent motion representations that, for a given motion, ranks the semantically close ones nearby in the embedding space. We then use TMR to perform motion-to-motion retrieval. This is in similar spirit to using CLIP [Radford et al., 2021] for image similarity. We construct our dataset by finding such motion pairs from the AMASS MoCap collection [Mahmood et al., 2019].

From each motion in AMASS, we first extract TMR motion embeddings

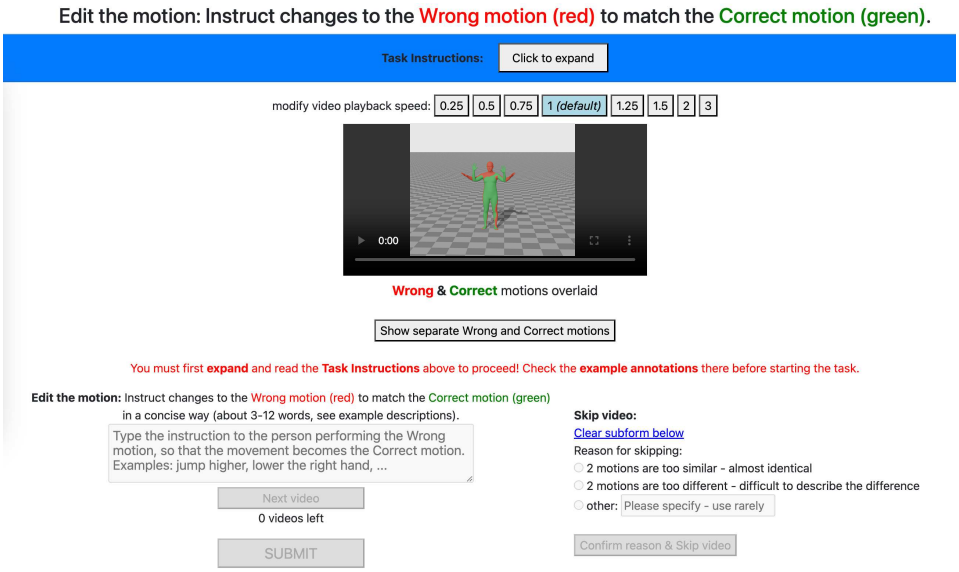


FIGURE 5.2: **Annotation interface used to collect MotionFix dataset:** We show the annotation interface used to collect the MotionFix after automatically retrieving candidate motion pairs suitable for editing. We explicitly restrict the annotators to label the pair with an edit text while avoiding overly detailed description of the differences with long and detailed texts. To further enforce this, we provide them with example of edits texts (i.e., 'jump higher', 'lower the right hand'.

with sliding windows of 3 to 5 seconds. We explicitly choose those durations as we observed that shorter sequences are not rich enough to retrieve different enough motion pairs, while longer sequences tend to be too different. In preliminary analysis, we found that using longer motion pairs reduces the probability of finding good candidates that differ by simple edits, while using shorter ones usually yields motion pairs that have a high probability to be almost identical. Then, we compute the pairwise embedding similarities and filter out all the motions pairs that have similarity $\geq$ 0.99 to avoid identical motions. We extract the top-2 most similar motions for a given motion and include these pairs in the annotation pool. We experimented with thresholding instead of following a top-k selection approach, but the TMR feature similarity is not well calibrated across motion pairs, which would make finding a constant threshold difficult. Finally, we align each motion pair to have the same initial translation and global orientation for the gravity axis to avoid labeling redundant edits that can be trivially created by changing the initial body translation and orientation.

Once we curate a list of candidate motion pairs, we give them to annotators from AMT. In the annotation interface, seen in Figure 5.2, along with the instructions, we give representative examples with multiple plausible edit texts for each motion pair. We allow the option to skip motion pairs if they are too similar (no difference to describe), or if they are too different (no easy way to describe the difference). Quantitatively, 7% and 55% of the pairs were considered too similar or too different, respectively. For the remaining pairs, we found that the majority are suitable candidates for editing.

We performed several rounds of data collection. After the initial round, we observed that some annotators tend to overanalyze the edit, which tends to describe the target motion alone or produce overcomplicated edits. Hence, after computing the statistics for a manually curated set of good annotations, we started encouraging the annotators to keep their edit texts around $3-12$ words, but no longer than 15 words. We explicitly request the annotators to refer to the source motion and encourage them to use words indicative of edit texts, e.g.,"instead", "higher/lower",

"same/opposite".

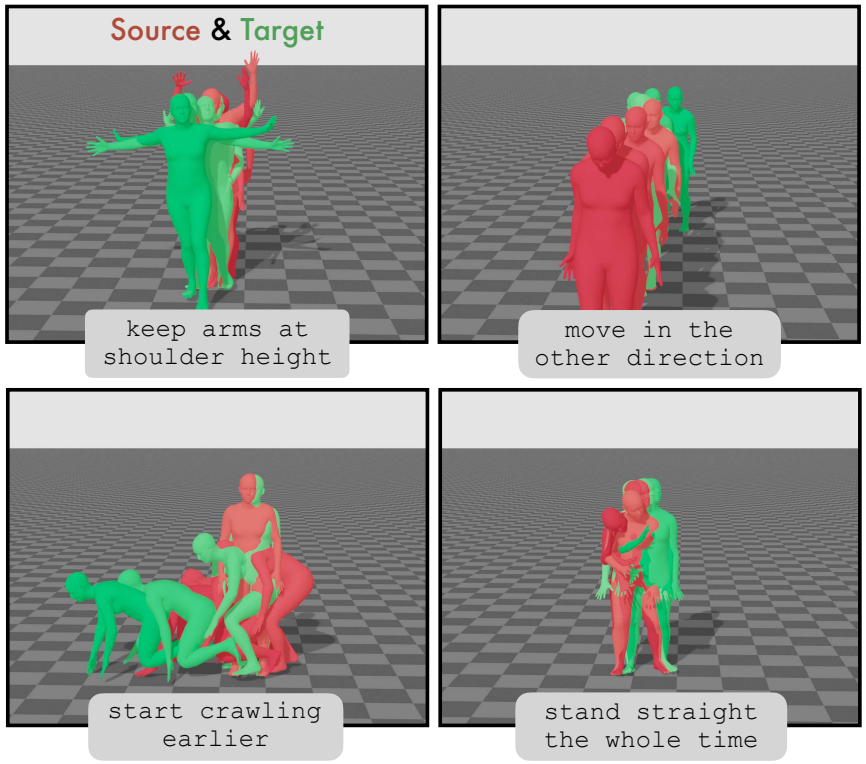


FIGURE 5.3: **Dataset samples:** We display source motions (red) overlaid with target motions (green) from our MotionFix dataset, together with their corresponding text annotations.

The resulting MotionFix dataset contains 6730 triplets of source-target motions and text annotations. We partition the data into train/validation/test splits randomly with 80%/5%/15% ratios, and obtain 5387/330/1013 triplets for each split, respectively. As shown in Table 5.1, in contrast to previous motion description datasets that provide text-motion pairs [Plappert et al., 2016; Guo et al., 2022a; Punnakkal et al., 2021], our dataset enables training for motion editing, by also including a source motion. MotionFix is similar in spirit to PoseFix [Delmas et al., 2023], but our labels describe the difference between dynamic motions, as opposed to static poses. Our dataset involves unrestricted edits, leading to different edit types such as spatial edits "throw from higher", temporal subtraction of actions "start standing not bent down", mixture of

both "bend down a bit more, stand up faster", and repetitions with adjustments of the whole body motion "do one more repetition and extend arms and legs wider apart". We include several visual examples in Figure 5.3 that show body part editing ("keep arms at shoulder height"), directional ("move in the other direction") and temporal ("start crawling earlier") changes. We provide dynamic video examples in our webpage (https://motionfix.is.tue.mpg.de), through our video[1] and the data exploration interface[2]. Detailed statistics regarding the texts and motions can also be viewed in Section 5.3.

5.3 MotionFix statistics

We provide additional information about our MotionFix dataset. Since our task is about *editing* rather than describing motion, it is important that the edits in our dataset contain phraseology of that kind. To confirm this, in Figure 5.4 we present qualitative and quantitative visualizations of the word statistics or our dataset's edit texts. Aside from words that describe body parts, we observe that many of the popular words have an inherently comparative nature. For example, words such as "keep", "instead", "slower", "faster", "closer", "earlier", "right", "left", are frequently used as shown in the word cloud visualization. Additionally, in Table 5.2 we provide statistics related to the annotated text describing a motion edit. Notice that MotionFix contains a wide spectrum of text descriptions of motion edits ranging from single word commands (e.g., "slower") to more elaborate texts that describe edits of multiple body parts at different time segments of the motion. Moreover, MotionFix contains 4649, 4165 unique source and target motions, respectively.

5.4 Text-driven motion editing diffusion model

We introduce TMED, a text-driven motion editing diffusion model. Given a short 3D human motion, a textual instruction describing a modification,

[1] https://www.youtube.com/watch?v=cFa6V6Ua-TY
[2] https://motionfix.is.tue.mpg.de/explore.php

FIGURE 5.4: **Word frequencies in the MotionFix dataset:** On the top, we display a word cloud for the text annotations in the dataset. The most frequent words appear in larger fonts. Examples of such words are 'hand', 'arm' referring to body parts, 'instead' referring to the source motion, 'higher', 'lower', 'opposite', 'slower' referring to spatial, directional or speed edits. On the bottom, we show the histogram of the 30 most frequent words in the data.

Text in MotionFix triplets	
Total #triplets	6730
#Unique texts	5992
#Unique words	1479
Avg #words per text	8.5
Median #words per text	8.0
Std of #words per text	4.9
Min #words per text	1
Max #words per text	43

TABLE 5.2: **Statistics of the MotionFix textual data:** There are relatively low number of duplicate texts (given 6730 triplets and 5992 unique texts). The vocabulary is diverse (1479 unique words) and the average number of words per text (8.46) has a good trade-off between conciseness and expressiveness.

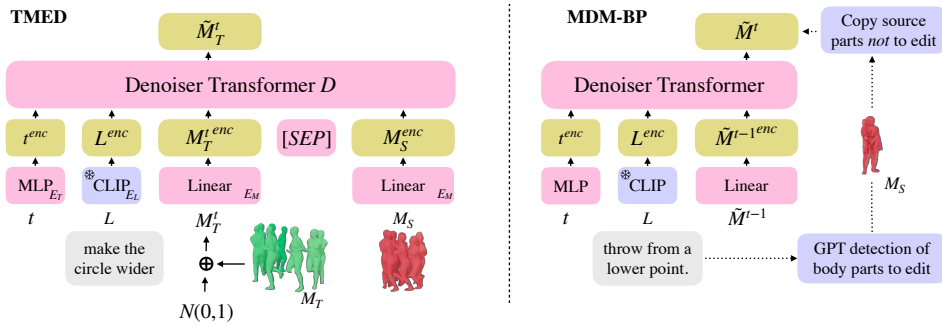


FIGURE 5.5: **Models overview:** (left) We illustrate our TMED model during training. We noise the target motion for t steps, and the transformer model is trained to denoise it back by one step. The conditions – text and source motion – are appended to the input. The CLIP backbone is frozen, while components denoted in pink are learned during training. At test time, the iterative diffusion process is initialized from random noise instead of the noised target. (right) Our MDM-BP baseline is repurposed from a pretrained text-to-motion generation model to be used only at test time for motion editing. The model is initialized from random noise and the body parts not to be edited according to GPT are copied from the source motion.

and a noise vector to enable randomness, the model generates an edited motion. Similar tasks have been addressed in the image domain for text-based image editing [Brooks et al., 2023], from which we take inspiration for our model design. We further build on the Motion Diffusion Model (MDM) [Tevet et al., 2023] that takes only text as input and generates a motion. In contrast, our model has an additional condition on the source, thus requires a different training dataset (as described in Section 5.2). In the following, we present the components of our TMED model.

5.4.1 3D human motion representation

We use a sequence of SMPL [Loper et al., 2015] body parameters to represent a human motion. SMPL is a linear function that maps the shape, and pose parameters of J joints, along with the global body translation and orientation, to a 3D mesh. The joint positions, J_p, can be obtained from vertices via the learned SMPL joint regressor. Following previous work [Petrovich et al., 2024] that discards the shape parameters, we set the shape parameters to zero (mean shape), since motion is parameterized primarily by pose parameters.

Various alternative representations have been used based on joint positions with respect to the local coordinate system of the body [Guo et al., 2022a; Holden et al., 2016; Starke et al., 2019]. Unlike prior work [Tevet et al., 2023; Guo et al., 2022a; Zhang et al., 2023b] that fits SMPL bodies to skeleton generations, we aim to enable direct regression of SMPL parameters, bypassing the need of a costly post-processing optimization [Bogo et al., 2016], and thus making our method ready to use for animation frameworks.

A common approach for representing SMPL pose parameters within a learning framework is to employ 6D rotations [Zhou et al., 2019], and to apply first-frame canonicalization for motions, as in [Petrovich et al., 2022] and the work in the previous Chapter [Athanasiou et al., 2022]. Similarly, we canonicalize our motions prior to training, so that all face the same direction in the first frame and have the same initial global

position. Inspired by [Holden et al., 2016; Petrovich et al., 2024], we represent the global body translation as differences between consecutive frames. Supervising with such relative translations helps the denoiser to generate better trajectories, as we observed unsmooth generations when using the absolute translation. Similar to STMC [Petrovich et al., 2024], we factor out the z-rotation from the pelvis orientation and separately represent the global orientation as the xy-orientation and the z-orientation as the differences between rotations in consecutive frames (resulting in 12 features, i.e., 6D representation for xy and z). We represent the body pose with 6D rotations [Zhou et al., 2019]. Similar to [Petrovich et al., 2022], we exclude the hand joints as they mostly do not move in the datasets we use. We additionally append the local joint positions after removing the z-rotation of the body [Holden et al., 2016; Petrovich et al., 2024] (resulting in 192 dimensional features with 6×21 for rotations and 22×3 for joints including the root joint). Thus, each motion frame has a dimension $d_p = 207$, consisting of 3 features for the global translation, 12 for the global orientation, and 192 for the body pose. The motion is represented as a sequence of the pose representations. During training, all features are normalized according to their mean and variance over the training set.

5.4.2 TMED conditional diffusion model

To learn TMED, we use our new training data, where each data sample comprises a source motion M_S, target motion M_T, and a language instruction L. We train a conditional diffusion model that learns to edit the source motion with respect to the instruction. We design a model similar to that of InstructPix2Pix [Brooks et al., 2023], where the generation from a random noise vector is conditioned on two further inputs S^e and M_S. Here, instead of a sequence of image patch tokens, the motion modality is represented as a variable-length sequence of motion frames. The noised target motion, the text condition S^e, and the source motion condition M_S are all fed as input to the denoiser at every diffusion step.

Diffusion models [Sohl-Dickstein et al., 2015] learn to gradually turn random noise into a sample from a data distribution by a sequence of denoising autoencoders. This is achieved by a diffusion process that adds noise ϵ_t to an input signal, M_T. We denote the noise level added to the input signal by using t, the diffusion timestep, as a superscript. This produces a diffused sample, M_T^t. The amount of noise added at timestep $t = 1, \ldots, N$ is defined a-priori through a noise schedule. We train a denoiser network, to reverse this process given the timestep t, the instruction S^e, the noised target motion M_T^t and the source motion M_S. As supervision, the output of the denoiser network $\tilde{M}_T^t$ is compared against the ground-truth denoised target motion M_T. Our model is therefore trained to minimize:

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(0,1), t, S^e, M_S} \left\| D\left(M_T^t; t, L, M_S\right) - M_T \right\|^2 \tag{5.1}$$

We use standard mean-squared-error as the loss function to compare the diffusion output with the ground-truth target motion. We choose to predict the denoised target motion, as we found this to produce better results visually than predicting the noise itself.

The architecture overview is illustrated in Figure 5.5 (left). Our model consists of multiple encoders for each input modality (E_T for timestep, E_L for the text S^e, and E_M for motion) and a transformer encoder D that operates on all inputs. The timestep t is encoded via E_T similar to MDM [Tevet et al., 2023], by first converting into a sinusoidal positional embedding, and then projecting through a feed-forward network (consisting of two linear layers with a SiLU activation [Elfwing et al., 2018] in between). As in [Tevet et al., 2023], we use the CLIP [Radford et al., 2021] text encoder for E_L. We pass the source and noised target motions through a linear layer (E_M), shared across frames, and obtain $M_S^{enc} = E_M(M_S)$ and $M_T^{t\ enc} = E_M(M_T^t)$. Given the variable duration of source and target motions, we add a learnable separation token `SEP` in between [Devlin et al., 2019] when appending them (so that the information on when the target motion ends and the source motion starts is communicated to the transformer). Once all encoded

inputs have the same feature dimensionality d, they are combined into a single sequence to be fed to the transformer, as shown in Figure 5.5, and sinusoidal positional embeddings are subsequently added. During training, to enable classifier-free guidance, the source motion condition is randomly dropped 5% of the time, the text condition 5%, both conditions together 5%, and all the inputs are used 85% of the time. For sampling from a diffusion model with two conditions, we apply classifier-free guidance with respect to two conditions: the input motion M_S and the text instruction S^e. We introduce separate guidance scales s_{M_S} and s_L that allow adjusting the influence of each conditioning.

For simplicity, we now abuse the notation by dropping the timestep subscripts when deriving the sampling process. Our generative model, TMED, learns the probability distribution over the target motions, M_T, conditioned on the source motions and text condition, $P(M_T \mid M_S, S^e)$. Expanding this conditional probability gives:

$$\begin{aligned} P(M_T \mid M_S, S^e) &= \frac{P(M_T, M_S, S^e)}{P(S^e, M_S)} \\ &= \frac{P(S^e \mid M_S, M_T) P(M_S \mid M_T) P(M_T)}{P(S^e, M_S)}. \end{aligned} \tag{5.2}$$

As in the original diffusion, we formulate this as a score function optimization problem by first taking the logarithm of Eq. (5.2):

$$\begin{aligned} \log(P(M_T \mid M_S, S^e)) = &\log(P(S^e \mid M_S, M_T)) \\ &+ \log(P(M_S \mid M_T)) \\ &+ \log(P(M_T)) \\ &- \log(P(S^e, M_S)). \end{aligned} \tag{5.3}$$

Then, the derivative with respect to the input of Eq. (5.3) gives the score estimate $\tilde{e}_\theta(M_T, s_{M_S}, s_L)$, learned under classifier-free guidance:

$$\begin{aligned} \nabla_{M_T} \log(P(M_T \mid M_S, S^e)) = &\nabla_{M_T} \log(P(S^e \mid M_S, M_T)) \\ &+ \nabla_{M_T} \log(P(M_S \mid M_T)) \\ &+ \nabla_{M_T} \log(P(M_T)). \end{aligned} \tag{5.4}$$

Finally, from Eq. (5.4), we sample from TMED using the modified score estimate of two-way conditioning a diffusion model as:

$$\begin{aligned}\tilde{e}_\theta(M_T, s_{M_S}, s_L) = e_\theta(M_T, \emptyset, \emptyset) \\ + s_{M_S} \cdot (e_\theta(M_T, M_S, \emptyset) - e_\theta(M_T, \emptyset, \emptyset)) \\ + s_L \cdot (e_\theta(M_T, M_S, S^e) - e_\theta(M_T, M_S, \emptyset)).\end{aligned} \tag{5.5}$$

To determine the effect of each condition based on Eq. 5.5, we further perform a sensitivity analysis on the guidance scales (s_{M_S} and s_L corresponding to the source motion and edit text) which control the generation at test time in Section 5.5.

Implementation details. All models are trained for 1000 epochs using cosine noise schedule with DDPM scheduler [Ho et al., 2020]. We use $N = 300$ diffusion timesteps, as we find this is a good compromise between speed and quality. The guidance scales are chosen for each model based on their best performance in the validation set of MotionFix($S_L = 2, s_{M_S} = 2$). We follow the same process for training MDM [Tevet et al., 2023] baselines described in the next section. In terms of architectural details, the dimensionality of the embeddings before inputting to the transformer is $d = 512$. We use a pre-trained and frozen CLIP [Radford et al., 2021] with all 77 token outputs of the ViT-B/32 backbone [Dosovitskiy et al., 2021] as our text encoder E_L. We use the text masks from E_L to mask the padded area of the text inputs. The motion encoder E_M that precedes the transformer is a simple linear projection with dimensionality $d_p \times d$, where the feature dimension of each motion frame is $d_p = 207$ (as described in Section 5.4.1).

5.5 Experiments

We start by describing our evaluation metrics for the new MotionFix benchmark (Section 5.5.1). We then present the main results on this task, comparing our proposed model to our baseline designs (Section 5.5.2). Next, we provide ablations on the training data size and guidance hyperparameters (Section 5.5.4). Moreover, we demonstrate

qualitative results, comparisons with the baselines and samples from our dataset (Section 5.5.7). Finally, we include a discussion on limitations (Section 5.5.8).

5.5.1 Evaluation metrics

Similar to text-to-motion synthesis, distance-based metrics to evaluate motion generation quality are problematic due to multiple plausible ground-truth motions for a given text. Prior work has extensively used text-to-motion retrieval metrics for evaluating text-to-motion synthesis [Guo et al., 2022b; Tevet et al., 2023], by training a text-motion contrastive model and using its features. To evaluate motion editing, we introduce motion-to-motion retrieval metrics. Given a generated motion, we measure how well the source (**generated-to-source retrieval**) or the target motion (**generated-to-target retrieval**) can be retrieved. We use TMR [Petrovich et al., 2023] as the feature extractor, but train it ourselves to support our feature representation, using the same regime as in the original paper with HumanML3D data [Guo et al., 2022a]. We report standard metrics, R@1, R@2, R@3 and AvgR using a gallery size of 32 randomly sampled batches for retrieval from the test set. Recall at rank k (R@k) computes the percentage of times the correct motion is among the top k results. Note that we fix the batches, so there is no randomness across evaluations. The performance is averaged across batches. We report results using the full test set as gallery, instead of 32 motions used in the batch metrics, in Tables 5.5 and 5.6, where the same conclusions hold. While the main performance measure is according to generated-to-target retrieval, we also monitor how close our generations remain to the source. As indicative values, we provide ground-truth (GT) values for the latter. We provide additional measures (FID, L2) and perceptual studies in Sections 5.5.5 and 5.5.6, respectively, which further confirm the results of our proposed retrieval-based benchmarking.

5.5.2 Comparison to baselines

We report our main results in Table 5.4. We compare the performance of TMED trained on our MotionFix triplets against several baselines trained on the larger, HumanML3D text-motion pairs [Guo et al., 2022a]. We build our baselines by training MDM [Tevet et al., 2023] with our human motion representation and by repurposing this model for motion editing, described next.

We first introduce two simple baselines: (a) **MDM** that purely uses the edit text as input to the text-to-motion generation (i.e., without a source motion), and (b) **MDM**$_S$ that additionally uses the source motion as input instead of noise during inference. For the latter, we also investigated reducing the number of diffusion steps when initializing from the source; however, we observed performance drops and therefore kept the full 300 diffusion steps. Inspired by synthetic data creation in Chapter 4, we design two additional strong baselines (**MDM-BP**$_S$ and **MDM-BP**), that are based on body part labels extracted by querying GPT with the edit texts. We automatically detect body parts that are irrelevant to the text and keep them constant via masking. We again initialize the diffusion process either from the source motion (MDM-BP$_S$) or from noise (MDM-BP) for the body parts that need to change according to the GPT response. For more details on the query and example GPT outputs, Section 5.5.3.

5.5.3 GPT-4 based annotation in MDM-BP

For the MDM-BP baseline described in Section 5.5.2, in order to automatically extract which body parts to edit, we prompt GPT-4 through its publicly available API (`https://openai.com/api/`, version: gpt-4-turbo). We use a language model to provide an automated approach for text-based motion editing, given the absence of text-based motion datasets and methods. Since prior methods, dealing with motion editing, use text-to-motion models and rely on manual selection of body parts to be edited [Zhang et al., 2023b; Tevet et al., 2023], we establish a baseline using a state-of-the-art text-to-motion model, MDM Tevet et al., 2023. Our approach is similar to the one followed in Chapter 4 to create synthetic

data for spatial compositions by detecting involved body parts in actions, but here we use the edit text to extract the body parts that need to be edited. We feed the edit text to the LLM and prompt it to provide the body parts that should be edited in order to achieve the given motion edit. We use the same body parts as in Chapter 4, seen in Figure 4.4 and give them to GPT-4 as the list to choose from. We experimented with various prompts and eventually used the following:

`You will be given an edit text that is supposed to be used to edit a motion. Your task is given the text to determine what are the parts of the motion that should change based on that edit text. The instructions for this task are to choose your answers from the list below: left arm, right arm, left leg, buttocks, waist, right leg, torso, neck. Here are some examples of the question and answer pairs for this task:`
`Question:` `What are the body parts that should be edited in the motion if the edit text is:` *`faster?`*
`Answer:` `right leg, left leg, buttocks, left arm, right arm, torso, neck.`
`Question:` `What are the body parts that should be edited in the motion if the edit text is:` *`do it with the opposite leg?`*
`Answer:` `right leg, left leg, buttocks.`
`Question:` `What are the body parts that should be edited in the motion if the edit text is:` *`stop moving in the end?`*
`Answer:` `right leg, left leg, buttocks, left arm, right arm, torso, neck.`
`Question:` `What are the body parts that should be edited in the motion if the edit text is: [EDIT TEXT].`

We qualitatively observe that by giving some examples in the prompt, GPT-4 is able to provide better responses. In Table 5.3 we present examples of edit texts and the respective GPT-4 responses.

We first observe from Table 5.4 that, for all the baselines, initializing from noise performs better than initializing from source motion. Our strong baselines based on body-part detection (MDM-BP, MDM-BP$_S$) clearly outperform the naive baselines. However, all baselines fall behind

Edit Text	Body Parts to Edit
spread your legs more as you jump	right leg, left leg
Instead of using both hands use one hand	right arm, left arm
instead of completing throwing motion bring hands close as if to pray	right arm, left arm
don't swing from side to side	left leg, right leg, buttocks, torso
turn around	right leg, left leg, buttocks, right arm, left arm, torso, neck
Sit little bit slower	right leg, left leg, buttocks
bend knees more start swinging left	right leg, left leg, buttocks
move faster in the end add back tilt	right leg, left leg, buttocks, left arm, right arm, torso, neck
step forward sooner and reach straight ahead	right leg, left leg, buttocks, torso
oppose arms that started movement	right arm, left arm
reach further back and lower with the same hand	right arm, torso
raise your hand faster and a little higher	right arm, left arm
jump slower and only four times	right leg, left leg, buttocks

TABLE 5.3: **Example GPT responses:** We show several input-output pairs for the MDM-BP baseline that uses automatically extracted body parts from edit text. GPT reasonably identifies the body parts that need to be edited e.g., 'turn around' involves the whole body, 'raise your hand faster and a little higher' involves both arms as this cannot be inferred purely from text.

Methods	Data	Source input	generated-to-target retrieval				generated-to-source retrieval			
			R@1	R@2	R@3	AvgR	R@1	R@2	R@3	AvgR
GT	n/a	n/a	100.0	100.0	100.0	1.00	74.01	84.52	89.91	2.03
MDM	HML3D	✗	4.03	7.56	10.48	15.55	2.62	6.15	9.38	15.88
MDM$_S$	HML3D	✓, init	3.63	7.06	10.08	15.64	2.62	6.25	9.78	15.84
MDM-BP$_S$	HML3D	✓, init&BP	38.10	48.99	54.84	6.47	60.28	69.46	73.89	4.23
MDM-BP	HL3D	✓, BP	39.10	50.09	54.84	6.46	61.28	69.55	73.99	4.21
TMED	MotionFix	✓, condition	**62.90**	**76.51**	**83.06**	**2.71**	**71.77**	**84.07**	**89.52**	**1.96**

TABLE 5.4: **Results on the MotionFix benchmark (test set):** We first evaluate several variants of our text-to-motion synthesis baseline (MDM) on the motion editing task. Subscript *S* denotes models that denoise the source motion initialization (init) instead of starting the diffusion from noise. BP indicates GPT-based body part labeling described in Section 5.4 to mask the source body parts which are kept unchanged during diffusion. Our model TMED effectively learns how to utilize the source motion conditioning, thanks to the MotionFix training data. See text for detailed comments.

our TMED, which successfully leverages the access to training triplets, and significantly outperforms alternatives.

Moreover, MDM-BP, MDM-BP$_S$ are both strong baselines, but relying on GPT body part labels might not capture all edit types, such as the ones that require modifying the overall body. We demonstrate this further in our video[3] from the project webpage[4] and our qualitative comparisons (Section 5.5.7).

5.5.4 Ablations

In the following, we investigate the effect of training data size and the guidance scales on the TMED model performance.

Training data size. In Table 5.7, we present the performance of TMED for different data sizes from MotionFix. We clearly observe, that increasing the data size has a large impact on the performance, justifying our data collection. The non-saturated trend is encouraging to scale up the training further.

[3] https://www.youtube.com/watch?v=cFa6V6Ua-TY

[4] https://motionfix.is.tue.mpg.de

Methods	Data	Source input	generated-to-target retrieval				generated-to-source retrieval			
			R@1	R@2	R@3	AvgR	R@1	R@2	R@3	AvgR
GT	n/a	n/a	64.36	88.75	95.56	1.74	20.83	33.66	40.47	33.13
MDM	HumanML3D	✗	0.00	0.30	0.49	476.65	0.10	0.10	0.10	486.26
MDM$_S$	HumanML3D	✓, init	0.00	0.10	0.10	477.68	0.00	0.10	0.30	485.02
MDM-BP$_S$	HumanML3D	✓, init&BP	8.39	14.81	18.36	181.38	30.11	36.72	40.77	107.11
MDM-BP	HumanML3D	✓, BP	8.69	14.71	18.36	180.99	30.21	36.82	40.47	106.05
TMED	MotionFix	✓, condition	**14.51**	**21.72**	**28.73**	**56.63**	**22.41**	**34.45**	**40.57**	**31.42**

TABLE 5.5: **Results on the MotionFix benchmark using the whole test set as a gallery:** We evaluate the models in Table 5.4 of this Chapter on the full test set of 1013 samples (as opposed to a random subset of 32). While the retrieval metrics are unsurprisingly lower due to a larger gallery size, the conclusions hold.

Methods	generated-to-target retrieval				generated-to-source retrieval			
	R@1	R@2	R@3	AvgR	R@1	R@2	R@3	AvgR
GT	64.36	88.75	95.56	1.74	20.83	33.66	40.47	33.13
10%	2.57	3.75	5.03	259.73	3.75	5.33	7.40	213.17
50%	8.00	14.71	18.36	104.85	13.03	20.73	25.07	77.08
100%	**14.51**	**21.72**	**28.73**	**56.63**	**22.41**	**34.45**	**40.57**	**31.42**

TABLE 5.6: **Text-based motion editing benchmark on MotionFix test set with different training data sizes.** We observe that the performance increases significantly when more data are used during training.

Guidance hyperparameters. In Figure 5.6, we present how TMED performs across different guidance values for both conditions. x-axis controls the text guidance s_L, and y-axis controls the source motion guidance s_{M_S} at test time. We report both generated-to-target (left) and source-to-target (right) R@1 retrieval results. We observe that there needs to be a balance between the two guidance values, and that performances decrease towards the extremes (e.g., top left and bottom right corners of the plots, where only one of the two conditions have higher guidance). This highlights the need to rely on both conditions to perform the task.

Methods	generated-to-target retrieval				generated-to-source retrieval			
	R@1	R@2	R@3	AvgR	R@1	R@2	R@3	AvgR
GT	100.0	100.0	100.0	1.00	74.01	84.52	89.91	2.03
10%	19.25	30.65	38.71	8.92	22.98	37.50	45.97	7.50
50%	47.08	61.49	69.66	4.23	54.44	70.06	78.12	3.33
100%	**62.90**	**76.51**	**83.06**	**2.71**	**71.77**	**84.07**	**89.52**	**1.96**

TABLE 5.7: **Effect of training data size in MotionFix:** We observe significant performance improvement as we increase the amount of training data.

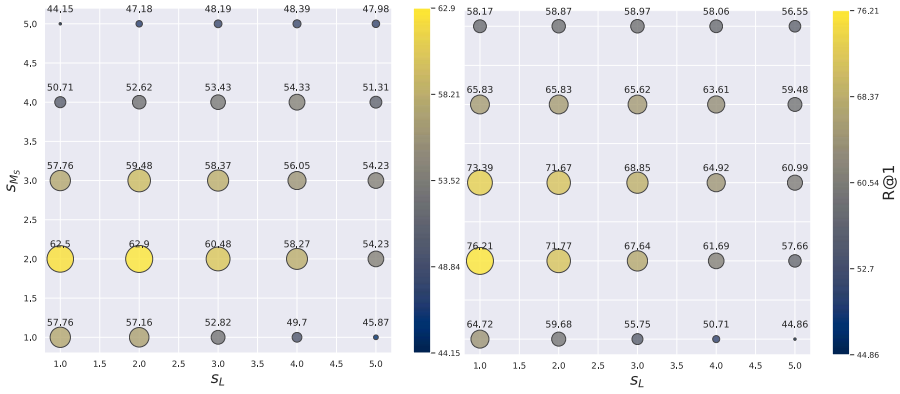


FIGURE 5.6: **Guidances of conditions:** We illustrate the R@1 performance of TMED for generated-to-target (left) and generated-to-source (right) retrieval benchmarks for $s_L, s_{M_S} \in [1, 5]$.

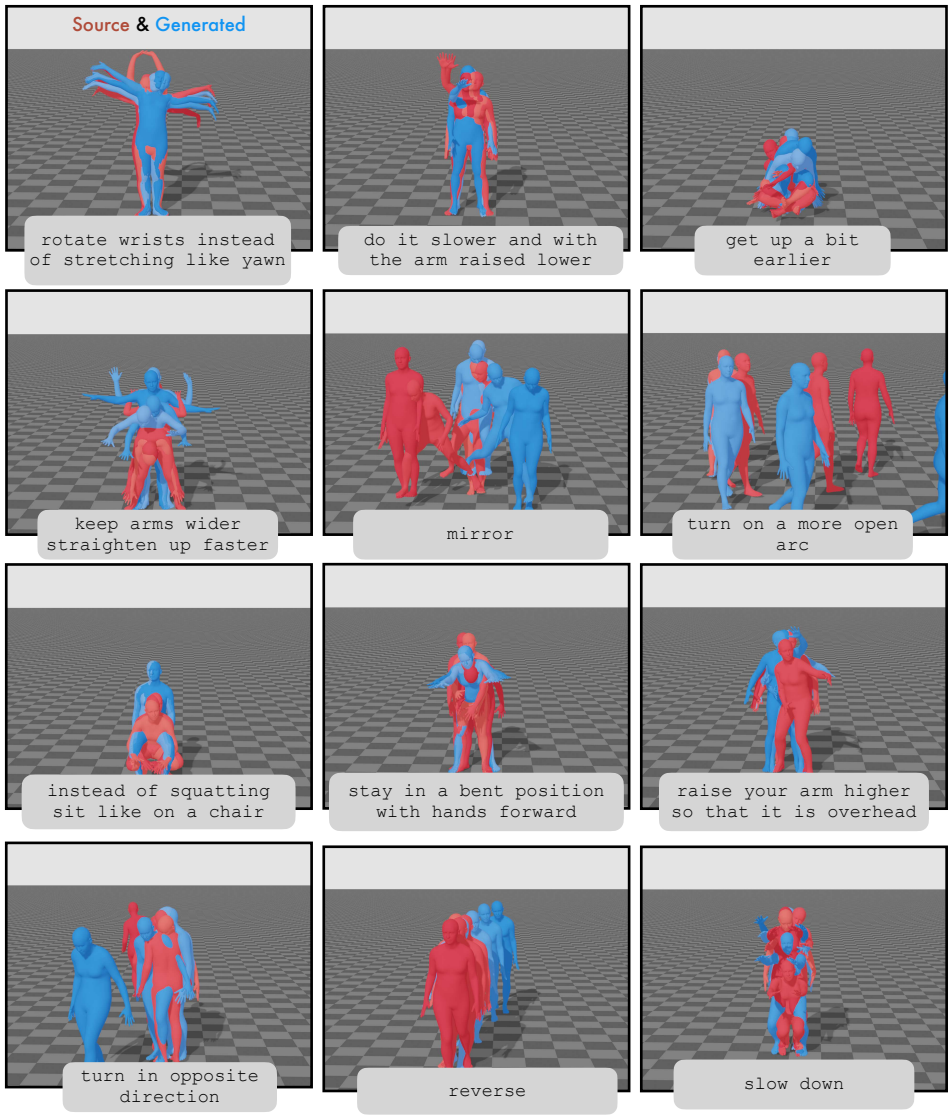


FIGURE 5.7: **TMED generations:** We illustrate several generations from our model with overlaid source (red) and generated (blue) motions. We present a variety of test cases ranging from elaborate edits (first example in top left) to short commands (e.g., "mirror"). TMED is able to perform both edits that describe temporal (e.g., "slow down") or spatial (e.g., "raise your arms higher so it is overhead") modifications.

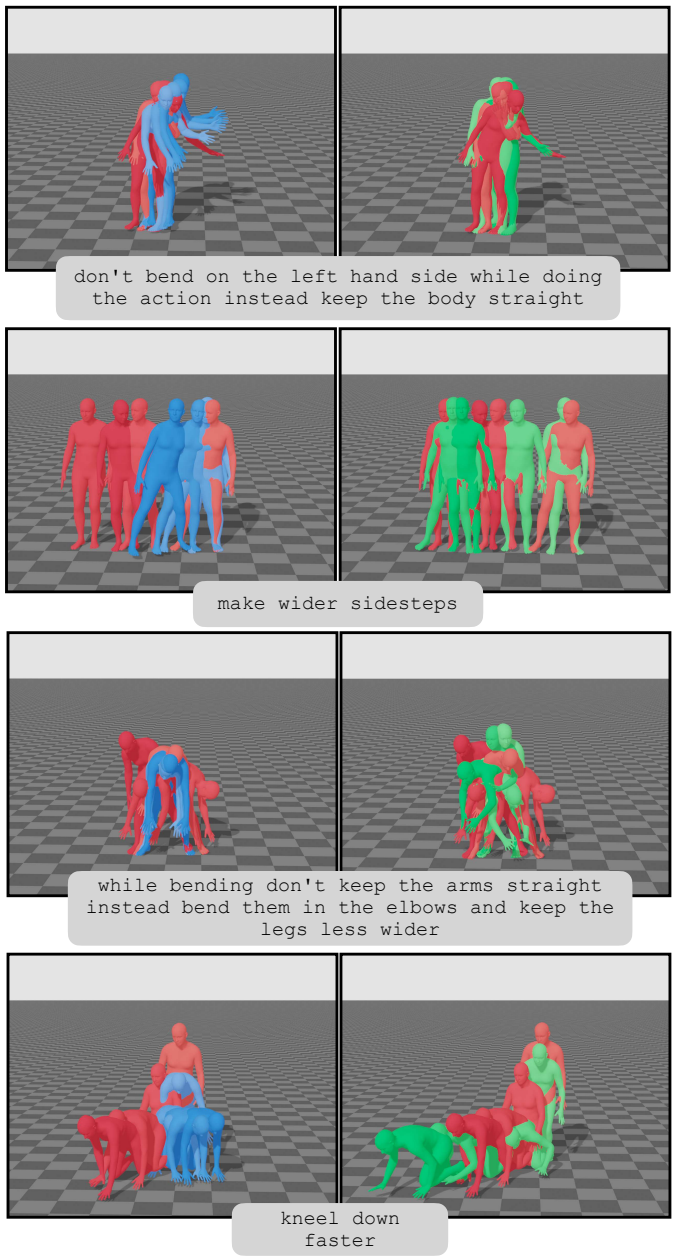


FIGURE 5.8: **Failure cases:** We show four failure examples from our model. For each sample, we provide the source motion (red) overlaid both with the generation (blue, left) or the ground-truth target motion (green, right). In the top row, we observe that the model may fail to generate the edited motions when the edit text is detailed and the motions differences are subtle. In the bottom row, although the generated motions follow the edit text, they diverge from the source motions.

5.5.5 Additional quantitative & qualitative evaluations

Using the entire test set as gallery. In addition to Tables 5.4 and 5.7 of this Chapter, for completeness, we provide evaluations that use the entire test set as a gallery in Tables 5.5 and 5.6. We use gallery to refer to the set of candidate motions; a query motion is matched against this set by computing similarity scores and retrieving the top results. Since in this setup, models need to retrieve the target or source motion from a larger pool of motions, it is expected to score lower recall. In terms of relative improvements and conclusions, we still observe the same trends between Table 5.5 and Table 5.4, as well as between Table 5.6 and Table 5.7.

Additional metrics. To provide a more comprehensive evaluation, in addition to our retrieval-based metrics, we compute additional measures: FID of motion features and L2 distance between joint positions. We compute both between the test set of MotionFix and the generated motions. For FID, we use the motion branch from the TMR model for the computation of the retrieval metrics. TMED has an FID of 0.129 outperforming the two best baselines 0.152 for MDM-BP$_S$ and 0.145 for MDM-BP. While TMED matches best the distribution of target motions, the baselines perform closely indicating that all models maintain certain degree of realism. Our model also has a lower L2 distance (TMED: 1.10cm, MDM-BP$_S$: 1.26cm, MDM-BP: 1.22cm). Note that coordinate-based metrics are not always reliable given multiple plausible generations due to language ambiguity.

5.5.6 Perceptual studies for quantitative comparisons

To further evaluate TMED, we conducted two perceptual studies: the first one evaluating the absolute quality using a Likert scale, the second one performing a comparative evaluation. In both studies, we used MDM-BP as the strongest baseline to compare with. We conducted the studies across 25 workers on AMT and averaged the results. We used 150 randomly chosen videos from the test set, with each study using a random subset of 50 videos for each method, from those 150 videos. For both studies, the source motion and generation (or ground-truth) were rendered overlaid.

Likert scale: Absolute quality. The objective of this study is to rate how well the instructions are reflected in the videos using a 5-point Likert scale, by asking workers to rate how much they agree with the statement, "the green (generated or ground-truth) motion follows the instruction given the red (source) motion." Choices range from "5: completely agree" to "1: completely disagree." The workers were presented with videos from 3 sets of motions: ground-truth, TMED, and MDM-BP. Each subset is represented by 50 videos, making a total of 150 videos. The videos are presented in 3 batches, each containing a randomized selection from the three sets. Each batch has 17 videos from each set, totaling 51 videos per batch. This ensures that workers evaluate a balanced mix of videos from all three in each batch. The ground-truth (GT) motions are ranked first, TMED second, and MDM-BP third, with means and standard deviations of $\mu \pm \sigma$: 3.93 ± 0.95 (GT), 3.59 ± 1.15 (TMED), and 3.52 ± 1.24 (MDM-BP). This is consistent with our conclusions in Table 5.4 and Table 5.5.

Pairwise comparison: Relative quality. The objective of the second study is to compare our model (TMED) with the best baseline (MDM-BP) by displaying them side-by-side and asking which motion generation better follows the instructions. This study involves a total of 50 videos, divided into two batches of 25 videos each. Additionally, each batch includes 5 initial comparisons to familiarize participants with the task, which are randomly repeated later and discarded from the computations. Moreover, we include 3 catch trials with obvious answers, and filter out the $2/25$ workers who fail them. In each example, two videos are presented side-by-side, randomly swapped for each example. Participants then compare the videos for the given edit text descriptions, and select which of the two follows the instructions better. TMED was chosen over MDM-BP in 65.8% of the comparisons, which again is consistent with the previous conclusions.

5.5.7 Qualitative results

We display several generations from TMED in Figure 5.7 to enable qualitative assessment. We observe that our model can perform different

types of edits such as the addition of actions ("rotate wrists instead of stretching like yawn"), temporal edits in a motion ("get up a bit earlier"), speed edits ("slow down") and combinations of these. We refer to our video for dynamic visualizations, which are easier to interpret [5].

In Figure 5.8, we further provide examples of failures cases from TMED. In the first and third row, we analyze cases with long edit texts. The model struggles with complex details and does not "keep the body straight" in the left example, nor follows "bend arms in the elbows" instructions on the right side, while wider legs are correctly edited. In the second and fourth row, we illustrate examples where the model faithfully follows the edit text, but does not resemble the source motion. In the second row generation, the steps are correctly wider, but the movement does not continue to the similar position as the source motion. Finally, on the right, the body is kneeling down faster as instructed, but towards the opposite direction.

We additionally provide a qualitative comparison in Figure 5.9, between TMED and various baselines. We provide two comparisons for each baseline (top block for MDM_S, middle block for MDM-BP_S, and bottom block for MDM-B).

We observe that MDM_S picks up the action from the prompt, but fails to faithfully follow the source motion. In the first row, the generation by MDM_S raises both hands, instead of adjusting only the height of the hand raised in the source motion. Similarly, in the second row, MDM_S generation raises the arm but in front of the body and not higher as prompted by the edit text.

In the next two rows, we visualize MDM-BP_S results. Given the text "rotate wrists instead of stretching like yawn", GPT correctly suggests editing both hands; however, the generated motion no longer resembles the source motion as the wide-open hands are not preserved. For the example edit "turn in the opposite direction", all body parts are involved, but MDM-BP_S does not deviate too much from the source, perhaps because traditional text-to-motion generation models rarely see relative words such as "opposite direction".

[5] https://www.youtube.com/watch?v=cFa6V6Ua-TY

Finally, we illustrate generations from our strongest baseline MDM-BP in the last two rows. Both generations involve all parts of the body (e.g., "slow down") making it hard to follow the source motion. In comparison, our model faithfully performs most of the edits. The disadvantage of TMED, on the other hand, might be the generalization to motion pairs where the TMR similarity is low as such edits were unseen during training. We briefly discuss more limitations in the following.

5.5.8 Limitations.

Our approach comes with limitations. Assuming two TMR-similar motions being an editing distance apart is not always accurate but serves as a good starting point. Furthermore, in our data collection, we constrain the motions to be up to 5 seconds since longer motions produce many dissimilar pairs. Regarding model performance, TMED exhibits difficulty generalizing to unseen or complex edit texts and maintaining faithfulness to the source motion. Moreover, while our model can be used iteratively—i.e., the edited motion can be fed as input along with a new edit instruction to further edit the motion—we do not explore this capability in this Chapter and leave it for future work.

5.6 Conclusion

In this Chapter, we studied the task of motion editing from language instructions. Given the scarcity of training data, we introduced a new dataset MotionFix, collected in a semi-automatic manner. We exploit motion retrieval models to obtain "edit-ready" motion pairs which we annotate with language labels. We design a conditional diffusion model TMED that is trained on MotionFix, and generates edited motions that follow the source motion and the edit text. We show both quantitatively and qualitatively that our model outperforms all baselines. We hope that our dataset and findings will assist the research community and pave the way for exploring this new task.

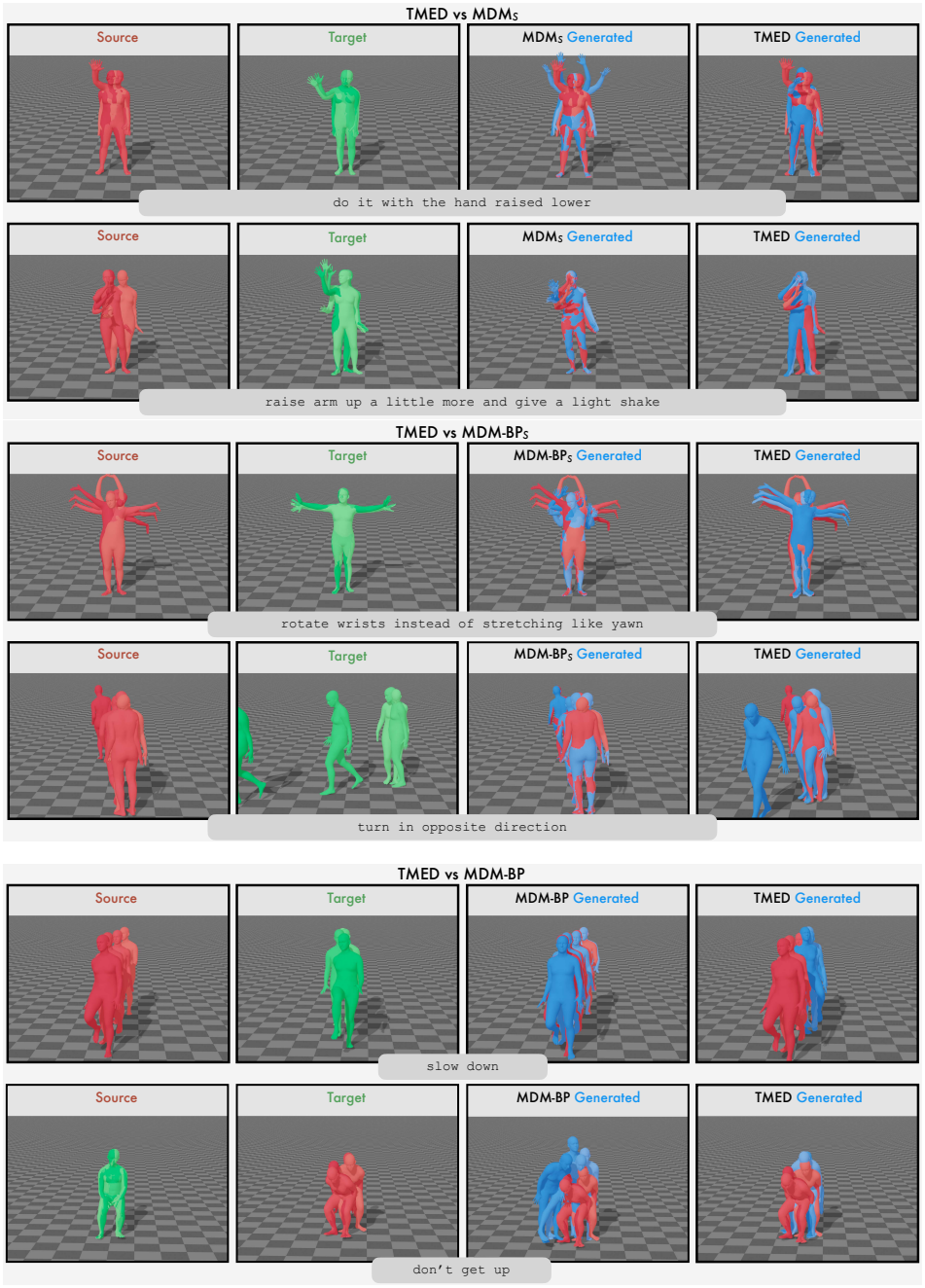


FIGURE 5.9: **Qualitative comparisons with baselines:** We provide example results comparing TMED against the baselines on the MF test set: **MDM**$_S$ (top), **MDM-BP**$_S$ (middle) and **MDM-BP** (bottom). The first two columns show the source (red) and ground-truth target (green) motions. The third column is reserved for baselines, the last column for TMED. Generations are denoted in blue.

Chapter 6

Conclusions & Future Directions

6.1 Summary of contributions

6.1.1 Main conclusions

In this thesis, we explored advancements at the intersection of natural language and 3D human motion, addressing three critical tasks: motion generation, spatial composition, and motion editing. We introduced new datasets, benchmarks, and models tailored to these tasks, each contributing to the broader goal of bridging language and motion in meaningful ways.

For motion generation, we established a new benchmark for action compositions over time, leveraging recursive methods to improve the continuity and realism of transitions between actions. We identified areas for improvement, such as explicitly modeling world contact and incorporating future action anticipation to better emulate human behavior.

For spatial composition, we developed a novel framework that generates simultaneous multi-action motions from textual descriptions. By utilizing GPT-3 for action-to-body-part mappings, we enriched training with synthetic data, enabling our model to generalize to more complex motions. Our findings stimulated further research on learning spatio-temporal compositions [Petrovich et al., 2024; Li et al., 2025] and provided solid approaches for further research.

For motion editing, we introduced a semi-automatically curated dataset, MotionFix, designed for text-driven motion modifications. Leveraging motion retrieval models and a diffusion-based approach, our method generates edited motions that align closely with source motions and textual instructions. Quantitative and qualitative evaluations demonstrate significant improvements over baseline methods.

These contributions collectively advance the integration of language and 3D motion, opening pathways for further research in finegrained motion synthesis, realistic motion editing, and multi-action generation. We hope that our datasets, models, and findings will inspire further progress in this field, much like the synergy observed between language and 2D image research.

6.1.2 Limitations

Our work relies on synthetic data creation and certain assumptions that introduce limitations. Regarding our work in motion editing from text presented in Chapter 5, the use of TMR-similarity to estimate motion-editing distances is not always precise, though it serves as a useful starting point. Additionally, the constraint of limiting motions to sequences of up to 5 seconds avoids excessive dissimilarities but restricts the method's applicability to longer and more complex motions. Furthermore, regarding the work in Chapter 4, the use of language models like GPT-3 for body-part mappings can lead to ambiguities or overly detailed associations, such as misinterpreting actions like "walking" as involving unnecessary body parts. Expanding to finegrained associations, including fingers and facial expressions, remains an area for future exploration.

Our models struggle to generalize to unseen or complex editing prompts. This is partly due to the limited training data. To improve, we need to scale up using large and diverse video datasets from the internet. Leveraging vision-language models (VLMs) trained on such data can help models understand a wider range of actions and contexts. Motion realism also remains an area for improvement, particularly in modeling contact

with the environment and generating realistic transitions between actions. Another key limitation, regarding our work in Chapter 3, is the lack of a "looking ahead" mechanism, as our approach assumes the character only looks backward in time, unlike humans, who plan their actions based on future goals. Incorporating such goal-oriented behavior could further enhance motion plausibility. Future work explores such directions [Zhang et al., 2023d], improving on them and generating better transitions.

Another significant limitation lies in evaluation. While we introduce new metrics like TEMOS score and a new benchmark for text-driven motion editing based on retrieval scores, these measures might fail to capture subjective qualities of motion. Robust evaluation metrics that provide perceptually meaningful insights into motion quality and realism remain a critical open challenge for the field. This gap limits the ability to comprehensively assess the performance of models, especially when dealing with finegrained or nuanced motion edits. Current work looks into benchmarking different metrics for text-to-motion generation [Voas et al., 2023], and highlights the importance of retrieval-based metrics, while illustrating why FID aligns poorly with motion quality. However, further research is needed to pinpoint which set of metrics must be used for each task and how these metrics complement each other.

Finally, the scope of the data used in this thesis presents some constraints. By focusing on motion types, such as pairs of actions or short sequences, and exploring mostly MoCap data, we limit the exploration of diverse and complex motion scenarios. Internet videos can provide a rich source of data containing diverse everyday scenarios, rich context for our actions and their scale can unlock the creation of foundation models. Collecting such data at scale will extend beyond pairs of actions to handle more intricate compositions and longer sequences which will broaden the applicability of our framework. In addition, building on recently introduced bigger datasets [Lin et al., 2023] that are based on internet RGB data would assist generalization and broaden impact. Finally, the exploration of iterative usage of text-based motion editing is an important aspect of the task itself, as editing is an inherently iterative task.

6.2 Future directions

While this work has advanced the field of text-driven 3D motion generation, editing, and composition, several exciting avenues remain open for future exploration.

One key direction is improving the *realism and diversity of generated motions*. Incorporating explicit modeling of environmental interactions, such as contact dynamics with the ground or objects, would enable more natural and physically plausible motions. Additionally, adopting a "looking ahead" mechanism, where models anticipate future actions based on goal-oriented behavior, could bring generated motions closer to human-like planning and execution. The environment and our interaction with it is a key component for future moving agents. Future work should be able to use and build on top of existing approaches to generate realistic interactions for hand-object and full-body-object interactions.

Improving *dataset quality and annotation methodologies* is another important area for future work. Current datasets rely on synthetic MoCap, e.g., the ones created in Chapter 4), existing MoCap collections, e.g., AMASS [Mahmood et al., 2019]. This is labor intense and hard to scale especially for hand and facial motions, as we explained in Chapter 1.

Raw internet videos lack structured motion labels—but that is no longer a dealbreaker. With recent advances in self-supervised learning, 3D human pose and shape estimation (e.g., [Kocabas et al., 2020; Kocabas et al., 2024; Shin et al., 2024]), and inverse rendering, it is now feasible to extract pseudo-3D ground truth at scale from monocular video. These labels may be noisy, but across millions of diverse clips, they become a powerful learning signal—especially when combined with noise-aware training or diffusion-based models. Automatic pipelines can capture semantic richness far beyond lab-controlled MoCap: co-articulation, emotion, failed attempts, social interaction. This shift to in-the-wild motion unlocks stronger priors and better generalization across domains like robotics, animation, and AR/VR. Curating internet-scale motion datasets—even with pseudo ground truth—could be a paradigm shift, just as LAION-5B [Schuhmann et al., 2022] transformed vision-language

models [Ramesh et al., 2022; Brooks et al., 2024]. The challenge now: build scalable pipelines for robust 3D motion labeling and semantic alignment from raw video.

In this second challenge, recent advances in VLMs and LLMs can be proven extremely valuable, as they make it increasingly feasible to automatically generate textual descriptions of actions in internet videos. These models can provide high-level semantic labels, aligning motion with natural language at scale—dramatically reducing the need for manual annotation and enabling more intuitive training data. As captioning models improve, so does our ability to richly annotate complex human motion with minimal supervision. This direction has shown great progress and has been proven crucial for creating bigger models and unprecedented generation quality from the field of text-to-image and text-to-video [Brooks et al., 2024] generation models.

Future research can more deeply extract *structured motion knowledge from LLMs and VLMs*, beyond simple verb-to-motion mappings. These models can help infer which body parts are involved, what contacts are made, and even detect self-interactions (e.g., crossing arms, touching the face) or object affordances (e.g., graspable, pushable, vertices in contact, etc.). This opens the door to learning priors about spatial relationships, motion feasibility, and physical realism—directly from language or multimodal embeddings. In parallel, LLMs and VLMs can serve as high-level planners for both motion generation and reinforcement learning. They can predict goal sequences, provide semantic constraints, and act as reward shaping mechanisms in RL setups—suggesting what outcomes are desirable or plausible in a given scene. This makes them valuable not only as passive priors but also as active reasoning components in control pipelines.

Developing *perceptually meaningful evaluation metrics* is also critical to advancing the field. Current metrics, while innovative, lack the ability to capture subjective qualities of motion, such as naturalness or realism. Future work should focus on designing metrics that align closely with human perception, enabling more robust assessment of motion quality and highlighting realism.

Finally, expanding the scope of current models to handle *more complex and multi-faceted tasks* offers tremendous potential. This includes exploring iterative motion editing, handling simultaneous compositions of multiple actions, generating longer sequences with intricate spatial and temporal relationships, including goals, objects, and the environment. By tackling these challenges, future work can push the boundaries of how language and 3D motion interact, paving the way for richer, more intuitive applications in animation, gaming, and beyond.

Appendix A

BABEL: Bodies, Action and Behavior with English Labels

A.1 Overview of BABEL

A key goal in computer vision is to understand human movement in semantic terms. Relevant tasks include predicting semantic labels for a human movement, e.g., action recognition [Herath et al., 2017], video description [Xu et al., 2016], temporal localization [Sedmidubsky et al., 2019; Zhao et al., 2019], and generating human movement that is conditioned on semantics, e.g., motion synthesis conditioned on actions [Guo et al., 2020], or sentences [Ahuja and Morency, 2019; Lin et al., 2018].

Large-scale datasets that capture variations in human movement and language descriptions that express the semantics of these movements, are critical to making progress on these challenging problems. Existing datasets contain detailed action descriptions for only 2D videos, e.g., ActivityNet [Sedmidubsky et al., 2019], AVA [Gu et al., 2018] and HACS [Zhao et al., 2019]. The large scale 3D datasets that contain action labels, e.g., NTU RGB+D 60 [Shahroudy et al., 2016] and NTU RGB+D 120 [Zou et al., 2022] do not contain ground truth 3D human motion but only noisy estimates. On the other hand, motion-capture (MoCap) datasets [Ghorbani et al., 2021; Harvey et al., 2020; Ionescu et al., 2014] are small in scale and are only sparsely labeled with very few actions. We address this shortcoming with BABEL, a large dataset of diverse, densely annotated,

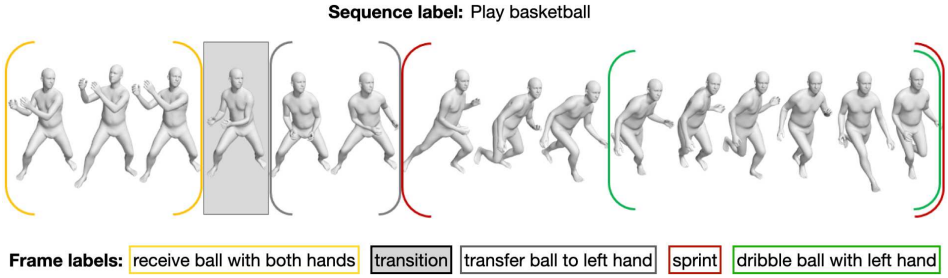


FIGURE A.1: People moving naturally often perform multiple actions simultaneously, and sequentially, with transitions between them. BABEL contains sequence labels describing the overall action in the sequence, and frame labels where all frames and all actions are labeled. Each frame label is precisely aligned with the frames representing the action (colored brackets). This includes simultaneous actions (nested brackets) and transitions between actions (shaded gray box).

actions with labels for all the actions in a motion capture (MoCap) sequence.

We acquire action labels for sequences in BABEL, at two different levels of resolution. Similar to existing MoCap datasets, we collect a sequence label that describes the action being performed in the entire sequence, e.g., `Play basketball` in Figure A.1. At a finer-grained resolution, the frame labels describe the action being performed at each frame of the sequence, e.g., `transfer ball to the left hand`, `sprint`, etc. The frame labels are precisely aligned with the corresponding frames in the sequence that represent the action. BABEL also captures simultaneous actions, e.g., `sprint` and `dribble ball with left hand`. When collecting frame labels, we ensure that all frames in a sequence are labeled with at least one action, and all the actions in a frame are labeled. This results in dense action annotations for high-quality MoCap data.

BABEL contains action annotations for about 43.5 hours of MoCap from AMASS, with 15472 unique language labels. Via a semi-automatic process of semantic clustering followed by manual categorization, we organize these into 260 action categories such as `greet`, `hop`, `scratch`, `dance`, `play instrument`, etc. The action categories in BABEL belong to 8 broad semantic categories involving simple actions (`throw`, `jump`), complex

activities (`martial arts`, `dance`), body part interactions (`scratch`, `touch face`), etc. (see Section A.2.4).

BABEL contains a total of 28055 sequence labels, and 63353 frame labels. This corresponds to dense per-frame action annotations for 10892 sequences ($>$ 37 hours of MoCap), and sequence-level annotations for all 13220 sequences ($>$ 43 hours of MoCap). On average, a single MoCap sequence has 6.06 segments, with 4.02 unique action categories. We collect the sequence labels via a web interface of our design, and the frame labels and alignments by adapting an existing web annotation tool, VIA [Dutta and Zisserman, 2019] (see Section A.2.1). Labeling was done by using AMT.

In this chapter, we make the following contributions: (1) We provide the largest 3D dataset of dense action labels that are precisely aligned with their corresponding movement spans in the MoCap sequence. (2) We categorize the raw language labels into over 250 action classes that can be leveraged for tasks requiring categorical label sets, such as 3D action recognition. (3) We analyze the actions occurring in BABEL sequences in detail, furthering our semantic understanding of MoCap data that is already widely used in vision tasks. (5) The dataset, is publicly available for academic research purposes at `https://babel.is.tue.mpg.de/`.

A.2 BABEL dataset

We first provide details regarding the crowdsourced data collection process. We then describe the types of labels in BABEL, and the label processing procedure.

A.2.1 Data collection

We collect BABEL by showing rendered videos of MoCap sequences from AMASS [Mahmood et al., 2019] to human annotators and eliciting action labels (Figure A.2). The MoCap is processed to make sure the person in the video faces the annotator in t he first frame. We observe that a sequence labeled as `pick up object` often also involves other

actions such as walking to the object, bending down to pick up the object, grasping the object, straightening back up, turning around and walking away. We argue that labeling the entire sequence with the single label is imprecise, and problematic. First, many actions such as `turn` and `grasp` are ignored and remain unlabeled, although they may be of interest to researchers [Taheri et al., 2020]. Second, sequence labels provide weak supervision to statistical models, which are trained to map the concept of `picking up object` to the whole sequence when it, in fact, contains many different actions. To illustrate this point, we examine a typical sequence (see Qualitative Example 1 in the project website), and find that only 20% of the duration of the sequence labeled as `pick up and place object` corresponds to this action. Crucially, walking towards and away from the object — actions that remain unlabeled — account for 40% of the duration. While this makes semantic sense to a human — picking up and placing an object is the only action that changes the state of the world and hence worth mentioning, this might be suboptimal training data to a statistical model, especially when the dataset also contains the confusing classes `walk`, `turn`, etc. Finally, using noisy labels as ground truth during evaluation does not accurately reflect the capabilities of models.

We address this with action labels at two levels of resolution — a label describing the overall action in the entire sequence, and finegrained labels that are aligned with their corresponding spans of movement in the MoCap sequence.

A.2.2 BABEL annotation interfaces

Recall that in BABEL, we label the actions in movement sequences in two stages. First, we acquire labels at a sequence level, where a single action describes the movement in the entire sequence. Then, if there are multiple actions in the sequence, we collect dense frame-level labels (as discussed in Section A.2.1).

In both tasks, we show annotators videos that are rendered from MoCap sequences. We ensure that the human figure in the rendered video faces the viewer (camera) in the first frame of the sequence.

Sequence labels. The web annotation interface for sequence-level action labels shows a rendered video of the motion-capture (MoCap) sequence. We first ask the annotator the following question — 'Does the video contain more than one action?'. In response, the annotator can choose either 'yes' or 'no'. If the response is 'yes', then we ask the following question — 'If you had to describe the whole video as one action, what would it be?.' On the other hand, if the response is 'no', we instruct the annotator to, 'Name the action:'. In both cases (i.e., when the sequence contains one or multiple actions), we collect the action label that describes the entire sequence — the sequence-level label, via a text-box (free-form labels). The text-box also has an auto-complete feature which matches the person's current input with a fixed list of (typically, single-word) action verbs. The annotator may choose one of the existing actions, or type in a new action. We observe that annotators often enter novel action labels.

We provide text instructions for the task and also provide annotators with example annotations in the interface. A demo of the sequence label annotation interface is available at our project webpage: `https://babel.is.tue.mpg.de/demos.html`.

Frame labels. We collect frame-level labels for a sequence if the annotators agree that it contains more than one action. We modify the VIA annotation software [Dutta and Zisserman, 2019] to suit our requirements. Specifically, we use the video annotation application that allows annotation of temporal segments and spatial regions. We remove the spatial annotation features and customize it for marking 'actions' in the video. We also include tests like making sure there is no gap in the annotation, there is more than one action in the video, etc.

In the frame-level labeling task, the person first watches a video of the movement sequence. Then, they list all the actions occurring in the sequence. This includes actions that occur sequentially and simultaneously. Once the annotator has confirmed entering all the actions in a text-box, a horizontal 'timeline bar' is created below the video for each action. The annotator can denote the start and the end time of the action in the sequence, by creating an 'action segment'. To create an action segment,

the annotator highlights the timeline for the action of interest, and presses the 'a' key on the keyboard. The start and end times are modified by mouse click and drag operations (or keyboard shortcuts) to accurately reflect the corresponding duration of the action in the video.

Since there is significant variance in the length of the sequences and the number of actions in a sequence, in addition to the fixed pay, we also provide annotators with an optional bonus payment. The bonus payment is proportional to the number of action segments that are labeled per sequence. To encourage annotators to be thorough, we mention this in the instructions.

When multiple actions occur in a sequence, there is often a transition between them. To reduce user effort, and encourage people to explicitly annotate transitions between actions, we populate the list of actions with a 'transition' action by default.

We provide detailed instructions for the task in the interface. In addition, we also ask annotators to watch a video tutorial explaining the interface and task with examples. Recall that, to ensure the quality of annotations, we first provide annotators with a test task and only qualify annotators who demonstrate a clear understanding of our task.

A demo of the frame label annotation interface is available at our project webpage: https://babel.is.tue.mpg.de/demos.html. In the demo, we provide the interface with 2 MoCap sequences. The first sequence, a labeled sample from BABEL, illustrates the level of detailed annotation of a completed frame-level labeling task. We leave the second sequence unlabeled in case there is interest in attempting the annotation task.

A.2.3 BABEL action labels

We collect BABEL labels in a two-stage process — first, we collect sequence labels, and determine whether the sequence contains multiple actions. We then collect frame labels for the sequences where 2 annotators agree that there are multiple actions.

Sequence labels. In this labeling task, annotators answer two questions regarding a sequence. We first ask annotators if the video contains more than one action (yes/no). Note that the initial 'T-pose' for calibration, followed by standing, are considered separate actions with a transition between them. If the annotator chooses 'no', we ask them to name the action in the video. If they instead choose 'yes', we elicit a sequence label with the question, "If you had to describe the whole sequence as one action, what would it be?" We provide the web-based task interface in the project website.

We ask annotators to enter the sequence labels in a text-box, with the option of choosing from an auto-complete drop-down menu that is populated with a list of basic actions. We specifically elicit free-form labels (rather than a fixed list of categories) from annotators to discover the diversity in actions in the MoCap sequences. We find that in most cases, annotators tend to enter their own action labels. This also presents a challenge, acting as a source of label variance. Apart from varying vocabulary, free-form descriptions are subject to ambiguity regarding the 'correct' level in the hierarchy of actions [Gu et al., 2018], e.g., `raise left leg`, `step`, `walk`, `walk backwards`, `walk backwards stylishly`, etc.

We collect 2 labels per sequence, and in case of disagreement regarding multiple actions, a third label. We determine that a sequence contains a single action or multiple actions based on the majority vote of annotators' labels. Overall, BABEL contains 28055 sequence labels[1].

Frame labels. Frame labels contain language descriptions of all actions that occur in the sequence, and precisely identify the span in the sequence that corresponds to the action. We leverage an existing video annotation tool, VIA [Dutta and Zisserman, 2019], and modify the front-end interface and back-end functionality to suit our annotation purposes. For instance, we ensure that every frame in the sequence is annotated with at least one action label. This includes 'transition' which indicates a transition between two actions, or 'unknown' which indicates that the annotator is unclear as to what action is being performed. This provides us with dense

[1]Note that a few sequences have additional labels.

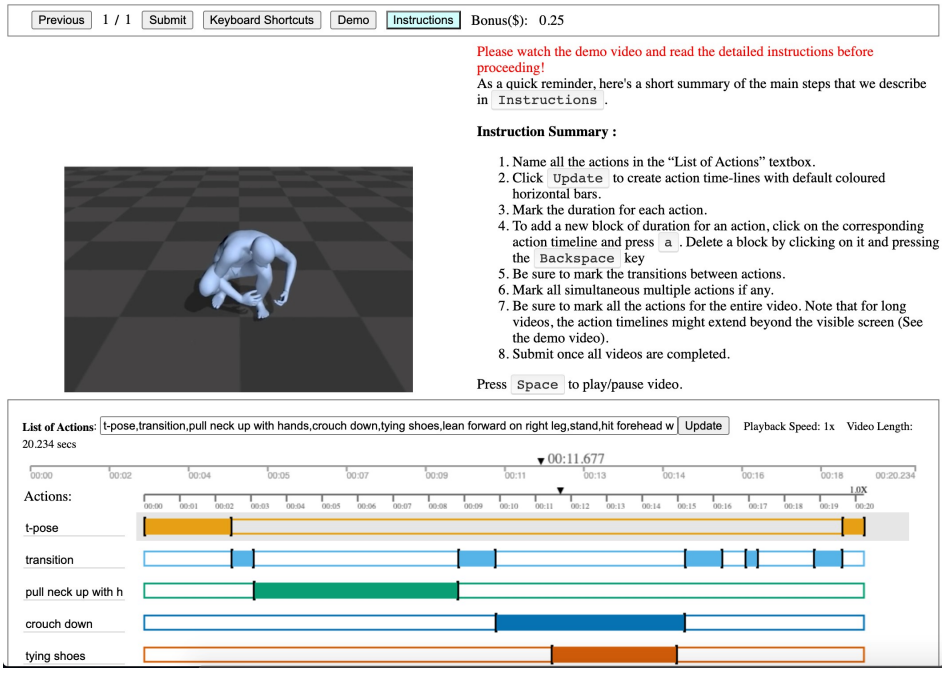


FIGURE A.2: BABEL annotation interface to collect frame-level action labels. Annotators first name all the actions in the video. They then, precisely align the length of the action segment (colored horizontal bar) with the corresponding duration of the action in the video. This provides dense action labels for the entire sequence.

annotations of action labels for the sequence. A screenshot of the AMT task interface for frame label annotation in BABEL is shown in Figure A.2.

To provide frame labels, an annotator first watches the whole video and enters all the actions in the 'List of Actions' text-box below the video. This populates a set of empty colored box outlines corresponding to each action. The annotator then labels the span of an action by creating a segment (colored rectangular box) with a button press. The duration of the segment and the start/end times can be changed via simple click-and-drag operations. The video frame is continuously updated to the appropriate time-stamp corresponding to the end time of the current active segment. This provides the annotator real-time feedback regarding the exact starting point of the action. Once the segment is placed, its precision can be verified by a 'play segment' option that plays the video span corresponding to the

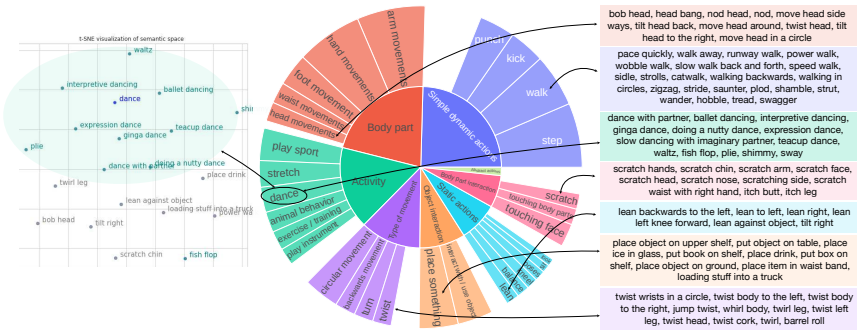


FIGURE A.3: *Left.* 2D t-SNE [Maaten and Hinton, 2008] visualization of the semantic space that we project raw labels into. Similar labels are grouped via K-means clustering. The green shading and points represent the 'dance' cluster and its members, respectively. *Center.* Distribution of (a subset of) action categories (outer circle) under each semantic category (inner circle) in BABEL. The angle occupied by the action category is proportional to the number of unique raw label strings associated with it. Action categories with a large number of finegrained descriptions are shown. *Right.* Subset of the finegrained descriptions associated with selected action categories.

current segment. In case of errors, the segment can be further adjusted. We provide detailed instructions via text, and a video tutorial that explains the task with examples, and demonstrates the operation of the annotation interface. The web interface of the task is provided in the project website.

We collect frame labels for 6663 sequences where both annotators who provided sequence labels agree that the sequence contains multiple actions[2].

Overall, BABEL contains dense annotations for a total of 66018 action segments for 10892 sequences. This includes both frame labels from sequences containing multiple actions, and sequence labels from sequences containing a single action. If an entire sequence has only a single action, it counts as 1 segment.

A.2.4 Label processing

BABEL contains a total of 15472 unique raw action labels. Note that while each action label is a unique string, labels are often semantically similar

[2]Note that some sequences are labeled by more than 1 annotator.

(`walk`, `stroll`, etc.), are minor variations of an action word (`walking`, `walked`, etc.) or are misspelled. Further, tasks like classification require a smaller categorical label set. We organize the raw labels into two smaller sets of semantically higher-level labels — action categories, and semantic categories.

Action categories. We map the variants of an action into a single category via a semi-automatic process that involves clustering the raw labels, followed by manual adjustment.

We first pre-process the raw string labels by lower-casing, removing the beginning and ending white-spaces, and lemmatization. We then obtain semantic representations for the raw labels by projecting them into a 300D space via Word2Vec embeddings [Mikolov et al., 2013]. Word2Vec is a widely used word embedding model that is based on the distributional hypothesis — words with similar meanings have similar contexts. Given a word, the model is trained to predict surrounding words (context). An intermediate representation from the model serves as a word embedding for the given word. For labels with multiple words, the overall representation is the mean of the Word2Vec embeddings of all words in the label. Labels, containing words that are semantically similar, are close in the representation space.

We cluster labels that are similar in the representation space via K-means ($K = 200$ clusters). This results in several semantically meaningful clusters, e.g., `walk`, `stroll`, `stride`, etc. which are all mapped to the same cluster. We then manually verify the cluster assignments and fix them to create a semantically meaningful organization of the action labels. Raw labels that are not represented by Word2Vec (e.g., `T-pose`) are manually organized into relevant categories in this stage. For each cluster, we determine a category name that is either a synonym ('walk' $\leftarrow$ {`walk`, `stroll`, `stride`}) or hypernym ('walk' $\leftarrow$ {`walk forward`, `walk around`}) that describes all action labels in the cluster.

Some raw labels, e.g., `rotate wrists` can be composed into multiple actions like `circular movement` and `wrist movement`. Thus, raw labels are occasionally assigned membership to multiple action categories.

Overall, the current version of BABEL has 260 action categories. Interestingly, the most frequent action in BABEL is 'transition' — a movement that usually remains unlabeled in most datasets. There are 18447 transitions between different actions in BABEL. Unsurprisingly, the frequency of actions decreases exponentially following Zipf's law — the 50th most frequent action category `catch` occurs 417 times, the 100th most frequent action category `misc. activities`, occurs 86 times, and the 200th most frequent action category `disagree`, occurs 8 times. We visualize the action categories containing the largest number of raw labels (cluster elements) in Figure A.3 (outer circle). Raw labels corresponding to these categories are shown on the right. We provide histograms of duration and number of segments per-action, in Section A.2.5 and project webpage.

Semantic categories of labels. Action categories often reflect qualitatively different types of actions like interacting with objects, actions that describe the trajectory of movement, complex activities involving multiple actions, etc. We formalize the different types of actions in BABEL into 8 semantic categories (inner circle in Figure A.3): We categorize actions into several types. **Simple dynamic actions** include low-level atomic motions such as `walk`, `run`, `kick`, and `punch`. **Static actions** refer to postures, either assumed or maintained, like `sit`, `stand`, or `kneel`. **Object interaction** involves manipulating objects, e.g., `place something`, `move something`, or `use object`. **Body part interaction** includes actions like `touching face` or `scratch`, which typically involve self-contact between body parts. **Body part actions** describe the motion of specific parts—such as `raise arm`, `lower head`, or `rotate wrist`. The **type of movement** refers to the trajectory or nature of the motion itself, e.g., `twist` or `circular movement`. **Activities** are more complex, often comprising multiple simpler actions—for example, `play sports={run, jump}` or `dance={stretch, bend}`. Finally, **abstract actions**—such as `excite`, `endure`, `learn`, or `find`—tend to reflect emotional or cognitive states and have highly variable physical realizations; only a few of these appear in BABEL.

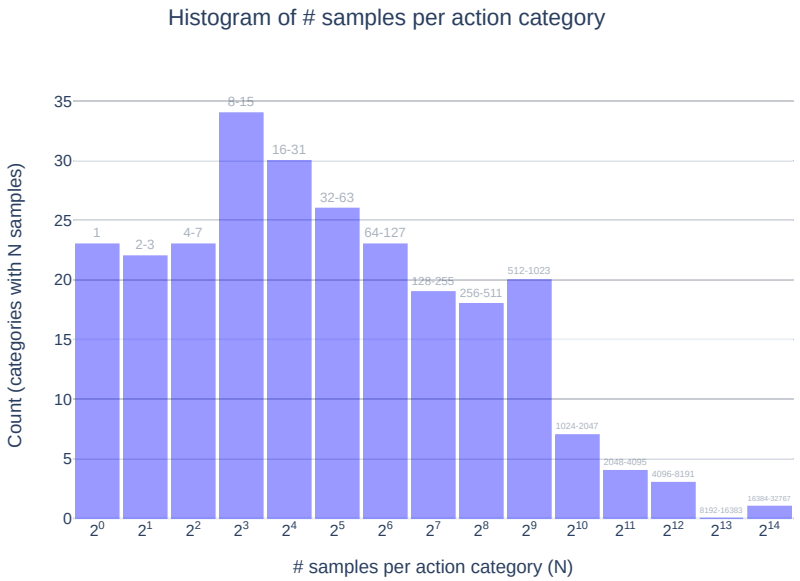


FIGURE A.4: The distribution of the number of samples across action categories in BABEL is long-tailed. Y-axis denotes the number of action categories that contain N samples. X-axis denotes the number of samples belonging to an action category (N) in `log` scale.

We provide a search interface to visualize samples from BABEL here: https://babel.is.tue.mpg.de/explore.html.

A.2.5 BABEL labels

Label organization. We provide the action categories and raw action labels in BABEL in https://babel.is.tue.mpg.de/explore.html. Note that the visualization is interactive — hovering the mouse over each semantic category shows, in the 'value' field, the number of action categories within the semantic category. The width of each action category is proportional to the number of raw labels associated with the category, which is also visible under the 'value' field on hovering over each action category.

Label distribution. The MoCap sequences in BABEL are acquired from AMASS [Mahmood et al., 2019] which contains many MoCap datasets,

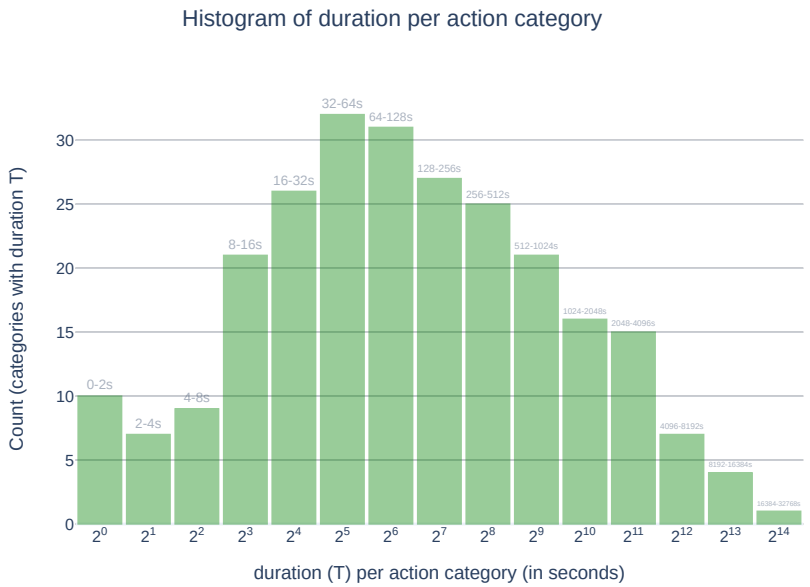


FIGURE A.5: The distribution of the duration across action categories in BABEL is long-tailed but less skewed than the number of samples per category. Y-axis denotes the number of action categories that account for duration T. X-axis denotes the total duration (T) that an action category accounts for, in `log` scale.

as described in Section A.2.1. Thus, BABEL does not have a strictly controlled distribution of categories, unlike many other datasets. The distribution of action categories in BABEL, is long-tailed, as shown in Figure A.4. Similarly, the overall duration that action categories account for, also follow a long-tailed distribution, as shown in Figure A.5. While we expect that the shape of these distributions will change as BABEL grows, we do expect the distributions to remain skewed, similar to many naturally occurring distributions. We believe that learning from skewed distributions that occur naturally, is an important, and challenging problem. BABEL has the potential to serve as a benchmark which encourages and reflects progress in the ability to deploy algorithms to real-world applications that have skewed class distributions.

In BABEL, action categories such as 'walk', 'transition', 'stand', etc. occur quite frequently in the dataset, and the frequency of other

classes decreases exponentially, following Zipf's law. We visualize the overall number of segments for each action category in BABEL, in https://babel.is.tue.mpg.de/stats.html.

A.3 Analysis of BABEL

Natural human movement often contains multiple actions and transitions between them. Modeling the likelihood of simultaneous actions and action transitions has applications in reasoning about action affordances in robotics and virtual avatars, motion synthesis [Tanke et al., 2019], activity forecasting [Kitani et al., 2012], animation [Starke et al., 2019], and action recognition.

A.3.1 Simultaneous actions

Although people often perform multiple actions simultaneously in real life, this is rarely captured in labeled datasets. Recall from Section A.2.3 that in BABEL, we ask annotators to label all actions that are occurring in each frame of the sequence. Overall, BABEL has 49952 instances of simultaneous actions that occur with 2907 unique pairs of action categories. Simultaneous actions are defined as actions that overlap for a duration of > 0.1 seconds. We exclude the overlap of an action with `transition` since this implies adjacent actions.

Simultaneous actions often exhibit relationships such as:

- **Hierarchical**. Some simultaneous actions reflect the hierarchical structure in actions. For instance, a complex activity & action comprising the activity, e.g., `eating food` & `raise right hand to mouth`, and `dancing` & `extend arms`.
- **Complementary**. The two actions are independent, e.g., `hold with left hand` & `look right`.
- **Superimposed**. An action can move a certain body part that partly modifies another action; e.g., `carry with right hand` modifies the

complex activity `walk`, and `right high kick` partly modifies the static (full body) action `fight stance`.

- **Compositional**. Actions involving the same body parts that result in a body or part movement that is a function of both actions, e.g., `walking` & `turn`.

A.3.2 Temporally adjacent actions

The dense labels in BABEL capture the progression of actions in MoCap sequences. We analyze adjacent actions where where action a_i follows a_j (denoted by $a_j \rightarrow a_i$). a_i and a_j denote action segments, i.e., a contiguous set of frames corresponding to an action (and not the action for a single frame). Thus, $a_i \neq a_j$ if the actions are adjacent. We say $a_j \rightarrow a_i$ if the frame succeeding the last frame of a_j is the first frame of a_i. In practice, we account for imprecise human temporal annotations by ignoring a small overlap in duration (< 0.1 sec.) between a_i and a_j. We also disregard the separation of actions by `transition`; i.e., $a_j \rightarrow a_i$ if $a_j \rightarrow a_t$ and $a_t \rightarrow a_i$, where $a_t =$ `transition`.

We visualize the frequent transitions between actions, i.e., $a_j \rightarrow a_i$ sorted by `Count`$(a_j \rightarrow a_i)$ in BABEL, in Figure A.6. We observe that `walk`, unsurprisingly, has the most diverse set of adjacent actions, i.e., `Count(walk`$\rightarrow a_i$`)` and `Count(`$a_j \rightarrow$`walk)` are large. While transitions between action pairs such as (`jog`, `turn`), (`walk`, `t-pose`) are bidirectional (with $\sim$ equal frequency), others have fewer adjacent actions. Some action categories with few transitions illustrate semantically meaningful action chains, e.g., `sit` $\rightarrow$ `stand up` $\rightarrow$ `walk` and `walk` $\rightarrow$ `bend` $\rightarrow$ `pick something up` $\rightarrow$ `place something`. Unidirectional transitions such as `sit` $\rightarrow$ `stand up` and `walk` $\rightarrow$ `sit` implicitly indicate the arrow of time [Pickup et al., 2014]. Interestingly, the transition from `sit` $\rightarrow$ `stand up`, and the lack of transition from `sit` $\rightarrow$ `stand` delineates the subtle difference between the labels `stand` (static action of 'maintaining an upright position') and `stand up` (dynamic action of 'rising into an upright position').

Given the temporally adjacent actions in BABEL, we attempt to model the transition probabilities between actions, i.e., $P(a_i|a_j)$. This can be done

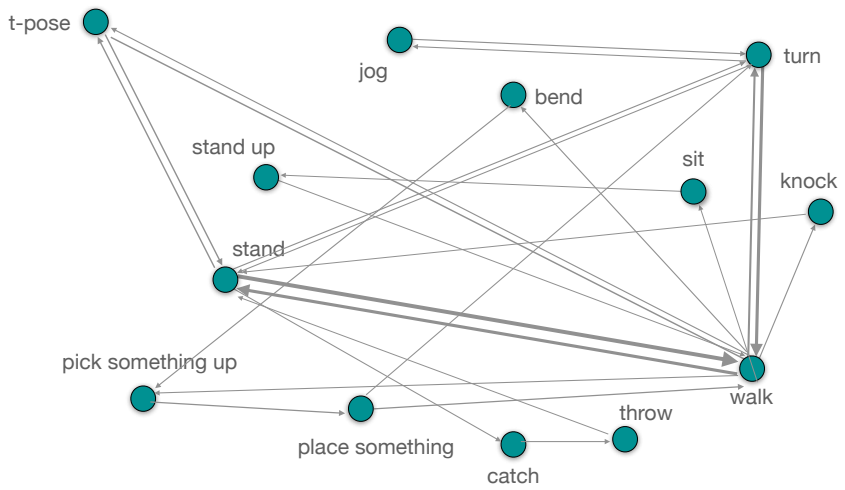


FIGURE A.6: Node represent actions, and an edge represents a transition between these actions in the MoCap sequence. Edge thickness $\propto \texttt{Count}(a_i \rightarrow a_j)$ (frequency of transition) in BABEL.

by using an order 3 Markov Chain [Markov, 1906]. Concretely, we compute $P(a_i|a_{i-1}, a_{i-2}, a_{i-3})$, and observe in Table A.1 that random walks along this chain generate plausible action sequences for human movement.

AMASS. The MoCap sequences in BABEL are derived from AMASS [Mahmood et al., 2019], which is a large corpus of MoCap datasets. As a consequence, BABEL could have inherited the same biases present in these existing MoCap datasets.

Each dataset in AMASS differs in the number and distribution of actions, number of subjects, the duration of the sequence, etc. We examine a few of these in more detail below.

Sequence duration. In Figure A.7, we present a histogram of sequence durations of AMASS MoCap sequences. We see that AMASS has a bimodal distribution over sequence durations. Interestingly, we observe a large, narrow spike in the bin of range $(4.90, 4.95)$, indicating the presence of 418 sequences with duration sec., i.e., $(29.86, 30.91)$ sec. $318/418$ sequences in this duration-range belong to the Eyes Japan dataset which are exactly 30.0

#	Transition of actions
1	walk, transition, pick up, set down, transition, walk clockwise, transition, stand
2	a-pose, transition, wave hands in and out, wave arms in front of left, transition, cross left leg in circle gesture series, transition, t-pose
3	looking left, standing, transition, looking right, standing
4	stepping forward, standing, turning back, walking back, walking forward, standing, losing balance, transition, turning around, walking, standing
5	step back, stand, transition, walk to the left

TABLE A.1: Random walk samples based on action transition probabilities learned from BABEL. The generated samples are plausible action sequences simulating natural human movement.

sec. long. We note that in real-world, it is likely that we encounter action lengths of varying durations. Training models on carefully constructed movement sequences of fixed constant durations might negatively impact the generalization of the models to real-world data.

A.3.3 Action segments

Accurately labeling a movement sequence with all actions and their precise spans is a difficult, attention-demanding task. When an annotator chooses the option to mark the span of an action, a segment is added to the selected action, beginning from the current time of the video. It is likely that the annotator's task will be easier if the span of the default segment is close to the actual span of the action in the video. In our interface, the default segment duration is fixed at 1 sec.

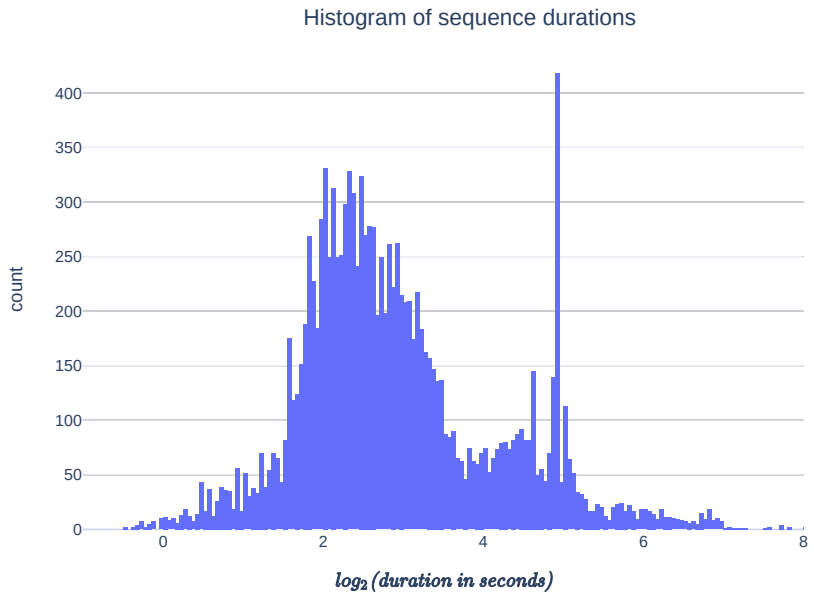


FIGURE A.7: The distribution of the sequence durations in AMASS. Y-axis denotes the number of segments within duration specified by the bin. X-axis denotes durations of bins in `log` scale.

We provide the histogram of the durations of segments in BABEL in Figure A.8. We observe that the distribution of segment durations approximates a log-normal distribution, with the mode of the distribution around 2^0, i.e., 1 sec. However, we also observe a spike around the mode (1 sec.). There are about twice as many segments in the $\sim$ 1 sec. bin, compared to its neighbors. This indicates a potential bias towards 1 sec. segments. We think that a likely reason for this bias could be that the default segment duration in our interface is 1 sec. A possible hypothesis is that when the span of an action in the video is close to 1 sec., annotators avoid additional precise manual editing of the ending of the default segment.

As part of our data quality control process, we visualize many random sequences along with their frame-level levels, and manually verify that the labeling is accurate (with a small error margin). Further, we only make the task available to around 130 annotators whose work we determine to be

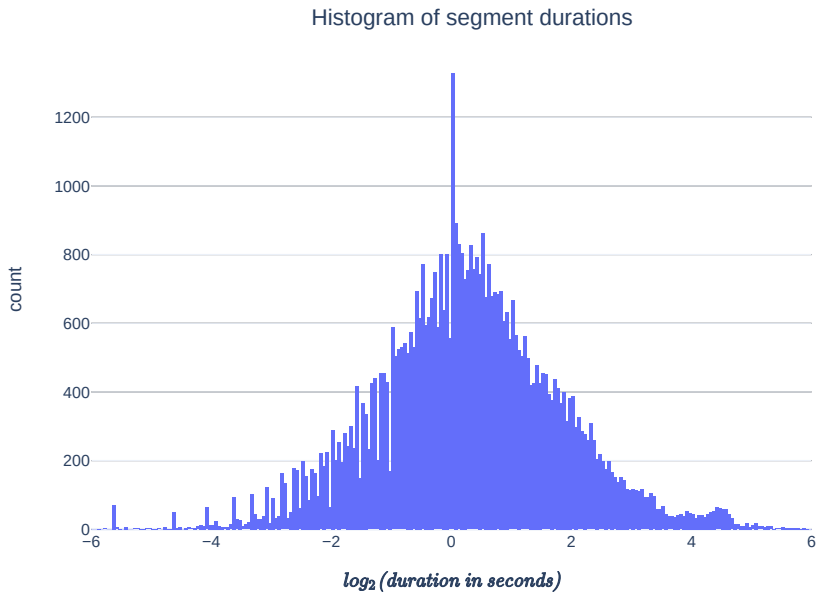


FIGURE A.8: The distribution of the segment durations in BABEL, with a mode around 1 sec. duration. Y-axis denotes the number of segments within duration specified by the bin. X-axis denotes durations of bins in `log` scale.

reliable. So we do not expect the effect of this bias to largely impact the quality of the dataset. Indeed, this bias was not obvious to us from the label visualization — this is apparent only when viewing the histogram of segment durations with a small bin size.

A possible solution for the future could be an interface design where every segment is initialized with a random duration. It would be interesting to: (1) identify if there are still a relatively larger number of 1 sec. segments, and (2) measure how often annotators edit the default segment duration. However, there exists an important trade-off with a design where the default segment duration may be quite different from the duration of the majority of the action spans. Having to perform larger, and more frequent edits of segments could negatively impact the overall labeling quality, throughput, and annotator satisfaction with the task.

Bibliography

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. "Image2StyleGAN++: How to Edit the Embedded Images?" In: *Computer Vision and Pattern Recognition (CVPR)*. 2020.

[2] Kfir Aberman, Yijia Weng, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. "Unpaired Motion Style Transfer from Video to Animation". In: *Transactions on Graphics (TOG)* (2020).

[3] Hyemin Ahn, Timothy Ha, Yunho Choi, Hwiyeon Yoo, and Songhwai Oh. "Text2Action: Generative Adversarial Synthesis from Language to Action". In: *International Conference on Robotics and Automation (ICRA)*. 2018.

[4] Chaitanya Ahuja and Louis-Philippe Morency. "Language2Pose: Natural Language Grounded Pose Forecasting". In: *International Conference on 3D Vision (3DV)*. 2019.

[5] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. "Social LSTM: Human Trajectory Prediction in Crowded Spaces". In: *Computer Vision and Pattern Recognition (CVPR)*. 2016.

[6] Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. "Listen, Denoise, Action! Audio-Driven Motion Synthesis with Diffusion Models". In: *Transactions on Graphics (TOG)* (2023).

[7] Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Lars Petersson, and Stephen Gould. "A Stochastic Conditioning Scheme for Diverse Human Motion Prediction". In: *Computer Vision and Pattern Recognition (CVPR)*. 2020.

[8] Brett Allen, Brian Curless, and Zoran Popović. “The Space of Human Body Shapes”. In: *Transactions on Graphics (TOG)* (2003).

[9] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. “SCAPE: Shape Completion and Animation of People”. In: *Transactions on Graphics (TOG)* (2005).

[10] Tenglong Ao, Zeyi Zhang, and Libin Liu. “GestureDiffuCLIP: Gesture Diffusion Model with CLIP Latents”. In: *Transactions on Graphics (TOG)* (2023).

[11] João Pedro Araújo, Jiaman Li, Karthik Vetrivel, Rishi Agarwal, Jiajun Wu, Deepak Gopinath, Alexander Clegg, and C Karen Liu. “CIRCLE: Capture in Rich Contextual Environments”. In: *Computer Vision and Pattern Recognition (CVPR)*. 2023.

[12] Okan Arikan, David A Forsyth, and James F O’Brien. “Motion Synthesis from Annotations”. In: *Transactions on Graphics (TOG)* (2003).

[13] Kumar Ashutosh, Tushar Nagarajan, Georgios Pavlakos, Kris Kitani, and Kristen Grauman. “ExpertAF: Expert actionable feedback from video”. In: *Computer Vision and Pattern Recognition (CVPR)*. 2025.

[14] Nikos Athanasiou, Alpár Ceske, Markos Diomataris, Michael J Black, and Gül Varol. “MotionFix: Text-driven 3D Human Motion Editing”. In: *International Conference on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH ASIA)*. 2024.

[15] Nikos Athanasiou, Mathis Petrovich, Michael J Black, and Gül Varol. “SINC: Spatial Composition of 3D Human Motions for Simultaneous Action Generation”. In: *International Conference on Computer Vision (ICCV)* (2023).

[16] Nikos Athanasiou, Mathis Petrovich, Michael J Black, and Gül Varol. “TEACH: Temporal Action Composition for 3D Humans”. In: *International Conference on 3D Vision (3DV)*. 2022.

[17] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *International Conference on Learning Representations (ICLR)* (2014).

[18] Alexandru O Balan, Leonid Sigal, Michael J Black, James E Davis, and Horst W Haussecker. "Detailed Human Shape and Pose from Images". In: *Computer Vision and Pattern Recognition (CVPR)*. 2007.

[19] German Barquero, Sergio Escalera, and Cristina Palmero. "Seamless Human Motion Composition with Blended Positional Encodings". In: *Computer Vision and Pattern Recognition (CVPR)*. 2024.

[20] Emad Barsoum, John Kender, and Zicheng Liu. "HP-GAN: Probabilistic 3D Human Motion Prediction Via GAN". In: *Computer Vision and Pattern Recognition (CVPR)*. 2017.

[21] Robbert-Jan Beun, Eveliene de Vos, and Cilia Witteman. "Embodied Conversational Agents: Effects on Memory Performance and Anthropomorphisation". In: *ACM International Conference on Intelligent Virtual Agents*. 2003.

[22] Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. "BEDLAM: A Synthetic Dataset of Bodies Exhibiting Detailed Lifelike Animated Motion". In: *Computer Vision and Pattern Recognition (CVPR)*. 2023.

[23] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. "Keep It SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image". In: *European Conference on Computer Vision (ECCV)*. 2016.

[24] Richard Bowden. "Learning Statistical Models of Human Motion". In: *Computer Vision and Pattern Recognition (CVPR)*. 2000.

[25] Tim Brooks, Aleksander Holynski, and Alexei A Efros. "InstructPix2Pix: Learning to Follow Image Editing Instructions". In: *Computer Vision and Pattern Recognition (CVPR)*. 2023.

[26] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, and Eric Luhman. *Video Generation Models as World Simulators*. 2024.

[27] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, and Amanda Askell. "Language Models Are Few-shot Learners". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2020.

[28] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-end Object Detection with Transformers". In: *European Conference on Computer Vision (ECCV)*. 2020.

[29] Edwin Catmull. "A System for Computer Generated Movies". In: *Proceedings of the ACM Annual Conference*. 1972.

[30] Ling-Hao Chen, Shunlin Lu, Ailing Zeng, Hao Zhang, Benyou Wang, Ruimao Zhang, and Lei Zhang. "MotionLLM: Understanding Human Behaviors from Human Motions and Videos". In: *arXiv:2405.20340* (2024).

[31] Muxi Chen, Yi Liu, Jian Yi, Changran Xu, Qiuxia Lai, Hongliang Wang, Tsung-Yi Ho, and Qiang Xu. "Evaluating Text-to-image Generative Models: An Empirical Study on Human Image Synthesis". In: *arXiv:2403.05125* (2024).

[32] Wenzheng Chen, Huan Wang, Yangyan Li, Hao Su, Zhenhua Wang, Changhe Tu, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. "Synthesizing Training Images for Boosting Human 3D Pose Estimation". In: *International Conference on 3D Vision (3DV)*. 2016.

[33] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. "Executing Your Commands Via Motion Diffusion in Latent Space". In: *Computer Vision and Pattern Recognition (CVPR)*. 2023.

[34] Kiran Chhatre, Radek Daněček, Nikos Athanasiou, Giorgio Becherini, Christopher Peters, Michael J Black, and Timo Bolkart. "AMUSE: Emotional Speech-driven 3D Body Animation Via Disentangled Latent Diffusion". In: *Computer Vision and Pattern Recognition (CVPR)*. 2024.

[35] Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. "Flexible Motion In-betweening with Diffusion Models". In: *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 2024.

[36] Michael F Cohen. "Interactive Spacetime Control for Animation". In: *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 1992.

[37] B. T. Davies. "A Review of "the Co-ordination and Regulation of Movements" by N. Bernstein. (pergamon Press, 1967.) [Pp. Xii + 196.] 505". In: *Ergonomics* (1968).

[38] Ginger Delmas, Philippe Weinzaepfel, Thomas Lucas, Francesc Moreno-Noguer, and Grégory Rogez. "PoseScript: 3D Human Poses from Natural Language". In: *European Conference on Computer Vision (ECCV)*. 2022.

[39] Ginger Delmas, Philippe Weinzaepfel, Francesc Moreno-Noguer, and Grégory Rogez. "PoseFix: Correcting 3D Human Poses with Natural Language". In: *International Conference on Computer Vision (ICCV)* (2023).

[40] J Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. 2019.

[41] Christian Diller and Angela Dai. "CG-HOI: Contact-Guided 3D Human-Object Interaction Generation". In: *Computer Vision and Pattern Recognition (CVPR)*. 2024.

[42] Markos Diomataris, Nikos Athanasiou, Omid Taheri, Xi Wang, Otmar Hilliges, and Michael J Black. "WANDR: Intention-guided Human Motion Generation". In: *Computer Vision and Pattern Recognition (CVPR)*. 2024.

[43] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations (ICLR)*. 2021.

[44] Yinglin Duan, Tianyang Shi, Zhengxia Zou, Yenan Lin, Zhehui Qian, Bohan Zhang, and Yi Yuan. "Single-shot Motion Completion with Transformer". In: *arXiv:2103.00776* (2021).

[45] Abhishek Dutta and Andrew Zisserman. "The VIA Annotation Software for Images, Audio and Video". In: *ACM International Conference on Multimedia (MM)*. 2019.

[46] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. "Sigmoid-weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning". In: *Neural Networks* (2018).

[47] Yao Feng, Jing Lin, Sai Kumar Dwivedi, Yu Sun, Priyanka Patel, and Michael J Black. "ChatPose: Chatting About 3D Human Pose". In: *Computer Vision and Pattern Recognition (CVPR)*. 2024.

[48] Wallace O Fenn. "A Cinematographic Study of Sprinters". In: *The Scientific Monthly* (1931).

[49] Mihai Fieraru, Mihai Zanfir, Silviu Cristian Pirlea, Vlad Olaru, and Cristian Sminchisescu. "AIFit: Automatic 3D Human-interpretable Feedback Models for Fitness Training". In: *Computer Vision and Pattern Recognition (CVPR)*. 2021.

[50] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. "Recurrent Network Models for Human Dynamics". In: *International Conference on Computer Vision (ICCV)*. 2015.

[51] Aphrodite Galata, Neil Johnson, and David Hogg. "Learning Variable-length Markov Models of Behavior". In: *Computer Vision and Image Understanding (CVIU)* (2001).

[52] D M Gavrila and L S Davis. “3-d Model-based Tracking of Humans in Action: A Multi-view Approach”. In: *Computer Vision and Pattern Recognition (CVPR)*. 1996.

[53] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, and Tanzer. “Gemini 1.5: Unlocking Multimodal Understanding across Millions of Tokens of Context”. In: *arXiv:2403.05530* (2024).

[54] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F Troje. “MoVi: A Large Multi-purpose Human Motion and Video Dataset”. In: *PloS one* (2021).

[55] Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. “Synthesis of Compositional Animations from Textual Descriptions”. In: *International Conference on Computer Vision (ICCV)*. 2021.

[56] Purvi Goel, Kuan-Chieh Wang, C Karen Liu, and Kayvon Fatahalian. “Iterative Motion Editing with Natural Language”. In: *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 2024.

[57] Kehong Gong, Dongze Lian, Heng Chang, Chuan Guo, X Zuo, Zihang Jiang, and Xinchao Wang. “TM2D: Bimodality Driven 3D Dance Generation Via Music-text Integration”. In: *International Conference on Computer Vision (ICCV)* (2023).

[58] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative Adversarial Nets”. In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2014.

[59] Anand Gopalakrishnan, Ankur Mali, Dan Kifer, Lee Giles, and Alexander G Ororbia. “A Neural Temporal Model for Human Motion Prediction”. In: *Computer Vision and Pattern Recognition (CVPR)*. 2019.

[60] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. "AVA: A Video Dataset of Spatio-temporally Localized Atomic Visual Actions". In: *Computer Vision and Pattern Recognition (CVPR)*. 2018.

[61] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. "Generating Diverse and Natural 3D Human Motions from Text". In: *Computer Vision and Pattern Recognition (CVPR)*. 2022.

[62] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. "TM2T: Stochastic and Tokenized Modeling for the Reciprocal Generation of 3D Human Motions and Texts". In: *European Conference on Computer Vision (ECCV)*. 2022.

[63] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, and Sun. "Action2Motion: Conditioned Generation of 3D Human Motions". In: *ACM International Conference on Multimedia (MM)*. 2020.

[64] Félix G Harvey and Christopher Pal. "Recurrent Transition Networks for Character Locomotion". In: *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 2018.

[65] Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. "Robust Motion In-betweening". In: *Transactions on Graphics (TOG)* (2020).

[66] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael Black. "Stochastic Scene-aware Motion Prediction". In: *International Conference on Computer Vision (ICCV)*. 2021.

[67] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael Black. "Resolving 3D Human Pose Ambiguities with 3D Scene Constraints". In: *International Conference on Computer Vision (ICCV)*. 2019.

[68] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. "Learning Joint Reconstruction of Hands and Manipulated Objects". In: *Computer Vision and Pattern Recognition (CVPR)*. 2019.

[69] Samitha Herath, Mehrtash Tafazzoli Harandi, and Fatih Porikli. "Going Deeper into Action Recognition: A Survey". In: *Image and Vision Computing (IVS)* (2017).

[70] Alejandro Hernandez, Jurgen Gall, and Francesc Moreno. "Human Motion Prediction Via Spatio-Temporal Inpainting". In: *International Conference on Computer Vision (ICCV)*. 2019.

[71] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "GANs Trained by a Two Time-scale Update Rule Converge to a Local Nash Equilibrium". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2017.

[72] Geoffrey E Hinton. "Using Relaxation to Find a Puppet". In: *Artificial Intelligence and the Simulation of Behaviour* (1976).

[73] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2020.

[74] S Hochreiter and J Schmidhuber. "Long Short-term Memory". In: *Neural computation* (1997).

[75] David Hogg. "Model-based Vision: A Program to See a Walking Person". In: *Image and Vision Computing (IVS)* (1983).

[76] Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. "Learned Motion Matching". In: *Transactions on Graphics (TOG)* (2020).

[77] Daniel Holden, Taku Komura, and Jun Saito. "Phase-functioned Neural Networks for Character Control". In: *Transactions on Graphics (TOG)* (2017).

[78] Daniel Holden, Jun Saito, and Taku Komura. "A Deep Learning Framework for Character Motion Synthesis and Editing". In: *Transactions on Graphics (TOG)* (2016).

[79] Yiming Huang, Weilin Wan, Yue Yang, Chris Callison-Burch, Mark Yatskar, and Lingjie Liu. "CoMo: Controllable Motion Generation through Language Guided Pose Code Editing". In: *European Conference on Computer Vision (ECCV)*. 2024.

[80] E Ilg, N Mayer, T Saikia, M Keuper, A Dosovitskiy, and T Brox. "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks". In: *Computer Vision and Pattern Recognition (CVPR)*. 2017.

[81] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. "Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2014).

[82] Nan Jiang, Hongjie Li, Ziye Yuan, Zimo He, Yixin Chen, Tengyu Liu, Yixin Zhu, and Siyuan Huang. "Dynamic Motion Blending for Versatile Motion Editing". In: *Computer Vision and Pattern Recognition (CVPR)*. 2025.

[83] S X Ju, M J Black, and Y Yacoob. "Cardboard People: A Parameterized Model of Articulated Image Motion". In: *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*. 2002.

[84] Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. "Guided Motion Diffusion for Controllable Human Motion Synthesis". In: *International Conference on Computer Vision (ICCV)*. 2023.

[85] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. "Geometric Skinning with Approximate Dual Quaternion Blending". In: *Transactions on Graphics (TOG)* (2008).

[86] Hyounghun Kim, Abhay Zala, Graham Burri, and Mohit Bansal. "FixMyPose: Pose Correctional Captioning and Retrieval". In: *AAAI Conference on Artificial Intelligence* (2021).

[87] Jihoon Kim, Taehyun Byun, Seungyoun Shin, and Won. "Conditional Motion In-betweening". In: *Pattern Recognition* (2022).

[88] Jihoon Kim, Jiseob Kim, and Sungjoon Choi. "FLAME: Free-form Language-based Motion Synthesis & Editing". In: *AAAI Conference on Artificial Intelligence* (2022).

[89] Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. "Variational Diffusion Models". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2021.

[90] Diederik P Kingma and Max Welling. "Auto-encoding Variational Bayes". In: *International Conference on Learning Representations (ICLR)*. 2014.

[91] Michael Kipp, Kerstin H Kipp, Alassane Ndiaye, and Patrick Gebhard. "Evaluating the Tangible Interface and Virtual Characters in the Interactive COHIBIT Exhibit". In: *ACM International Conference on Intelligent Virtual Agents*. 2006.

[92] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. "Activity Forecasting". In: *European Conference on Computer Vision (ECCV)*. 2012.

[93] Reinhard Klette and Garry Tee. "Understanding Human Motion: A Historic Review". In: *Human Motion*. 2008.

[94] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. "VIBE: Video Inference for Human Body Pose and Shape Estimation". In: *Computer Vision and Pattern Recognition (CVPR)*. 2020.

[95] Muhammed Kocabas, Ye Yuan, Pavlo Molchanov, Yunrong Guo, Michael J. Black, Otmar Hilliges, Jan Kautz, and Umar Iqbal. "PACE: Human and Camera Motion Estimation from in-the-wild Videos". In: *International Conference on 3D Vision (3DV)*. 2024.

[96] L A Kosman. "Aristotle's Definition of Motion". In: *Phronesis* (1969).

[97] Taras Kucherenko, Rajmund Nagy, Youngwoo Yoon, Jieyeon Woo, Teodor Nikolov, Mihail Tsakov, and Gustav Eje Henter. "The GENEA Challenge 2023: A Large-scale Evaluation of Gesture Generation Models in Monadic and Dyadic Settings". In: *Proceedings of theth International Conference on Multimodal Interaction*. 2023.

[98] Gilwoo Lee, Zhiwei Deng, Shugao Ma, Takaaki Shiratori, Siddhartha Srinivasa, and Yaser Sheikh. "Talking with Hands 16.2M: A Large-Scale Dataset of Synchronized Body-Finger Motion and Audio for Conversational Motion Analysis and Synthesis". In: *International Conference on Computer Vision (ICCV)*. 2019.

[99] Jehee Lee and Sung Yong Shin. "A Hierarchical Approach to Interactive Motion Editing for Human-like Figures". In: *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 1999.

[100] Taeryung Lee, Gyeongsik Moon, and Kyoung Mu Lee. "MultiAct: Long-Term 3D Human Motion Generation from Multiple Action Labels". In: *AAAI Conference on Artificial Intelligence*. 2022.

[101] Chuqiao Li, Julian Chibane, Yannan He, Naama Pearl, Andreas Geiger, and Gerard Pons-Moll. "Unimotion: Unifying 3D Human Motion Synthesis and Understanding". In: *International Conference on 3D Vision (3DV)*. 2025.

[102] Gen Li, Kaifeng Zhao, Siwei Zhang, Xiaozhong Lyu, Mihai Dusmanu, Yan Zhang, Marc Pollefeys, and Siyu Tang. "EgoGen: An Egocentric Synthetic Data Generator". In: *Computer Vision and Pattern Recognition (CVPR)*. 2024.

[103] Jiaman Li, Alexander Clegg, Roozbeh Mottaghi, Jiajun Wu, Xavier Puig, and C Karen Liu. "Controllable Human-object Interaction Synthesis". In: *European Conference on Computer Vision (ECCV)*. 2024.

[104] Jiaman Li, Jiajun Wu, and C Karen Liu. "Object Motion Guided Human Motion Synthesis". In: *Transactions on Graphics (TOG)* (2023).

[105] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. "AI Choreographer: Music Conditioned 3D Dance Generation with AIST++". In: *International Conference on Computer Vision (ICCV)*. 2021.

[106] Shuai Li, Sisi Zhuang, Wenfeng Song, Xinyu Zhang, Hejia Chen, and Aimin Hao. "Sequential Texts Driven Cohesive Motions Synthesis with Natural Transitions". In: *International Conference on Computer Vision (ICCV)*. 2023.

[107] Angela S Lin, Lemeng Wu, Rodolfo Corona, Kevin Tai, Qixing Huang, and Raymond J Mooney. "Generating Animated Videos of Human Activities from Natural Language Descriptions". In: *Conference on Neural Information Processing Systems Workshops (NeurIPSW)*. 2018.

[108] Jing Lin, Yao Feng, Weiyang Liu, and Michael J Black. "ChatHuman: Language-driven 3D Human Understanding with Retrieval-augmented Tool Reasoning". In: *arXiv:2405.04533* (2024).

[109] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. "Motion-X: A Large-scale 3D Expressive Whole-body Human Motion Dataset". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2023.

[110] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. "Character Controllers Using Motion VAEs". In: *Transactions on Graphics (TOG)* (2020).

[111] Haiyang Liu, Zihao Zhu, Giorgio Becherini, Yichen Peng, Mingyang Su, You Zhou, Xuefei Zhe, Naoya Iwamoto, Bo Zheng, and Michael J Black. "EMAGE: Towards Unified Holistic Co-Speech Gesture Generation Via Expressive Masked Audio Gesture Modeling". In: *Computer Vision and Pattern Recognition (CVPR)*. 2024.

[112] Haiyang Liu, Zihao Zhu, Naoya Iwamoto, Yichen Peng, Zhengqing Li, You Zhou, Elif Bozkurt, and Bo Zheng. "BEAT: A Large-scale

Semantic and Emotional Multi-modal Dataset for Conversational Gestures Synthesis". In: *European Conference on Computer Vision (ECCV)*. 2022.

[113] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. "Visual Instruction Tuning". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2023.

[114] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. "NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2020).

[115] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows". In: *International Conference on Computer Vision (ICCV)* (2021).

[116] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. "SMPL: A Skinned Multi-Person Linear Model". In: *Transactions on Graphics (TOG)* (2015).

[117] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations (ICLR)*. 2019.

[118] Lingni Ma, Yuting Ye, Fangzhou Hong, Vladimir Guzov, Yifeng Jiang, Rowan Postyeni, Luis Pesqueira, Alexander Gamino, Vijay Baiyya, Hyo Jin Kim, Kevin Bailey, David Soriano Fosas, C Karen Liu, Ziwei Liu, Jakob Engel, Renzo De Nardi, and Richard Newcombe. "Nymeria: A Massive Collection of Multimodal Egocentric Daily Motion in the Wild". In: *arXiv:2406.09905* (2024).

[119] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data Using t-SNE". In: *Journal of Machine Learning Research (JMLR)* (2008).

[120] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael Black. "AMASS: Archive of Motion

Capture as Surface Shapes". In: *International Conference on Computer Vision (ICCV)*. 2019.

[121] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. "Generating Images from Captions with Attention". In: *International Conference on Learning Representations (ICLR)*. 2015.

[122] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. "Weakly-supervised Action Transition Learning for Stochastic Human Motion Prediction". In: *Computer Vision and Pattern Recognition (CVPR)*. 2022.

[123] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. "Learning Trajectory Dependencies for Human Motion Prediction". In: *International Conference on Computer Vision (ICCV)*. 2019.

[124] A A Markov. "Extension of the Law of Large Numbers to Dependent Events". In: *Bulletin of the Society of the Physics Mathematics, Kazan* (1906).

[125] D Marr and H K Nishihara. "Representation and Recognition of the Spatial Organization of Three-dimensional Shapes". In: *Proceedings of the Royal Society of London* (1978).

[126] E C Marsi and F van Rooden. "Expressing Uncertainty with a Talking Head in a Multimodal Question-answering System". In: *Proceedings of the workshop on multimodal output generation (MOG)*. 2007.

[127] Julieta Martinez, Michael J Black, and Javier Romero. "On Human Motion Prediction Using Recurrent Neural Networks". In: *Computer Vision and Pattern Recognition (CVPR)*. 2017.

[128] Ian Mason, Sebastian Starke, and Taku Komura. "Real-time Style Modelling of Human Locomotion Via Feature-wise Transformations and Local Motion Phases". In: *Proceedings of the ACM on Computer Graphics and Interactive Techniques (i3D)* (2022).

[129] Delle Rae Maxwell. "Graphical Marionette : A Modern-day Pinocchio". 1983.

[130] Tomas Mikolov, Ilya Sutskever, Kai Chen, and Corrado. "Distributed Representations of Words and Phrases and their Compositionality". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2013.

[131] Aymen Mir, Xavier Puig, Angjoo Kanazawa, and Gerard Pons-Moll. "Generating Continual Human Motion in Diverse 3D Scenes". In: *International Conference on 3D Vision (3DV)*. 2024.

[132] Davide Moltisanti, Jinyi Wu, Bo Dai, and Chen Change Loy. "BRACE: The Breakdancing Competition Dataset for Dance Motion Synthesis". In: *European Conference on Computer Vision (ECCV)*. 2022.

[133] Susanne van Mulken, Elisabeth André, and Jochen Müller. "The Persona Effect: How Substantial Is It?" In: *International Conference on Human-Computer Interaction*. 1998.

[134] Alex Nichol, Prafulla Dhariwal, A Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, I Sutskever, and Mark Chen. "GLIDE: Towards Photorealistic Image Generation and Editing with Text-guided Diffusion Models". In: *International Conference on Machine Learning (ICML)* (2021).

[135] Dirk Ormoneit, Michael J Black, Trevor Hastie, and Hedvig Kjellström. "Representing Cyclic Human Motion Using Functional Analysis". In: *Image and Vision Computing (IVS)* (2005).

[136] Dirk Ormoneit, Hedvig Sidenbladh, Michael Black, and Trevor Hastie. "Learning and Tracking Cyclic Human Motion". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2000.

[137] Ahmed A A Osman, Timo Bolkart, and Michael J Black. "STAR: A Sparse Trained Articulated Human Body Regressor". In: *European Conference on Computer Vision (ECCV)*. 2020.

[138] Ahmed A A Osman, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. "SUPR: A Sparse Unified Part-Based Human Body Model". In: *European Conference on Computer Vision (ECCV)*. 2022.

[139] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. "Normalizing Flows for Probabilistic Modeling and Inference". In: *Journal of Machine Learning Research (JMLR)* (2021).

[140] Priyanka Patel, Chun-Hao P Huang, Joachim Tesch, David T Hoffmann, Shashank Tripathi, and Michael J Black. "AGORA: Avatars in Geography Optimized for Regression Analysis". In: *Computer Vision and Pattern Recognition (CVPR)*. 2021.

[141] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A A Osman, Dimitrios Tzionas, and Michael J Black. "Expressive Body Capture: 3D Hands, Face, and Body from a Single Image". In: *Computer Vision and Pattern Recognition (CVPR)*. 2019.

[142] Dario Pavllo, Christoph Feichtenhofer, Michael Auli, and David Grangier. "Modeling Human Motion with Quaternion-based Neural Networks". In: *International Journal of Computer Vision (IJCV)* (2020).

[143] Dario Pavllo, David Grangier, and Michael Auli. "QuaterNet: A Quaternion-based Recurrent Model for Human Motion". In: *British Machine Vision Conference (BMVC)*. 2018.

[144] Xiaogang Peng, Yiming Xie, Zizhao Wu, Varun Jampani, Deqing Sun, and Huaizu Jiang. "HOI-diff: Text-driven Synthesis of 3D Human-object Interactions Using Diffusion Models". In: *arXiv:2312.06553* (2023).

[145] K Perlin. "Real Time Responsive Animation with Personality". In: *IEEE transactions on visualization and computer graphics* (1995).

[146] Mathis Petrovich, Michael J Black, and Gül Varol. "Action-Conditioned 3D Human Motion Synthesis with Transformer VAE". In: *International Conference on Computer Vision (ICCV)*. 2021.

[147] Mathis Petrovich, Michael J Black, and Gül Varol. "TEMOS: Generating Diverse Human Motions from Textual Descriptions". In: *arXiv:2204.14109* (2022).

[148] Mathis Petrovich, Michael J Black, and Gül Varol. "TMR: Text-to-Motion Retrieval Using Contrastive 3D Human Motion Synthesis". In: *International Conference on Computer Vision (ICCV)*. 2023.

[149] Mathis Petrovich, Or Litany, Umar Iqbal, Michael J Black, Gül Varol, Xue Bin Peng, and Davis Rempe. "Multi-Track Timeline Control for Text-Driven 3D Human Motion Generation". In: *Computer Vision and Pattern Recognition (CVPR)*. 2024.

[150] Lyndsey C Pickup, Zheng Pan, Donglai Wei, Yichang Shih, Changshui Zhang, Andrew Zisserman, Bernhard Scholkopf, and William T Freeman. "Seeing the Arrow of Time". In: *Computer Vision and Pattern Recognition (CVPR)*. 2014.

[151] Leonid Pishchulin, Arjun Jain, Christian Wojek, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. "Learning People Detection Models from Few Training Samples". In: *Computer Vision and Pattern Recognition (CVPR)*. 2011.

[152] Matthias Plappert, Christian Mandery, and Tamim Asfour. "The KIT Motion-Language Dataset". In: *BigData* (2016).

[153] Ralf Plänkers and Pascal Fua. "Tracking and Modeling People in Video Sequences". In: *Computer Vision and Image Understanding (CVIU)* (2001).

[154] Zoran Popović and Andrew Witkin. "Physically Based Motion Transformation". In: *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 1999.

[155] Junfu Pu and Ying Shan. "Music-driven Dance Regeneration with Controllable Key Pose Constraints". In: *arXiv:2207.03682* (2022).

[156] Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. "BABEL: Bodies, Action and Behavior with English Labels". In: *Computer Vision and Pattern Recognition (CVPR)*. 2021.

[157] Alec Radford, Jong Wook Kim, Chris Hallacy, A Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and I Sutskever. "Learning Transferable Visual Models from Natural Language Supervision". In: *International Conference on Machine Learning (ICML)* (2021).

[158] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. "Hierarchical Text-conditional Image Generation with CLIP Latents". In: *arXiv:2204.06125* (2022).

[159] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. "Zero-shot Text-to-image Generation". In: *International Conference on Machine Learning (ICML)*. 2021.

[160] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. "Generative Adversarial Text to Image Synthesis". In: *International Conference on Machine Learning (ICML)*. 2016.

[161] Danilo Jimenez Rezende and Shakir Mohamed. "Variational Inference with Normalizing Flows". In: *International Conference on Machine Learning (ICML)*. 2015.

[162] Grégory Rogez and C Schmid. "MoCap-guided Data Augmentation for 3D Pose Estimation in the Wild". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2016.

[163] K Rohr. "Towards Model-based Recognition of Human Movements in Image Sequences". In: *Computer Vision and Image Understanding (CVIU)* (1994).

[164] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. "High-resolution Image Synthesis

with Latent Diffusion Models". In: *Computer Vision and Pattern Recognition (CVPR)*. 2022.

[165] Javier Romero, Dimitrios Tzionas, and Michael J Black. "Embodied Hands: Modeling and Capturing Hands and Bodies Together". In: *Transactions on Graphics (TOG)* (2017).

[166] Charles Rose, Brian Guenter, Bobby Bodenheimer, and Michael F Cohen. "Efficient Generation of Motion Transitions Using Spacetime Constraints". In: *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 1996.

[167] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning Representations by Back-propagating Errors". In: *Nature* (1986).

[168] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S S Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. "Photorealistic Text-to-image Diffusion Models with Deep Language Understanding". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2022.

[169] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. "Improved Techniques for Training GANs". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2016.

[170] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. "DistilBERT, a Distilled Version of BERT: Smaller, Faster". In: *arXiv:1910.01108* (2019).

[171] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. "LAION-5B: An open large-scale

dataset for training next generation image-text models". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2022.

[172] Jan Sedmidubsky, Petr Elias, and Pavel Zezula. "Benchmarking Search and Annotation in Continuous Human Skeleton Sequences". In: *ACM International Conference on Multimedia Retrieval (ICMR)*. 2019.

[173] Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H Bermano. "Human Motion Diffusion as a Generative Prior". In: *International Conference on Learning Representations (ICLR)*. 2024.

[174] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis". In: *Computer Vision and Pattern Recognition (CVPR)*. 2016.

[175] Soyong Shin, Juyong Kim, Eni Halilaj, and Michael J. Black. "WHAM: Reconstructing World-grounded Humans with Accurate 3D Motion". In: *Computer Vision and Pattern Recognition (CVPR)*. 2024.

[176] Ken Shoemake. "Animating Rotation with Quaternion Curves". In: *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 1985.

[177] Hedvig Sidenbladh, Michael J Black, and David J Fleet. "Stochastic Tracking of 3D Human Figures Using 2D Image Motion". In: *European Conference on Computer Vision (ECCV)*. 2000.

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

[179] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: *International Conference on Learning Representations (ICLR)*. 2021.

[180] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations (ICLR)*. 2021.

[181] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. "Neural State Machine for Character-scene Interactions". In: *Transactions on Graphics (TOG)* (2019).

[182] Sanjay Subramanian, Evonne Ng, Lea Müller, Dan Klein, Shiry Ginosar, and Trevor Darrell. "Pose Priors from Language Models". In: *arXiv:2405.03689* (2024).

[183] Omid Taheri, Nima Ghorbani, Michael J Black, and Tzionas. "GRAB: A Dataset of Whole-Body Human Grasping of Objects". In: *European Conference on Computer Vision (ECCV)*. 2020.

[184] Julian Tanke, Andreas Weber, and Juergen Gall. "Human Motion Anticipation with Symbolic Label". In: *arXiv:1912.06079* (2019).

[185] Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. "Modeling Human Motion Using Binary Latent Variables". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2007.

[186] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. "MotionCLIP: Exposing Human Motion Generation to Clip Space". In: *European Conference on Computer Vision (ECCV)*. 2022.

[187] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. "Human Motion Diffusion Model". In: *International Conference on Learning Representations (ICLR)* (2023).

[188] Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. "xR–EgoPose: Egocentric 3D human pose from an HMD camera". In: *Computer Vision and Pattern Recognition (CVPR)*. 2019.

[189] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. "LLaMA: Open and Efficient Foundation Language Models". In: *arXiv:2302.13971* (2023).

[190] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. "Llama 2: Open Foundation and Fine-tuned Chat Models". In: *arXiv:2307.09288* (2023).

[191] Jonathan Tseng, Rodrigo Castellon, and C Karen Liu. "EDGE: Editable Dance Generation from Music". In: *Computer Vision and Pattern Recognition (CVPR)*. 2023.

[192] Raquel Urtasun, David J Fleet, and Neil D Lawrence. "Modeling Human Locomotion with Topologically Constrained Latent Variable Models". In: *Human Motion – Understanding, Modeling, Capture and Animation*. 2007.

[193] Guillermo Valle-Pérez, Gustav Eje Henter, Jonas Beskow, Andre Holzapfel, Pierre-Yves Oudeyer, and Simon Alexanderson. "Transflower: Probabilistic Autoregressive Dance Generation with Multimodal Attention". In: *Transactions on Graphics (TOG)* (2021).

[194] Gül Varol, Ivan Laptev, Cordelia Schmid, and Andrew Zisserman. "Synthetic Humans for Action Recognition from Unseen Viewpoints". In: *International Journal of Computer Vision (IJCV)*. 2021.

[195] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. "Learning from Synthetic Humans". In: *Computer Vision and Pattern Recognition (CVPR)*. 2017.

[196] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention Is All You Need". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2017.

[197] Pascal Vincent. "A Connection between Score Matching and Denoising Autoencoders". In: *Neural Computation* (2011).

[198] Jordan Voas, Yili Wang, Qixing Huang, and Raymond Mooney. "What Is the Best Automated Metric for Text to Motion Generation?" In: *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 2023.

[199] Jingbo Wang, Yu Rong, Jingyuan Liu, Sijie Yan, Dahua Lin, and Bo Dai. "Towards Diverse and Natural Scene-aware 3D Human Motion Synthesis". In: *Computer Vision and Pattern Recognition (CVPR)* (2022).

[200] Jingbo Wang, Yu Rong, Jingyuan Liu, Sijie Yan, Dahua Lin, and Bo Dai. "Towards Diverse and Natural Scene-aware 3D Human Motion Synthesis". In: *Computer Vision and Pattern Recognition (CVPR)*. 2022.

[201] Xi Wang, Gen Li, Yen-Ling Kuo, Muhammed Kocabas, Emre Aksan, and Otmar Hilliges. "Reconstructing Action-Conditioned Human-Object Interactions Using Commonsense Knowledge Priors". In: *International Conference on 3D Vision (3DV)*. 2022.

[202] Ye Wang, Sipeng Zheng, Bin Cao, Qianshan Wei, Qin Jin, and Zongqing Lu. "Quo Vadis, Motion Generation? from Large Language Models to Large Motion Models". In: *arXiv:2410.03311* (2024).

[203] Zan Wang, Yixin Chen, Baoxiong Jia, Puhao Li, Jinlu Zhang, Jingze Zhang, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. "Move as You Say, Interact as You Can: Language-guided Human Motion Generation with Scene Affordance". In: *arXiv:2403.18036* (2024).

[204] Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. "HUMANISE: Language-conditioned Human Motion Generation in 3D Scenes". In: *Conference on Neural Information Processing Systems (NeurIPS)* (2022).

[205] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. "Finetuned Language Models Are Zero-shot Learners". In: *International Conference on Learning Representations (ICLR)* (2021).

[206] Andrew Witkin and Michael Kass. "Spacetime Constraints". In: *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 1988.

[207] Zhen Wu, Jiaman Li, Pei Xu, and C Karen Liu. "Human-object Interaction from Human-level Instructions". In: *arXiv:2406.17840* (2024).

[208] Zeqi Xiao, Tai Wang, Jingbo Wang, Jinkun Cao, Wenwei Zhang, Bo Dai, Dahua Lin, and Jiangmiao Pang. "Unified Human-scene Interaction Via Prompted Chain-of-Contacts". In: *arXiv:2309.07918* (2023).

[209] Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. "OmniControl: Control Any Joint at Any Time for Human Motion Generation". In: *International Conference on Learning Representations (ICLR)*. 2024.

[210] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. "GHUM & GHUML: Generative 3D Human Shape and Articulated Pose Models". In: *Computer Vision and Pattern Recognition (CVPR)*. 2020.

[211] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. "MSR-VTT: A Large Video Description Dataset for Bridging Video and Language". In: *Computer Vision and Pattern Recognition (CVPR)*. 2016.

[212] Liang Xu, Ziyang Song, Dongliang Wang, Jing Su, Zhicheng Fang, Chenjing Ding, Weihao Gan, Yichao Yan, Xin Jin, Xiaokang Yang, Wenjun Zeng, and Wei Wu. "ActFormer: A GAN-based Transformer towards General Action-Conditioned 3D Human Motion Generation". In: *International Conference on Computer Vision (ICCV)*. 2023.

[213] Sijie Yan, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin. "Convolutional Sequence Generation for Skeleton-based Action Synthesis". In: *International Conference on Computer Vision (ICCV)*. 2019.

[214] Sicheng Yang, Zhiyong Wu, Minglei Li, Zhensong Zhang, Lei Hao, Weihong Bao, Ming Cheng, and Long Xiao. "DiffuseStyleGesture: Stylized Audio-driven Co-speech Gesture Generation with Diffusion Models". In: *AAAI Conference on Artificial Intelligence*. 2023.

[215] Xu Yang, Hanwang Zhang, and Jianfei Cai. "Shuffle-Then-Assemble: Learning Object-Agnostic Visual Relationship Features". In: *European Conference on Computer Vision (ECCV)*. 2018.

[216] Payam Jome Yazdian, Mo Chen, and Angelica Lim. "Gesture2Vec: Clustering Gestures Using Representation Learning Methods for

Co-speech Gesture Generation". In: *International Conference on Intelligent Robots and Systems (IROS)*. 2022.

[217] Hongwei Yi, Hualin Liang, Yifei Liu, Qiong Cao, Yandong Wen, Timo Bolkart, Dacheng Tao, and Michael J Black. "Generating Holistic 3D Human Motion from Speech". In: *Computer Vision and Pattern Recognition (CVPR)*. 2023.

[218] Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. "Visual Relationship Detection with Internal and External Linguistic Knowledge Distillation". In: *International Conference on Computer Vision (ICCV)*. 2017.

[219] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis E H Tay, Jiashi Feng, and Shuicheng Yan. "Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet". In: *International Conference on Computer Vision (ICCV)* (2021).

[220] Ye Yuan and Kris Kitani. "Diverse Trajectory Forecasting with Determinantal Point Processes". In: *arXiv:1907.04967* (2019).

[221] Ye Yuan and Kris Kitani. "DLow: Diversifying Latent Flows for Diverse Human Motion Prediction". In: *European Conference on Computer Vision (ECCV)*. 2020.

[222] Canyu Zhang, Youbao Tang, Ning Zhang, Ruei-Sung Lin, Mei Han, Jing Xiao, and Song Wang. "Bidirectional Autoregessive Diffusion Model for Dance Generation". In: *Computer Vision and Pattern Recognition (CVPR)*. 2024.

[223] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. "T2M-GPT: Generating Human Motion from Textual Descriptions with Discrete Representations". In: *arXiv:2301.06052* (2023).

[224] Mingyuan Zhang, Zhongang Cai, Liang Pan, and Hong. "MotionDiffuse: Text-driven Human Motion Generation with Diffusion Model". In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2023).

[225] Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. "FineMoGen: Fine-Grained Spatio-Temporal Motion Generation and Editing". In: *Conference on Neural Information Processing Systems (NeurIPS)*. 2023.

[226] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. "Multi-scale Vision Longformer: A New Vision Transformer for High-resolution Image Encoding". In: *International Conference on Computer Vision (ICCV)*. 2021.

[227] Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming-Yu Liu. "DiffCollage: Parallel Generation of Large Content with Diffusion Models". In: *arXiv:2303.17076* (2023).

[228] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. "COUCH: Towards Controllable Human-chair Interactions". In: *European Conference on Computer Vision (ECCV)*. 2022.

[229] Xinyi Zhang and Michiel van de Panne. "Data-driven Autocompletion for Keyframe Animation". In: *Annual ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG)*. 2018.

[230] Yan Zhang, Michael J Black, and Siyu Tang. "We Are More Than Our Joints: Predicting How 3D Bodies Move". In: *Computer Vision and Pattern Recognition (CVPR)*. 2021.

[231] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. "HACS: Human Action Clips and Segments Dataset for Recognition and Temporal Localization". In: *International Conference on Computer Vision (ICCV)*. 2019.

[232] Rui Zhao, Hui Su, and Qiang Ji. "Bayesian Adversarial Human Motion Synthesis". In: *Computer Vision and Pattern Recognition (CVPR)*. 2020.

[233] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. "On the Continuity of Rotation Representations in Neural Networks". In: *Computer Vision and Pattern Recognition (CVPR)*. 2019.

[234] Yi Zhou, Jingwan Lu, Connelly Barnes, Jimei Yang, Sitao Xiang, and Hao Li. "Generative Tweening: Long-term Inbetweening of 3D Human Motions". In: *arXiv:2005.08891* (2020).

[235] Qiran Zou, Shangyuan Yuan, Shian Du, Yu Wang, Chang Liu, Yi Xu, Jie Chen, and Xiangyang Ji. "ParCo: Part-Coordinating Text-to-Motion Synthesis". In: *European Conference on Computer Vision (ECCV)*. 2024.

[236] Shihao Zou, Xinxin Zuo, Sen Wang, Yiming Qian, Chuan Guo, and Li Cheng. "Human Pose and Shape Estimation from Single Polarization Images". In: *IEEE Transactions on Multimedia* (2022).

Controlling 3D human motion through natural language is key to interactive experiences in animation, gaming, and virtual reality. This thesis argues that truly controllable motion generation requires compositional thinking: chaining actions over time, layering them across body parts, and refining them through iterative editing. It presents methods and datasets that tackle each of these axes — temporal composition (TEACH), spatial composition (SINC), and text-based motion editing (MotionFix) — advancing language-driven motion generation toward the compositional, editable control that real-world applications demand.

PERCEIVING SYSTEMS
MAX PLANCK INSTITUTE FOR
INTELLIGENT SYSTEMS