Deep Learning for 3D Human Action Modeling and Understanding

by

Chuan Guo

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering

University of Alberta

Abstract

Studying 3D human actions is a fundamental task in computer graphics, computer vision, and robotics, with a broad range of applications such as VR/AR, AAA gaming, filmmaking, and artistic creation. Conventional approaches often necessitate labor-intensive manual intervention, consuming substantial time and resources. While deep learning has fundamentally transformed the analysis of many other visual modalities like images and videos, its potential in human action analysis remains largely unexplored. In this thesis, we address this gap by developing solutions and resources for modeling and understanding human motions using deep generative models.

Firstly, we collect and annotate two large-scale multimodal human action datasets: (i) HumanAct12, which consists of 1,191 motion clips and 90,099 frames, annotated with 12 coarse-grained and 34 fine-grained action classes; and (ii) HumanML3D, which contains 44,970 textual descriptions and 14,616 motions, totaling 28.59 hours of data. Secondly, we investigate the use of neural networks designed to accommodate various input modalities for motion generation, including action categories (action2motion), textual descriptions (text2motion), and style cues (motion stylization). Distinct from previous deterministic approaches to motion synthesis, our methods properly adopt a series of Variational Autoencoder (VAE)-base frameworks, which fully recognize the inherently stochastic nature of human movements, thereby achieving diverse and natural 3D human motion generation from various conditions. Thirdly, we develop an automated character animation pipeline that transfers the motions to the character in a single image (motion2video). This is accomplished through reconstructing the 3D shape and texture of the character from input image, rigging, animating, and rendering the 3D sequence into 2D video. Fourthly, we explore the reciprocal relationship between human motion and human language (i.e, text), offering a unified framework for both motion understanding and generation. To bridge the gap between human motion and text modalities, we introduce the concept of discrete motion tokens – a novel motion representation formed through deep vector quantization techniques. These motion tokens allow for the seamless translation of human motion into natural language, akin to neural machine translation processes. Furthermore, we present a inverse alignment technique, and demonstrate that understanding motions facilitates more accurate and robust motion generation.

Preface

This thesis is an original work by Chuan Guo.

Chapter 3 of this thesis has been published under the title [1] **Chuan Guo**, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. *Action2motion: Conditioned generation of 3D human motions.* In Proceedings of the 28th ACM International Conference on Multimedia, pages 2021–2029, 2020. The full version of this work has been published under the title [2] **Chuan Guo**, Xinxin Zuo, Sen Wang, Xinshuang Liu, Shihao Zou, Minglun Gong, and Li Cheng. *Action2video: Generating videos of human 3D actions.* International Journal of Computer Vision, 130(2):285–315, 2022. This research problem was identified by the collaboration of me, my supervisor Prof. Li Cheng, Xinxin Zuo and Sen Wang. Then I led the dataset annotation with the assistance from Annan Deng, Qingyao Sun. I was the primary contributor to the technical design and experiments, with support from Xinshuang Liu and Shihao Zou. Finally, I finished the paper writing with great help from Prof. Li Cheng and Prof. Minglun Gong.

Chapter 4 of this thesis has been published under the title [3] **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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 5152–5161, June 2022. This research project was initialized and defined by me, Li Cheng and Xinxin Zuo. I primarily collected the dataset, with casual discussion with Li Cheng, XinXin Zuo and Sen Wang. I also designed the technical approach and experiments, with support from Shihao Zou and Sen Wang. Prof. Xingyu Li and Prof. Li Cheng helped me on the paper writing.

Chapter 5 of this thesis has been accepted to The International Conference on Learning Representations under the title [4] **Chuan Guo**, Yuxuan Mu, Xinxin Zuo, Peng Dai, Youliang Yan, Juwei Lu, and Li Cheng. *Generative human motion stylization in latent space.* In The Twelfth International Conference on Learning Representations, 2024. This research problem was identified through the collaboration of me, Xinxin Zuo, Juwei Lu and Li Cheng. I was the principal contributor to the motion data processing, technical design, experiments, and paper writing. Yuxuan Mu helped me on implementing baseline methods and results visualization. Peng Dai and Youliang Yan provided me suggestions on paper writing.

Chapter 6 of this thesis has been published under the title [5] **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, 2022. I initialized this research project, with discussion with Xinxin Zuo and Sen Wang. I was the principal contributor to the technical design, experiments, evaluation metrics, and paper writing. Xinxin Zuo, Sen Wang and Li Cheng helped me on polishing the paper writing.

Acknowledgements

This thesis would not be possible without the support and assistance of many people.

My first and foremost gratitude goes to my advisor, Professor Li Cheng, for his generous support and guidance during my Ph.D. journey. His dedication to rigorous research and his open-minded approach have not only been instructive but also inspiring. I am deeply appreciative of the countless hours he dedicated to meticulously reviewing and revising all my paper submissions and scholarship applications. I was also honored to be funded through Alberta Graduate Excellence Scholarship and Alberta Innovate Graduate Scholarship.

My sincere thanks extend to the esteemed members of the University of Alberta faculty, especially my thesis committee members: Professor Hai Jiang, Professor Marek Reformat, Professor Xingyu Li, Professor Dale Schuurmans, Professor Greg Mori and Professor Lili Mou. Their guidance and investment of time in overseeing my studies and thesis are deeply appreciated. I would also like to express my gratitude to Professor Xingyu Li and Professor Minglun Gong for their support of my scholarship applications.

I am fortunate to work with the incredible individuals at the Vision and Learning Lab. My early years were brightened by Taivanbat Badamdorj, who brought laughter and joy into my life. I also cherish the warm memories of our group activities and the collaborations with Shihao Zou, Wei Ji, Jingjing Li, Yilin Wang, Yuxuan Mu, Size Wang and Gohar. Our shared moments, from research to leisurely games, will remain etched in my memory. I extend special thanks to my mentors, Xinxin Zuo, Sen Wang, and Shuang Wu. Without your invaluable guidance, expertise, and insights, these accomplishments would not have been possible. Collaboration with interns Qingyao Sun, Annan

Deng, and Xinshuang Liu brought forth substantial contributions to my research projects and is greatly appreciated.

To my friends in Edmonton, your companionship has added fun and joy to my Ph.D. journey. Thanks to Bernard Mou, Yingnan Wang, Gongyu Li, and many others. I want to express special thanks to Jiachen Wang, for his warmest support and help in my toughest lifetime.

My deepest gratitude is reserved for my family and my partner, who offered their unconditional love and unwavering support. They have been a constant source of inspiration and solid backup. My partner, Di, deserves special recognition for encouraging me to stay true to myself. I hold a deep hope that my grandparents could witness this moment. Their boundless love raised me and instilled in me the values of humanity. Although they are no longer with us, the memory of their love remains eternally with me.

Lastly, I wish to express gratitude to myself. Growing up in a small, remote, and impoverished village where most individuals discontinued their education after high school, I could never have envisioned standing on this stage, pursuing a Ph.D. degree. I appreciate the tireless effort and unrelenting dedication I have invested, and I hope to continue this journey of growth indefinitely.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Human actions play an important role in our daily activities. We continuously translate our thoughts and intentions into actions, such as a simple wave as a greeting or the act of pulling up a chair to sit and rest. Meanwhile, our actions transcend mere functionality; they imply our emotions, physical well-being, intentions, and even personalities. For example, by perceiving the way an individual walks down the street, we can not only recognize the person but also tell their emotional state, whether they are happy or sad. Studying human actions therefore becomes a subject of great interest in computer vision and computer graphics. The pursuit of understanding and modeling human actions carries profound implications in various application domains, spanning from animation [10, 13], healthcare [14], and sports science [15, 16, 17], to human-computer interaction [18, 19]. In particular, we have recently witnessed a surge of practices in the digital analysis and simulation of human actions for lifelike 3D character animations in movies, video games, and VR/AR, as well as public safety through anomaly detection. However, these efforts often demand substantial investments in resources, labor, and time. Take 3D character animation as an example. It involves intricate steps of 3D motion capture, avatar creation and modeling in 3D engine, as well as rendering these animations to produce video frames. This highlights the need for more streamlined and efficient approaches.

Expressive deep learning models have revolutionized the visual analysis in many domains, such

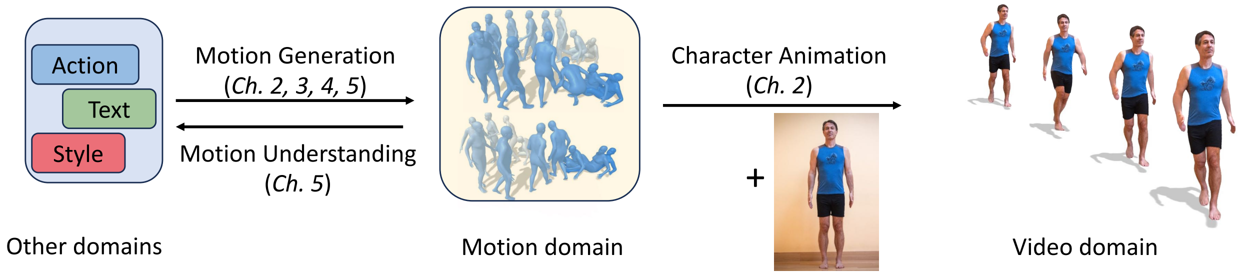


Figure 1.1: Thesis overview. The central themes of this thesis are human motion generation, single image-based character animation, and motion understanding.

as image/video generation [20, 21, 22, 23], understanding [24, 25, 26, 27], manipulation [28, 29], and recognition [30, 31, 32]. Unlike most traditional approaches which rely on predefined rules or manual intervention, deep learning methods gain insights and learn the patterns from vast data, which enables remarkable scalability and efficiency during inference. Nevertheless, when it comes to human actions, the problem is far away from being resolved. To fill this gap, as illustrated in Fig. 1.1, we delve into the deep learning-based analysis of 3D human actions, providing solutions and resources for automated human action modeling and understanding.

In what follows, we elaborate on the motivations and objectives of main components of this thesis in Section 1.1, followed by a summary of contribution in Section 1.2 and an outline of the thesis in Section 1.3.

1.1 Motivations and Objectives

As shown in Fig. 1.1, the central theme of this thesis is on 3D human motions, including interactions with other domains such as natural language. Our research topics have close implications for several practical applications including human motion acquisition, editing, interpretation, and character animation.

1.1.1 Human Motion Generation

The acquisition of 3D human motions is a fundamental step of 3D animation and analysis. However, obtaining such assets is usually beyond the means of the average user. It typically requires sophisticated motion sensors (e.g., optical marker, IMU) as well as advanced cameras (e.g., depth camera) for capturing real-world motions performed by actors. Alternatively, motions can also be manually crafted, but it also demands the expertise in keyframing animation and physical simulation. In terms of these challenges, we seek to develop deep learning models that facilitate the generation of natural 3D human motions from user-friendly inputs.

Expressing our thoughts and intentions through language or categorical attributes is an inherent skill of human beings. This premise motivates our first two projects—learning deep models for motion synthesis based on categorical action and textual description. While prior efforts [33, 34, 35, 36, 37, 38, 39] have made commendable progresses in this direction, several common limitations persist, including the motions limited in 2D space, distorted poses, and a lack of temporal consistency. Furthermore, the motion contents are not semantically well-aligned with the conditional input.

Zooming in on these observations, we find the principal reasons are several-fold. Firstly, existing annotated human motion datasets are typically limited in either quantity or quality. For instance, action-based motion analysis such as UTKinect-Action [40] and MSR-Action3D [41] only contain 200-500 human action sequences. NTU-RGBD [42], although considered as large dataset with 100,000 motions across 120 action classes, their joint positions are acquired from Kinect-I cameras, which are notorious for their inaccuracies. In terms of the motion-language dataset, the only available dataset KIT Motion-Language [43], comprising 3,911 motion sequences and 6,278 sentences, mostly focuses on locomotion movements such as walking and running. Secondly, as shown in Fig. 1.2 (a), previous approaches typically formulate the task of motion generation as deterministic one-to-one mapping, assuming one possible outcome given condition input. However, human motion is inherently stochastic. Imagine a person running on a playground, there are a spectrum of possible future trajectories to which this person can take. Failing to capture this stochasticity would

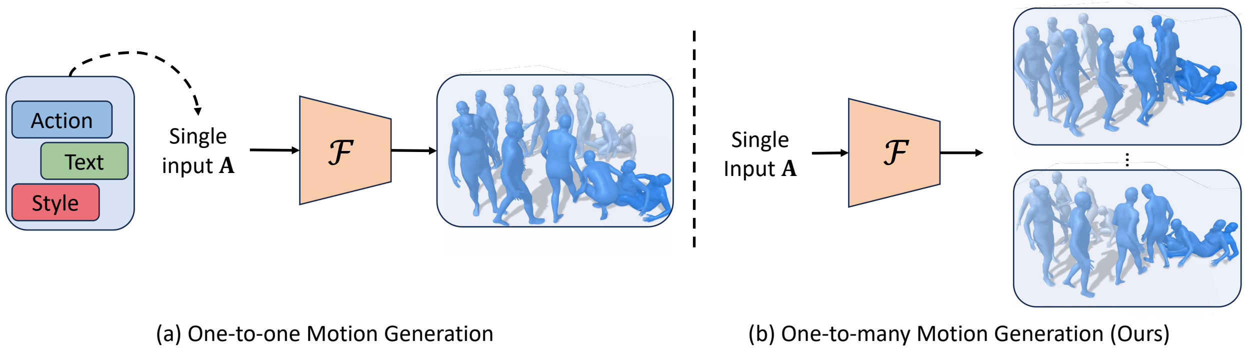


Figure 1.2: Deterministic vs non-deterministic motion generation. Our works emphasize both the quality and diversity of motion generation.

lead to mean motions that are lifeless and unnatural. With these challenges in mind, we collect two high-quality and large-scale multimodal human motion datasets in our first two projects: HumanAct12, consisting of 1,191 motions of 34 actions for action-based analysis; and HumanML3D, a dataset with 14,616 motions described by 44,970 textual descriptions for language-based analysis. To acknowledge the stochastic nature of human motion, we properly adapt generative models in our tasks, modeling the motion generation process in a non-deterministic manner (Fig. 1.2 (b)).

While the two aforementioned solutions empower users to generate plausible motions from scratch, they tend to produce neutral motions that lack expression of emotions, personalities, and other human factors. These factors however are essential for engaging character animation in the 3D industry. To address this, our third project introduces a versatile neural motion stylization framework. This framework enhances the stylistic characteristics of an existing motion by infusing cues from a user-provided reference style motion or style label. Akin to our previous two projects, our model adopts a non-deterministic approach, allowing for multiple ways in which a character can express the same style.

1.1.2 Human Character Animation

Following the motion generation procedures described in Sec. 1.1.1, we further consider a downstreaming application—creating real person animations using these generated motions. The conventional practices for human character animations rely on a high degree of expertise and manual intervention. It would be advantageous if even an average user could effortlessly create animations with their own customized 3D characters, with only the input of their own snapshots.

In pursuit of this objective, there are two prevalent approaches in the computer vision community. One approach is to directly generate pixel-level representations of 2D motions by learning the mapping between joint heatmaps and video pixels, as demonstrated in previous works [34, 33, 11, 44, 45]. However, learning such multimodal mappings is of high complexity. It often results in low-resolution and short videos where the human characters appear either blurry or surreal. The alternative approach involves modeling motion in 3D space while adhering to rigid skeletal constraints. For example, De et al. [46] generates human action videos by composing human motions and scenes using probabilistic graphical models in a 3D game engine, where 3D avatars are readily designed. The related works of [10, 9] extract 3D characters from single images and animate them using motions to produce 3D animations. Although these approaches enable the generation of higher-resolution videos, the characters often have distorted structures, and the textures on unseen views may appear unnatural. Additionally, these approaches still require human intervention in the middle stage, such as manual motion re-targeting.

These observations motivate our automated character animation pipeline (motion2video) presented in Chapter 3, which improves the 3D shape and texture reconstruction upon previous state-of-the-art methods and employs the parametric SMPL [47] model as a pivot for character rigging and animation. Moreover, we introduce a comprehensive action2video framework, which seamlessly integrates our motion2video pipeline with the action2motion module. This constitutes a fully integrated, user-friendly workflow, enabling users to generate a variety of realistic human videos by simply providing an action class and a single image.

1.1.3 Reciprocal Generation of 3D Motions and Texts

Motion generation and understanding, despite appearing as opposite tasks, share a deep interdependence. For instance, understanding the content of generated motions can inform improvements to the motion generator, leading to more faithful results. Meanwhile, motion understanding itself carries many applications including abnormal action detection, multimodal motion and video retrieval, and disability assistance. Thus, in our last chapter, we explore the intimate relationship between human motions and human language (i.e, text), encompassing both motion-to-text understanding and text-to-motion generation. While this connection is intuitive for humans, computers face challenges in comprehending visual concepts and associating them with natural language. Prior efforts [38, 39, 48] have employed advanced techniques, like GAN [49] and shared autoencoder, to address this problem. However, these approaches often yield short, incomplete, and detail-lacking language descriptions due to their direct mapping from pose sequences to language, two fundamentally distinct modalities. In our work, we propose a novel representation of motion sequence, namely motion token—a discrete and compact motion representation. This establishes a level playing field for both motion and text signals, enabling us to model the mapping between motion and text tokens akin to translating between two languages. Furthermore, we demonstrate that understanding the motion content can enhance the performance of its inverse counterpart, text-to-motion generation. We refer to the overall framework as the reciprocal generation of 3D human motions and texts.

1.2 Summary of Contributions

In summary, this thesis addresses several practical challenges in human action analysis. The primary contributions are several-fold.

- **New resources.** We contribute two large scale motion resources, HumanAct12 and HumanML3D, which have become important resources for multimodal human action analysis. HumanAct12 contains 1,191 motion clips and 90,099 frames, annotated with 12 coarse-grained and 34 fine-grained action classes. HumanML3D includes 44,970 textual descriptions and 14,616 motions, totaling 28.59 hours of data.

- **New task formulations.** We introduce the new formulations of existing motion generation tasks, including action2motion, action2video, text2motion, and motion stylization, and achieve state-of-the-art performance on each of these tasks. Unlike existing works that often produce deterministic results, our approaches emphasize both diversity and naturalness.

- **New techniques.** To address these problems, we properly adapt generative models as well as other task-specific technical designs. Specifically, we introduce novel techniques such as Lie algebraic pose representation and temporal variational autoencoder in Chapter 3 for action-to-motion generation, two-stage prediction of motion lengths and content in Chapter 4 for text-to-motion synthesis, and probabilistic style space for versatile motion stylization (Chapter 5). We further tackle the challenge of character animation by incorporating articulated motions, 3D character shapes and texture extraction from a single image (Chapter 3). The entire animation process is automated through our SMPL-based avatar rigging and deformation procedures. In Chapter 6, we present a framework that models the reciprocal relationship between human motion and human language. This framework introduces a discrete and compact motion representation, motion token, enabling seamless translation between motion signals and text signals on a level playing field. Through inverse alignment, our pretrained motion-to-text mapping improves the fidelity of text-to-motion synthesis.

- **New evaluation metrics.** We design a comprehensive set of evaluation metrics for various motion generation tasks. The metrics commonly respect the stochasticity nature of motion generation, statistically evaluate motions from several aspects: naturalness (e.g., FID), faithfulness (e.g., RPrecision, accuracy, multimodal distance) and diversity (e.g., diversity, multimodality).

1.3 Outline of Thesis

The organization and contributions of this thesis are as follows:

In Chapter 2, we provide a comprehensive overview of the knowledge and essential concepts that form the background for this thesis. This chapter begin with reviewing prior research in related fields, several deep generative models and their relevance to this thesis, as well as the important motion datasets. We also elaborate on the mathematical foundations of VAEs, which are important components in the majority of of our research.

In Chapter 3, we focus on the challenge of generating diverse and natural human motions and videos based on given action categories. Our technical approach involves a Lie algebraic pose representation to effectively encode the topology and articulation of these motions. Additionally, we introduce a novel temporal VAE framework that is capable of generating human pose sequences autoregressively. This is followed by our motion2video module, which automatically animates a 3D character based on a single-view image of a clothed human character. This pipeline allows users to generate realistic and customized animations by inputting a 2D person image and specifying an action class.

In Chapter 4, we delve into the complex problem of generating 3D human motions based on textual descriptions. This task poses significant challenges due to the inherent uncertainty in motion lengths and contents associated with textual input. To tackle this challenge, we introduce a two-stage framework: text2length sampling and text2motion generation. In the text2length stage, we sample from a learned distribution function of motion lengths conditioned on the input text. Subsequently, in the text2motion stage, we employ a temporal variational autoencoder (VAE) to synthesize a diverse set of human motions with the texts and sampled lengths.

In Chapter 5, our focus shifts to the task of 3D human motion stylization. We aim to update the style of an input motion while preserving its original content. Unlike previous approaches that directly manipulate motions in the pose space, we take advantage of the latent space learned by pretrained autoencoders, which offers a more expressive and robust representation for extracting and infusing motion styles. Building upon this, we introduce a novel generative model capable of producing diverse stylization results for a single motion (latent) code.

In Chapter 6, we reconsider the intricate relationship between 3D human motions and natural language, focusing on both motion understanding (motion2text) and motion generation (text2motion). We introduce a novel representation, motion token, which provides a discrete and compact representation of motion sequences. Leveraging this representation, we adapt neural machine translation (NMT) models to facilitate the mapping between the two modalities of motions and texts. Additionally, we integrate our motion2text module into the inverse alignment process of our text2motion training pipeline.

In Chapter 7, we summarize the works accomplished in this thesis and discuss future directions.

Chapter 2

Background

The purpose of this chapter is to provide essential background for this thesis. It involves several key elements, commencing with a review of prior efforts in related fields in Section 2.1. We subsequently discuss the applicability of various deep generative models to our research problem in Section 2.2, and introduce important motion datasets (Sec. 2.3) in these tasks. These foundational concepts are essential for understanding the core principles of our research.

2.1 Related Research Topics

Our discussion of research topics is not limited to 3D motions alone. Instead, we incorporate a broader perspective that encompasses motions, images, and videos. We delve into various aspects of generation, understanding, and stylization within this visual modality.

2.1.1 Human Motion Generation

There are several prior efforts in synthesizing 2D or 3D human motions based on audio, action categories, or texts. In terms of audio signal input, as audio is temporally aligned with its motion output, a common strategy is to employ a temporal sliding-window in translating the acoustic feature representation (e.g. MFCC) to individual human poses using recurrent neural networks

(RNNs). In [50], a Bi-Directional LSTM network is adopted to generate upper body gestures from the speech input. Similar LSTM-type models are also examined by [51] to predict upper body dynamics from piano and violin recital audios, and in [52] to capture the music-to-dance mapping. Recent works have started to address the stochastic nature of human dynamics grounded on audio signals. [53] employs a hybrid model of VAE and GAN to produce *non-deterministic* human dancing movements from music. The work of [54] further supports long-term music-to-dance generation with curriculum training. To synthesize human motions from scratch, [55] and [56] make use of Bayesian inference; the work of [57] instead considers a combined strategy of graph convolutional networks and GANs. The recent work of [58] synthesizes novel motions by free combinations of style and content codes extracted from the existing MoCap library.

At the time of our works, motion generation based on action categories or texts was still an emerging topic. For action-based motion generation, the only work of [33] adopts a two-stage GAN framework to generate 2D human motion progressively. Meanwhile, to translate text description to human motions, prior efforts such as [38, 37, 39, 59] resort to classical encoder-decoder RNN models, while it is proposed in [12] to learn a joint embedding space between natural language and 3D human dynamics. [59] considers the hierarchical pose structure as well as utilizing a pose discriminator. These methods however bear undesirable issues, as being deterministic one-to-one processes with fixed motion length.

For a more comprehensive overview of related works, we would like to introduce the works that have emerged concurrent with or after the publication of our research. TEMOS [60] takes advantage of Transformer VAE to optimize a joint variational space between natural language and motions, which is extended by TEACH [61] for long motion composition. MotionCLIP [62] and ohMG [63] model text-to-motion in an unsupervised manner using the large pretrained CLIP [64] model. The emerging autoregressive models and diffusion models have revolutionized the field of motion generation. Regarding autoregressive models, motions are firstly discretized as tokens via vector quantization [65]. Then, discrete motion tokens can be modeled by the expressive transformers as in the language model. This has become a popular trend in topics of action2motion [66], text2motion [5, 67, 68, 69], and dance generation [70, 71]. With diffusion models, a network is learned to gradually denoise the motion sequence, supervised by a scheduled diffusion process [72, 73, 74, 75, 76, 77, 78]. It shows promising ability in high-quality motion generation, whose inference efficiency however, is bottlenecked as it usually requires hundreds of sampling steps. Though several remedies are proposed, for example, latent diffusion [76, 79] and DDIM sampler [80], this shortfall remains unresolved.

2.1.2 Human Video Analysis

Action Video Generation. The task of generating human action videos has drawn research attention very recently. In the work of [33], 2D human motions are generated from known actions, they are then synthesized into 2D videos frame-by-frame with U-Net [81] and a dedicated image discriminator. In [34], based on an initial 2D pose extracted from a given image, a deterministic sequence of future 2D poses is produced for a given action category; this pose sequence is subsequently used to guide video generation via adversarial training. A similar method is considered in [11], where future 2D poses are instead generated stochastically with variational a auto-encoder. These efforts focus on tiny pixel-wise video generation, and human poses are manipulated in 2D image space. A recent work [46] proposes to generate 3D human videos directly from a 3D game engine using scene composition rules and procedural animation techniques. Our work differs from this work in two folds: 1)De et al. [46] generates 3D motions by extracting atomic motions from existing motion capture (MoCap) datasets, then stitches these atomic motions into action sequences through predefined rules. For example, a *walking* animation involves repetitions of swinging a left leg, then swinging a right leg, as well as corresponding pendular arm movements. However, this process is fairly labor-intensive. In our work, diverse 3D actions are automatically produced from a learned generative model end-to-end; 2) [46] animate artist-designed 3D avatars (rigid and clothed), while our method generates videos by rigging and animating characters with their 3D shapes and textures extracted from single 2D images.

Motion Transfer and Rigid Body Animation. Motion transfer is a traditional topic, aiming to transfer human motions from a source object to a target. Recent deep learning-based efforts typically consider 2D pixel-wise approaches, where mappings from source and target are based on local pixels or 2D patches. [82] and [44], for example, directly learn to map between human poses and appearances of one specific source subject. The aim of [83, 45, 84, 8] is to work toward a more general problem of driving an arbitrary target image with a source 2D pose sequence or videos. This is often realized by establishing connections between the source pose sequence and the target textured shape extracted from a given image, followed by warping the reference image to form the target video frame-by-frame. Although assembling promising results, the mainstream pixel-wise approaches nonetheless possess several limitations, including their innate difficulties in dealing with changing views or lifting to 3D motion spaces, as well as the level of complications in producing high-resolution and sharp images. The works of [85, 86] also consider a similar task, where motions from the source 3D character are re-targeted to 3D characters with different skeletons (e.g. joint number, bone lengths). Meanwhile, the 3D shapes of these target characters have been artistically designed and well-rigged ahead of time.

Meanwhile, there has also been a continuous line of research on rigid body animation of 2D/3D human characters that is especially empowered by advances in computer graphics techniques. Early work such as [87] uses a simple pose-retrieval framework, where a segmented garment database indexed by 2D skeleton poses is built for online searching during human image animation. Rigged human models are exploited in later endeavors for articulated object animation. In [88], characters extracted from 2D pictures are driven as rigidly as possible by external 3D MoCap sequences. In the intermediate steps, a 2D mesh with the 2D skeleton is constructed for the shape extracted from the input image. [10] further lifts this animation process into 3D space. Specifically, a semi-naked SMPL template is drawn out of 2D images and deformed to a rigged 3D mesh model with the boundary that closely matches the human silhouette in the input image. The recent work of [9] learns to directly predict a 3D animatable clothed human shape from a single image.

2.1.3 Visual Content Stylization

Image Style Transfer. Image style in computer vision and graphics is typically formulated as the global statistic features of images. Early work [89] finds it possible to transfer the visual style from one image to another by aligning their Gram matrices in neural networks. On top of this, [90, 91] enable faster transferring through additional feed-forward neural networks. The work of [92] realizes that the instance normalization (IN) layer could lead to better performance. However, these works can only be applied to single-style images. The following efforts [93, 94, 95] work on one-model multi-style transfer which maintains a candidate pool of style parameters for a pre-defined set of image styles. [96] facilitates arbitrary image style transfer by introducing adaptive instance normalization (AdaIN). Specifically, AdaIN modifies the mean and variance of the deep feature maps of content images using learned affine parameters from the style image. The effectiveness of IN and AdaIN editing style information has also been validated in several works such as CycleGAN [97] and styleGAN [20]. The aforementioned parametric image style transfer methods usually fail in detailed structure synthesis. Alternatively, in PatchGAN [98] and CycleGAN [97], textures and styles are translated between images by ensuring local similarity using a patch discriminator. A similar idea was adopted in [99], which proposes a patch co-occurrence discriminator that hypothesizes images with similar marginal and joint feature statistics appear perceptually similar.

Motion Style Transfer. Motion style transfer has been a long-standing challenge in computer animation. To manipulate styles, early machine learning approaches heavily depend on delicate-designed features in the temporal domain [100] or the spectrum [101]. The rise of data-driven methods that spontaneously learn from examples gets the elusive motion style free from precise mathematical definitions. The linear time-invariant (LTI) model is proposed based on per-frame pairwise between motions with similar content but heterogeneous styles [102]. To alleviate the demand for densely paired content, For example, the mixtures of autoregressive (MAR) models following a KNN search are designed to organize motion data more efficiently for online style transferXia et al. [7]. The early work of Xia et al. [7] design an online style transfer system based on KNN search. [103, 104, 105] transfers the style from reference to source motion through optimizing style statistic features, such as the Gram matrix, which are computationally intensive. Feed-forward based approaches [106, 6, 107] properly address this problem, where Aberman et al. [6] finalizes a two-branch pipeline based on deterministic autoencoders and AdaIN [96] for style-content disentanglement and composition; while [107] manages to stylize existing motions using one-hot style label, and models it as a class conditioned generation process. More recently, with the explosion of deep learning techniques, some works adopt graph neural networks (GNN) [108, 109], advanced time-series model [110, 111], or diffusion model [112] to the motion style transfer task. Specifically, Jang et al. [109] realize a framework that extracts style features from motion body parts. Meanwhile, the method from [112] uses diffusion model to learn the internal motifs from a single motion sequence.

2.1.4 Vision to Language

Image/Video Captioning. Vision-grounded text generation has a long history with extended literature. Here we only focus on the closely related topic of image and video captioning. Early efforts such as [113, 114] often resort to predefined sentence templates containing hand-crafted linguistic rules involving restricted categories of action and object. This has been fundamentally changed in the deep learning era, where we witness significant performance boosts by adopting a variety of powerful techniques including RNNs [115, 116], transformer [117], attentive context modeling [118], memory network [119, 120], GANs [121], and reinforcement learning [122, 123]. Take [25] for example, it starts by extracting high-level image features from pre-trained GoogleNet, which are then fed into a LSTM decoder to produce captions. In [115], an RNN-based video captioning model is considered, that extracts individual frame features from pre-trained CNN, and translates them to sentences through sequence-to-sequence learning. Further extensions are made through e.g. incorporating attention mechanism for better vision-language alignment [24, 118]. The following works further explore this framework via various directions, such as hierarchical encoder-decoder [124], replacing LSTMs with Transformer [117], feature fusion [125] and attentive context modeling [118]. A number of approaches also make efforts on other possible solutions.

For example, [119] deceives a memory module to model the textual contexts appearing in multiple related videos. [121] adopts adversarial inference by designing a discriminator to generate multi-sentence descriptions for long videos. [122, 123] utilize reinforcement learning techniques to directly optimize language metrics, like CIDEr [126].

Motion captioning. Research efforts on captioning 3D human motions are considerably more limited. [127] learns the mapping from human motions to language relying on two statistical models: one associates motions with words; the other assembles words back to form sentences. Recurrent networks are utilized by [38, 39] to address this task. In [38], motion and text features are extracted by two autoencoders respectively; this is followed by generating texts and motions from each other through shared latent vectors. Sequence-to-sequence RNNs are adopted in [39] to translate motions to scripts. Recently, the work of [48] proposes SeqGAN that extends the NMT model with a discriminator. Some common issues with existing motion2text results are typically short in length, often incomplete in content, and sometimes lacking in details.

2.1.5 Skeleton-based Representation

Numerous human pose representations have been considered over the years. The most-often used option is the joint-coordinate representation [128, 129] that directly characterizes the human pose by an ordered sequence of 2D/3D joint coordinates. It has a few variants: [130] consider incorporating the pair-wise relative positions of neighboring joints; meanwhile, only those informative joints are utilized in [131]. The part-based method is another line of pose representation. Specifically, a human pose is modeled as an ordered list of body parts. For example, in [132], the human body is divided into five main parts (i.e. torso and four limbs); pose sequences are then formulated by the displacement and rotations of body parts over time. Alternatively, the work of [133] models temporal information using dynamic time warping. Finally, Lie group or axis angle-based representation [134, 135, 136, 137, 138, 139] characterizes the skeleton as a kinematic tree, with its articulations realized by forward kinematics.

2.2 Related Methods

In this section, we introduce several deep generative models with a particular focus on their applications in sequence modeling, including generative adversarial networks (GANs), variational autoencoders (VAEs), diffusion models, and generative pre-trained Transformers (GPTs).

2.2.1 Generative Adversarial Networks

Generative Adversarial Networks are designed to learn an explicit mapping from noise to real data domain [49]. The fundamental idea behind a Generative adversarial network (GAN) [49] is to have two neural networks, known as the generator and discriminator, that are trained in a competitive manner. The generator network takes random noise as input and aims to generate data, such as images, that can deceive the discriminator. In contrast, the discriminator is trained to distinguish between generated data and real data. With the development of dedicated neural network architecture such as DCGAN [140], StyleGAN [20], and PatchGAN [98], GANs have found extensive use in high-quality static content generation, including images [99, 93, 22, 23] and 3D models [141].

On the other hand, scaling GAN to dynamic content generation is not easy. This is evidenced in several works of GAN-based audio generation [142, 143, 144], video generation [145, 21, 146] and early work of motion generation [34, 33, 147, 36]. The generated results in these cases are often short in length and exhibit blurriness. There are two-fold challenges to use GANs for sequence modeling. Firstly, GANs are known for their training instability, where one component, either the discriminator or generator, can dominate the learning process, causing convergence issues. Although various techniques (e.g. WGAN [148], non-staturate loss) have been proposed, this problem remains unresolved, especially in settings with sequential data. Secondly, sequence data have strong temporal dependencies. Capturing long-term dependencies and maintaining the coherence of motion over time is beyond the capability of traditional GAN architectures.

When it comes to human action generation, GAN-based approaches become even more complex.

GANs typically require large volumes of training data to generate high-quality content, while human motion and video datasets are usually limited in scale. Furthermore, GANs optimize by aligning data distributions through the discriminator, often failing to capture the intricate skeleton articulations. Minor deviations in skeleton features, such as rotations, can significantly affect the visual quality of motions. For example, motions generated by GANs often exhibit issues like sliding feet on the floor or feet penetrating through the ground.

2.2.2 Variational Autoencoders

Since Variational auto-encoder(VAE) [149] has been used in many places of this thesis, we would like to first elaborate on the theoretical foundation of VAEs, and then introduce their applications in related topics.

Mathematical Foundations of VAEs. VAE framework consists of an encoder and a decoder, which are normally two separate neural networks. Its goal is to learn a θ-parameterized generative model, $p_\theta(\mathbf{x}, \mathbf{z})$, over data $\mathbf{x}$ and latent variables $\mathbf{z}$. Technically, the learning objective is to maximize the likelihood function of $\mathbf{x}$, which could be further formulated as a marginal likelihood with regard to the latent variable $\mathbf{z}$, $p_\theta(x) = \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})$. Following the variational principle, a ϕ-parameterized neural network(i.e. encoder), $q_\phi(\mathbf{z}|\mathbf{x})$, is engaged to approximate the unknown posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$. We thus obtain the the following evidence lower bound (ELBO) to our data likelihood function:

$$\begin{aligned} \log p_\theta(\mathbf{x}) &= \log \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z}) \\ &\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})). \end{aligned} \tag{2.1}$$

The first ELBO term encourages the generated samples to be sufficiently close to the real samples; the second term penalizes KL-divergence between the prior and the approximated posterior distribution. Subsequently, the original objective of maximizing the data likelihood over data $\mathbf{x}$ becomes that of maximizing over the θ- and ϕ-parameterized ELBO function. In [150], a follow-up *conditional* variational auto-encoder (CVAE) framework is conceived by introducing a conditional variable, $\mathbf{y}$, as

$$\begin{aligned}\log p_\theta(\mathbf{x}|\mathbf{y}) &= \log \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z},\mathbf{y}) p(\mathbf{z}|\mathbf{y}) \\ &\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})} \log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{y}) - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) \parallel p(\mathbf{z})).\end{aligned} \tag{2.2}$$

VAEs have been widely used as a powerful learning technique in addressing various learning scenarios, including conditional generation [150], semi-supervised learning [151, 152], controllable generation [153], few-shot learning [154], disentangle representation learning [155, 156, 157] and VAE-GAN architecture [158].

To work with sequential data, VAEs are typically plugged into a recurrent network model, e.g. GRU and LSTM. Variational RNN [159], a pioneer work, uses vanilla RNN to model temporal dependencies in intermediate time-frames. The RNN output of the previous frame is used in generating posterior and prior distributions, as well as the follow-up decoding process. Variational RNN has been particularly favored in speech generation and handwriting character generation. [160] and [161] investigate the LSTM-based VAE for NLP modeling based on a sequence-to-sequence architecture, where the sequence encoder predicts a posterior distribution, from which the sequence decoder samples a latent vector and reconstructs the sequence. More specifically, temporal VAE models have been considered in motion and video generation. [162] consider generating videos from the textual caption, which is incorporated as semantic attentive vectors and fed to their temporal VAE. In VideoVAE [163], on the other hand, a structured latent unit is devised to model conditional factors including motion category and an initial frame to complete the rest frames. To predict future frames under uncertainty, [164] inspect the use of two separate RNNs to capture temporal dependencies of conditional posterior and prior spaces. A similar network structure is also scrutinized in [165], where it is extended to synthesize videos with pre-specified start and end frames. In terms of 3D motion prediction, given a start human pose, [166] complete the rest of 3D human motion with an LSTM-based VAE model. In [35], a similar model is engaged to learn the transition from observed sequence to future sequence for stochastic motion forecasting. A very recent work by [167] adopts VAE and a mix-and-perturbation strategy to statistically predict future motions.

2.2.3 Diffusion Models

Diffusion generative models have gained significant popularity in recent years, because of their capability to produce realistic and diverse samples while being relatively easy to train compared to other generative models like GANs. The core idea of the diffusion model is a diffusion process [168], which simulates the gradual transformation of the data distribution to a simple distribution (usually noise). Meanwhile, another denoising model is optimized to approximate the inverse transformation of the diffusion process. During generation, the denoising process starts with a simple noise sample and iteratively refines it to become more like the target data distribution. DDPM [169] introduces further insights into this framework, including a well-defined optimization objective, which has fostered the development of diffusion models. An extensive set of subsequent works have successfully applied diffusion models on image generation [170, 171, 79, 172, 173, 174], video generation [175, 176, 177, 178], and audio synthesis [179, 180]. One common concern of the diffusion model is its efficiency during generation. Theoretically, both the training and inference require hundreds of diffusion and denoising steps. The DDIM sampler [80] addresses this issue by introducing an implicit formulation of diffusion operations.

More recently, diffusion models have been scaled to 3D object and motion generation [181, 72, 74, 76, 75, 182]. To reduce the computation burden, LION [181] and MLD [183] firstly train an autoencoder to compress the 3D content into more compact latent features, and then learn diffusion models on the latent representation. MoFusion [182] proposes a versatile diffusion framework to integrate audio input and text inputs for human motion generation. It's worth noting that these diffusion-based models have emerged after the publication of our works.

2.2.4 Generative Pre-trained Transformer

Generative Pre-trained Transformer (GPT) [184] was originally designed for natural language processing. Unlike BERT [185], GPT uses unidirectional decoding order, learning to predict the next word in the sentence. By pretraining on vast language corpora, GPT exhibits remarkable capability in knowledge distillation, semantic understanding and generation. An example of this is ChatGPT, which contains 175 billion parameters and was trained on approximately 570GB of text data.

The generative power of GPT relies on discrete sampling from the token vocabulary. Transferring the insights of GPT to computer vision tasks presents a challenge due to the continuous nature of visual data, such as images, motion, or point clouds. Approaches like VQGAN [186] tackle this challenge by discretizing image pixels into discrete visual tokens through vector quantization in the initial stage. Subsequently, a GPT model learns to model the context of these visual tokens, enabling it to generate high-fidelity images. Cogview [187] and Cogvideo [188] employ similar strategies but operate on much larger image and video datasets, aiming to generate text-based visual content.

2.3 Related Datasets

An overview of human motion datasets is presented in Table 2.1.

Action Category and 3D Motion. Previous 3D Human Motion datasets are often constructed with the main purpose of developing action recognition tasks. Typically, these datasets involve the use of kinematic sensors or motion capture systems to capture the pose or 3D joint positions of human subjects. Notable datasets like CMU MoCap [189] and HDM05 [133] have more than 100,000 3D poses and 2,000 3D motion sequences that are associated with succinct textual descriptions. Unfortunately, the motions are mostly about locomotions. UTKinect-Action3D [40] and MSR-Action3D [41], on the other hand, have a much smaller tally of motion sequences. NTU-RGBD [190] is by far the largest human motion dataset, consisting of over 100,000 motions belonging to 120 classes. Nevertheless, the joint positions acquired from Microsoft Kinect-I cameras are notably inaccurate. These observations motivate us instead curate our in-house 3D human action dataset, HumanAct12, as well as revamping the pose annotations of NTU-RGBD.

Language and 3D Motion. There are a number of existing datasets of 3D motion captured human motions, such as CMU Mocap [189], Human3.6M [191], MoVi [192] and BABEL [193], in the form of everyday actions and sports movements. BABEL contains brief descriptions for single action motions, while the other datasets possess no text descriptions. KIT Motion-Language Dataset [43] is to date the only available dataset comprising both 3D human motions and their textual descriptions, which consists of 3,911 motion sequences & 6,278 sentences and is focused on locomotion movements. However, this scale of the dataset is insufficient for learning a capable motion generator. Therefore, in our Chap. 4, we collect a large-scale motion-language dataset with a broad range of actions, including dance, sports, and daily activities.

Style and 3D Motion. Aberman et al. [6] and Xia et al. [7] provide two commonly used motion-style datasets with high-quality motion data. Aberman et al. [6] collect 16 long-term motions concerning 16 styles, including old, zombie, femalemodel, and happy. In comparison, Xia et al. [7] is a relatively smaller motion dataset, where motions are normally around 3 seconds, and categorized into 8 styles.

Name	Venue	Subjects	Sequences	Length	Annotation	Remark
Human3.6M [191]	TPAMI 2014	11	-	5.0h	-	15 long actions.
CMU Mocap [189]	Online 2015	109	2605	9h	-	Brief descriptions.
HDM05 [133]	Online 2007	5	2337	5h	-	Brief descriptions.
AMASS [194]	ICCV 2019	344	11265	40h	-	-
UKinect-Action3D [40]	CVPRW 2012	10	200	-	10 action classes	-
MSR-Action3D [41]	CVPRW 2010	10	567	-	20 action classes	-
NTU-RGB-D [190]	TPAMI 2019	106	114.4K	74h	120 action classes	Noisy poses.
Movi [192]	PLoS One 2021	90	-	9h	20 action classes	-
HumanAct12 [1]	MM 2020	12	1191	6h	12 action classes	-
KIT-ML [43]	Big data 2016	111	3911	10.3h	6.3k text descriptions	Lack diversity.
BABEL [193]	CVPR 2021	344	4083	43.5h	32k short descriptions	Single-action sequences.
HumanML3D [3]	CVPR 2022	344	14.6k	28.5h	45k text descriptions	-
Aberman et al. [6]	SIGGRAPH 2020	1	32	2.3h	16 style classes	-
Xia et al. [7]	SIGGRAPH 2015	1	-	11m	8 style classes	-

Table 2.1: Datasets for human motion analysis.

Chapter 3

Action Category based Human Motion and Video Generation

As shown in Fig. 3.1, we aim to tackle the interesting yet challenging problem of generating videos of *diverse* and *natural* human motions from prescribed action categories. The key issue lies in the ability to synthesize multiple distinct motion sequences that are realistic in their visual appearances. It is achieved in this chapter by a two-step process that maintains internal 3D pose and shape representations, *action2motion* and *motion2video*. Action2motion stochastically generates plausible 3D pose sequences of a prescribed action category, which are processed and rendered by motion2video to form 2D videos. Specifically, the Lie algebraic theory is engaged in representing *natural* human motions following the physical law of human kinematics; a temporal variational auto-encoder (VAE) is developed that encourages *diversity* of output motions. Moreover, given an additional input image of a clothed human character, an entire pipeline is proposed to extract his/her 3D detailed shape, and to render in videos the plausible motions from different views. This is realized by improving existing methods to extract 3D human shapes and textures from single 2D images, rigging, animating, and rendering to form 2D videos of human motions. It also necessitates the creation and reannotation of 3D human motion datasets for training purpose. Thorough empirical experiments including ablation study, qualitative and quantitative evaluations manifest

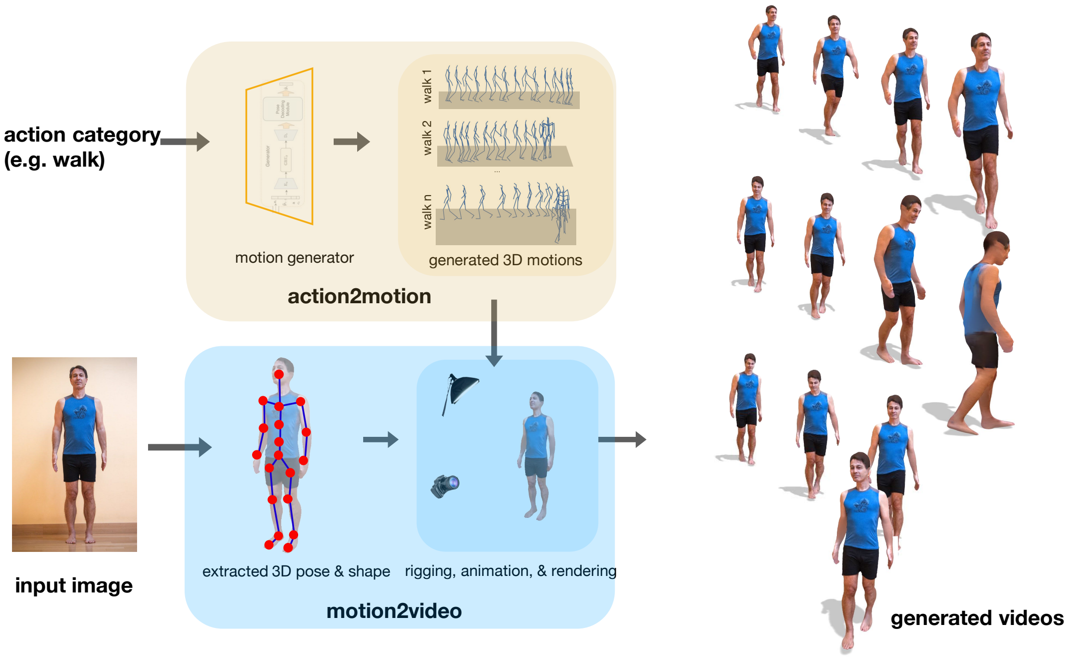


Figure 3.1: Our action2video pipeline generates human full-body motion videos of prescribed actions in two steps: *action2motion* first generates diverse and natural 3D motions of predefined actions; *motion2video* proceeds to extract 3D surface shape and texture from an additional 2D input image, and to render 2D videos of the generated motions.

the applicability of our approach, and demonstrate its competitiveness in addressing related tasks, where components of our approach are compared favorably to the state-of-the-arts. Action2motion work has been published as [1], and the full work has been published as [2]. The related code, data and pre-trained model are readily available: https://ericguo5513.github.io/action-to-motion/.

3.1 Introduction

Human-centric activities always play a key role in our daily life. In recent years, noticeable progresses have been made in video forecasting [195, 196] and synthesis [156, 21, 197, 164]. Meanwhile, it remains a substantial challenge in generating realistic videos of diverse and plausible human motions. This is evidenced in many recent video generation efforts [34, 33, 11], where the appearances of synthesized human characters are unfortunately either blurring or surreal, and are still far from being photo-realistic; their motions are often distorted and unnatural. These observations stress the importance of properly modeling human body postures & temporal articulations, as well as the surface shapes and textures of the local body parts. It also motivates us to examine the problem of generating videos of human motions based on action categories, the basic ingredient of human behaviors.

Due to the complexity of human articulations and pose dynamics, generating human videos is far from being trivial. Existing efforts usually represent human motions in 2D space, which are then rendered pixel-wise to form 2D videos. Moreover, extra information such as an initial 2D pose or a partial/entire motion sequence is usually required, which is practically undesirable. For instance, [34] produces deterministic sequence of 2D motions, which is followed by synthesizing the appearances frame-by-frame through adversarial training. Action-conditioned 2D human behavior modeling is also studied in [33], where 2D pose generator and motion generator are trained progressively. Very recently, the efforts of [10, 9] consider the related task of extracting 3D characters from single images, which is then animated to form 3D motions; [46] addresses another related task of generating human action videos by composing the human motions and scenes with probabilistic graphical models in 3D game engine. However, the motions used in both methods are real-life motions that have been made available in prior, instead of being synthesized on the spot.

Overall, the existing methods fall short in the following aspects: 1) direct modeling of 2D motions is inherently insufficient to capture the underlying 3D human pose articulations and shape deformations. The absence of 3D geometric information often leads to visual distortions and ambiguities; 2) coordinate locations of body joints are commonly used as the human pose representation, which undesirably entangle the human skeletons and their motion trajectories. Moreover, this creates extra barriers in modeling human kinematics; 3) initial poses often impede the diversity of generated human dynamics. For example, in actions such as *warm up* and *boxing*, initial poses crucially influence the formation of the rest sequences; and 4) the popular choice of pixel-to-pixel synthesis among existing efforts on action conditioned video generation has been evidenced incapable of generating detailed and high-resolution views. The aforementioned observations inspire us to consider a two-step pipeline: action2motion generates diverse & natural 3D human motions from prescribed action categories, and motion2video proceeds to extract human character out of an additional input image, to rig, animate, and render to form 2D videos, as illustrated in Fig. 1.1.

In action2motion, we aim at generating diverse motions to traverse the motion space, and to cover various styles of individuals performing the same type of actions; meanwhile, each motion is expected to be visually plausible. This leads to our temporal variational auto-encoder (VAE) approach using Lie algebra pose representation. Inspired by the work of [164] in generic video generation, here we leverage the posterior distribution learned from previous poses as a learned prior to gauge the generation of present pose; by tapping into the recurrent neural net (RNN) implementation, this learned prior also encapsulates temporal dependencies across consecutive poses. For pose representation, human pose could be characterized as a kinematic tree based on human body kinematics. There are multiple advantages of using Lie algebraic representation over the popular joint-coordinate representation: (i) Lie representation disentangles the skeleton anatomy, temporal dynamics, and scale information; (ii) it faithfully encodes the anatomical constraints of skeletons by following the forward kinematics [198]; (iii) the dimension of Lie algebraic space corresponds exactly to the degree of freedom (DoF), which is more compact compared to joint-coordinate representation. In practice, the adoption of Lie representation notably mitigates the change-of-length and trembling phenomenons prevailing in joint coordinates representations; it also facilitates the generation of natural, lifelike motions, and simplifies the training process. Furthermore, a global and local movement integration module is used to infer the global pose trajectory from temporal articulations of body parts. This promotes consistence between local shape deformations and global motion trajectory (i.e. direction and velocity), especially when synthesizing locomotion actions such as walking and jumping.

It is followed in our pipeline by motion2video, where a 3D character is extracted, rigged, animated according with stochastically generated motions, and rendered to form 2D videos. In fact, animating 3D characters remains an open problem. A common strategy is to extract their 3D shapes and textures from a single input image. Prior efforts such as [10] align the silhouette and texture of single image to a 3D human shape (e.g. SMPL [47]). Due to single input view, nonetheless, they fail to synthesize body textures of unseen views. Recent deep learning methods [199, 200, 201, 9, 202] shed lights on reliable recovery of 3D surfaces and textures from single images. Meanwhile their results suffer from either low-fidelity, with input image resolution limited to at most 512×512 [200, 9, 202], or ill-posed texturing on occluded areas and novel view [201]. A simple strategy is developed in our work, leading to improved texture mapping in these cases.

In summary, our main contributions are three-fold: first, a novel two-step pipeline of action2motion & motion2video is proposed to address the challenging problem of 3D human motion & video generation from action type and single image; second, a dedicated Lie Algebra based VAE framework is developed, capable of producing diverse life-like human motions from prescribed action categories; third, as part of our pipeline, an improved strategy is used in extracting 3D shapes and textures from single images, that is capable of synthesizing visually-appealing texture of unseen views. Moreover, an in-house 3D human motion dataset, HumanAct12, has been curated.

3.2 Our Approach

The pipeline of our approach, **action2video**, consists of two steps: step one (action2motion) synthesizes human pose sequences from a prescribed action category (Sec. 3.2.1); step two (motion2video) extracts a specific 3D human shape and texture from a reference image to render the generated motions into 2D videos (Sec. 3.2.2).

3.2.1 Step One: Action2Motion

Our action2motion framework comprises a temporal VAE (Sec. 3.2.1) with a Lie algebra based representation (Sec. 3.2.1). We also investigate four strategies to decode neural hidden unit to obtain global 3D positions of motions (Sec. 3.2.1 and Sec. 3.4).

Disentangled Representation with Lie Algebra

As shown in Fig. 3.2, a human pose could be characterized in the form of a kinematic tree that consists of five kinematic chains: main spine and four limbs. Meanwhile, this skeleton model is formed by N oriented edges (i.e. bones) $E = \{e_1, \ldots, e_N\}$ that interconnect $N + 1$ joints. By incorporating Lie algebraic apparatus, motion of 3D joints could be decomposed into three parts: skeleton anatomical information, motion trajectories, and bone lengths.

For each skeletal bone, e_n, a local coordinate is attached, with the bone itself being aligned with the x-axis and its starting joint being stuck to the coordinate origin. The relative 3D locations between two consecutive bones could be modeled as a series of 3D rigid transformations. Specifically, given two connected bones e_n and e_{n+1} along a kinematic chain, a joint $\mathbf{c} = (x, y, z)^\top$ in the local coordinate of e_n amounts to a transformed location $\mathbf{c}' = (x', y', z')^\top$ in the local coordinate of e_{n+1}, by exercising the following transformation

$$\begin{pmatrix} \mathbf{c}' \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R}_n & \mathbf{d}_n \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ 1 \end{pmatrix}. \tag{3.1}$$

where $\mathbf{R}_n \in \mathbb{R}^{3\times 3}$ is the rotation matrix, and $\mathbf{d}_n \in \mathbb{R}^3$ is the translation vector along x-axis. Furthermore, $\mathbf{d}_n$ could be written as $(b_n, 0, 0)^\top$ with b_n denoting the bone length of e_n, a constant number over time.

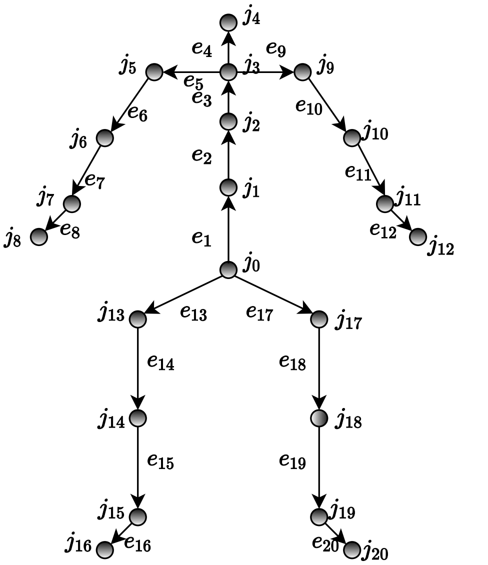


Figure 3.2: An example of human skeleton which consists of 21 joints and 20 body parts.

Here, the rotation matrix $\mathbf{R}_n \in \mathbb{R}^{3\times 3}$ between two local coordinates is an element of Special Orthogonal Group SO(3), which is a matrix Lie group. Hence, excluding bone lengths, the relative geometry between e_n and e_m is a point in SO(3) and the whole skeleton is represented as a point in SO(3) × SO(3) × ... × SO(3), which is a matrix Lie group endowed with a differentiable manifold structure [138, 136]. Similarly, the motion could be characterized as a curve in Manifold. To carry out optimization in Manifold, we could engage the proper mathematical apparatus of Lie algebra or tangent space that could be regarded as a flat space, thus our familiar linear algebra computations could be utilized.

Lie algebra $\mathfrak{so}(3)$ The 3 by 3 identity matrix I_3 is an element of a SO(3) and is referred to as the identity element in this group [198]. The tangent space at the identity element I_3 of SO(3) is known as the Lie algebra space, $\mathfrak{so}(3)$, a 3-dimensional vector space spanned by the elements of a 3×3 skew-symmetric matrix $\hat{W}$, as

$$\hat{W} = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix}. \tag{3.2}$$

The association between a $\mathbf{R} \in \mathrm{SO}(3)$ and its Lie algebra vector $\mathbf{w} \in \mathfrak{so}(3)$ could be given by the logarithm map $\log_{\mathrm{SO}(3)}$: $\mathrm{SO}(3) \rightarrow \mathfrak{so}(3)$, as

$$\mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \frac{\theta}{2\sin\theta} \begin{pmatrix} \mathbf{R}(3,2) - \mathbf{R}(2,3) \\ \mathbf{R}(1,3) - \mathbf{R}(3,1) \\ \mathbf{R}(2,1) - \mathbf{R}(1,2) \end{pmatrix}, \tag{3.3}$$

where $\theta = \arccos \frac{trace(R)-1}{2}$ [198]. Since $\mathbf{w}$ is not uniquely mapped, we use the value with norm in range$[-\pi, \pi]$. Similarly, the inverse transformation is given by the exponential map $\exp^{\mathfrak{so}(3)}$: $\mathfrak{so}(3) \rightarrow \mathrm{SO}(3)$. Please refer to [198] for more details.

For a 3D rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$, the associated Lie algebraic vector $\mathbf{w} \in \mathfrak{so}(3)$ is an axis-angle vector. For a human skeleton, the exact degree of freedom (DoF) of a axis-angle vector is determined by the rotation orientations of two successive bones, and is up to 3. For example, if two bones are oriented in the same or reverse direction, $\mathbf{w}$ is a zero vector with 0 DoF; if one bone only rotates along one axis, then the DoF reduces to 1.

Mapping Lie algebra parameters to 3D positions. Now we focus on an articulate object with K kinematic chains; assume the k-th chain have m_k joints, with each joint parameterized by a 3-dimensional $\mathfrak{so}(3)$ vector, $\mathbf{w}_i^k, i \in \{1, 2, \ldots, m_k\}$. A human pose is thus represented by composition of Lie algebra vectors of joints/bones on kinematics chains, $\mathbf{p}_{\mathrm{Lie}} = ({w_1^1}^\top, \ldots, {w_{m_1}^1}^\top, \ldots, {w_1^K}^\top, \ldots, {w_{m_K}^K}^\top)$. Now, the 3D position of a joint i in a chain k, $\mathbf{J}_i^k$, is obtained following a exponential map of the Lie algebraic values, also known as *forward kinematics*, as

$$\mathbf{J}_i^k = \left[\prod_{j=0}^{i-1} \exp(\hat{W}_j^k)\right] \mathbf{d}_i^k + \mathbf{J}_{i-1}^k. \tag{3.4}$$

Here $\mathbf{d}_i^k = (b_i^k, 0, 0)$, with b_i^k representing the bone length of e_i^k. In addition, forward kinematics typically starts from a root joint whose position $\mathbf{J}_0 \in \mathbb{R}^3$, and Lie algebraic values $\hat{W}_0$ stand for the global location and orientation of the entire human body. In our representation, the global location $\mathbf{J}_0$ is independent from the pose. Therefore, given a motion with T successive poses, the sequence $(\mathbf{J}_{0,1}, \ldots, \mathbf{J}_{0,T}) \in \mathbb{R}^{3\times T}$ makes up the body motion trajectory, with $\mathbf{J}_{0,t}$ denoting its global location at frame t.

Accordingly, the 3D coordinates vector of a body pose, formally denoted as $\mathbf{p} = ({\mathbf{J_1}^1}^\top, \ldots, {\mathbf{J_{m_1}}^1}^\top, \ldots, {\mathbf{J_1}^K}^\top, \ldots, {\mathbf{J_{m_K}}^K}^\top)$ could be obtained by the joint-wise forward kinematics of a composition of *bone lengths*, *root position*, and *Lie algebraic vector*. For simplicity, we denote this mapping as $\mathbf{\Gamma}(\mathbf{p}_{\text{Lie}}) : \mathbf{p}_{\text{Lie}} \rightarrow \mathbf{p}$. Overall, a human *motion* is represented by three parts:

- Lie algebra parameters $\mathbf{M}_{\text{Lie}} = \big(\mathbf{p}_{\text{Lie}}^1, \ldots, \mathbf{p}_{\text{Lie}}^T\big)$.
- Root trajectory $(\mathbf{J}_{0,1}, \ldots, \mathbf{J}_{0,T})$: root trajectory could be represented by either absolute root locations or relative translations between consecutive root locations. The latter works better in our setting.
- Bone lengths $(b_0, \ldots, b_N)$: due to the invariant nature of bone lengths of human skeleton, the skeleton bone lengths are acquired from typical real-life human bodies, and are fixed over time. This also reciprocally enables us to generate motions with controllable body scales by manipulating the bone lengths.

Conditioned Temporal VAE

Consider a real motion or pose sequence $\mathbf{M} = (\mathbf{p}_1, \ldots, \mathbf{p}_T)$. Our temporal VAE aims to maximize the likelihood of the pose sequence $\mathbf{M}$. At time t, a posterior network $q_\phi(\mathbf{z}_t|\mathbf{p}_{1:t})$ approximates

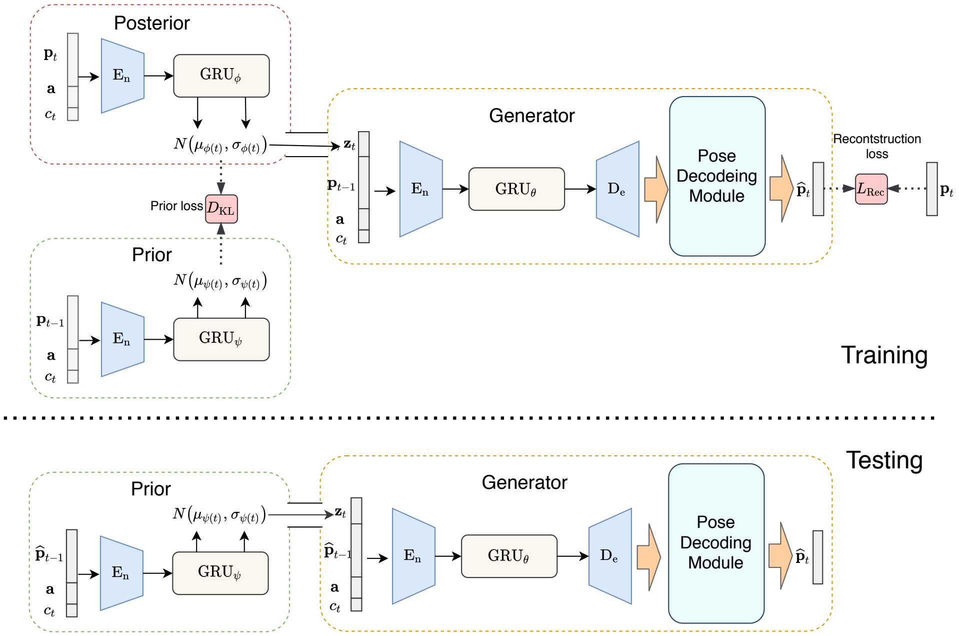


Figure 3.3: Visual diagram of action2motion. Top row shows the training phase: at time t, the posterior and prior networks take as input a concatenation of three parts - action category a, time counter c_t and immediate pose vector ($\mathbf{p}_t$ or $\mathbf{p}_{t-1}$). The generator receives an addition latent vector $\mathbf{z}_t$ that is sampled from the learned posterior distribution. Afterwards, the 3D joints of current pose is obtained from the decoder of generator through *pose decoding module*. Bottom row depicts the testing phase: a latent vector is alternatively sampled from the prior distribution, which triggers the aforementioned process in generating 3D pose sequences.

the true posterior distribution conditioned on $\mathbf{p}_{1:t-1}$. Then, with sampled latent variables $\mathbf{z}_{1:t}$ and previous states $\mathbf{p}_{1:t-1}$, our RNN generator $p_\theta(\mathbf{p}_t|\mathbf{p}_{1:t-1}, \mathbf{z}_{1:t})$ reconstructs the current pose $\mathbf{p}_t$. This leads to the following variation lower bound:

$$\begin{aligned}\log p_\theta(\mathbf{M}) \geq \sum_t \Big[&\mathbb{E}_{q_\phi(\mathbf{z}_t|\mathbf{p}_{1:t})} \log p_\theta(\mathbf{p}_t|\mathbf{p}_{1:t-1}, \mathbf{z}_{1:t}) \\ &- D_{\mathrm{KL}}\left(q_\phi(\mathbf{z}_t|\mathbf{p}_{1:t}) \parallel p(\mathbf{z}_t)\right)\Big].\end{aligned} \tag{3.5}$$

Note at time t, our RNN module takes as input the immediate past frame $\mathbf{p}_{t-1}$ and $\mathbf{z}_t$. The influence from previous time slices $\mathbf{p}_{1:t-2}$ and $\mathbf{z}_{1:t-1}$ lies in the ability of RNN module capturing long-term temporal dependencies.

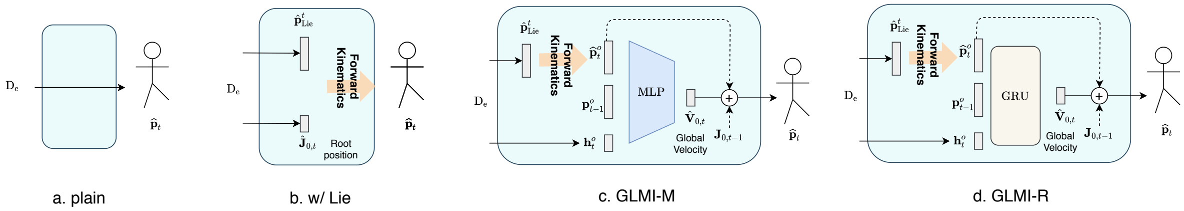


Figure 3.4: Four variants of the pose decoding module conceived in our work: (a) direct generation of 3D joint positions; (b) generation with Lie algebraic representation; and (c)-(d) *global and local movement integration* (**GLMI**)-based generation with Lie algebraic representation, implemented by multi-layer perceptron (GLMI-M) or GRU (GLMI-R).

In terms of the prior $p(\mathbf{z}_t)$, one option is to consider an identity Normal distribution, $\mathcal{N}(0, \mathbf{I})$. This is unsuitable though for the motion generation problem, as the pose variation varies over time. Take *running* motions as example, the temporal pose variances are typically relatively small, which however could become significantly larger when e.g. the runner makes a U-turn. Inspired by the observation that the variation of present pose is highly correlated to its past time-steps [164], we model its prior by a neural network that conditions on its previous steps $\mathbf{p}_{1:t-1}$, $p_\psi(\mathbf{z}_t|\mathbf{p}_{1:t-1})$. This leads to a re-formulation of the ELBO objective function

$$\begin{aligned}\log p_\theta(\mathbf{M}) \geq \sum_t \Bigg[& \mathbb{E}_{q_\phi(\mathbf{z}_t|\mathbf{p}_{1:t})} \log p_\theta(\mathbf{p}_t|\mathbf{p}_{1:t-1}, \mathbf{z}_{1:t}) \\ & - D_{\text{KL}}\left(q_\phi(\mathbf{z}_t|\mathbf{p}_{1:t}) \parallel p_\psi(\mathbf{z}_t|\mathbf{p}_{1:t-1})\right)\Bigg],\end{aligned} \tag{3.6}$$

where the distance penalty between prior and posterior distributions further encourages temporal consistency.

Architecture of Action2Motion

Our action2motion step consists of three main components: posterior network, prior network, and generator, which are shown in Fig. 3.3. The input vector contains the following parts: the pose vector $\mathbf{p}_t$ or $\mathbf{p}_{t-1}$, an one-hot vector $\mathbf{a}$ to encode action category, and $c_t \in [0, 1]$, a time-counter to keep record of where we are in the sequence generation progress. As depicted in Fig. 3.3, during training, a noise vector is sampled from the posterior distribution $q_\phi(\mathbf{z}_t|\cdot)$, and fed into the generator, which then produces the final 3D pose prediction by running through the pipeline of encoder $\mathrm{E_n}$, GRU unit GRU_θ, decoder $\mathrm{D_n}$, and pose decoding module. In testing, as the real data $\mathbf{p}_t$ is not available, $\mathbf{z}_t$ is instead sampled from the learned prior distribution, $p_\psi(\mathbf{z}_t|\cdot)$.

Specifically, our encoder $\mathrm{E_n}$ and decoder $\mathrm{D_n}$ are composed of linear fully connected layers with different weights, and updated with the whole network. Moreover, our posterior network (q_ϕ) and prior network (p_ψ) utilize the same architecture, but with different parameters. They are respectively described as:

$$\begin{aligned} \mathbf{h}_t &= \mathrm{E_n}(\mathbf{p}_t, \mathbf{a}, c_t), \;\; c_t = \frac{t}{T} \\ (\mu_\phi(t), \sigma_\phi(t)) &= \mathrm{GRU}_\phi(\mathbf{h}_t) \end{aligned} \tag{3.7}$$

and

$$\begin{aligned} \mathbf{h}'_{t-1} &= \mathrm{E_n}(\mathbf{p}_{t-1}, \mathbf{a}, c_t), \;\; c_t = \frac{t}{T} \\ (\mu_\psi(t), \sigma_\psi(t)) &= \mathrm{GRU}_\psi(\mathbf{h}'_{t-1}). \end{aligned} \tag{3.8}$$

Further investigation of the pose decoding module is provided in the following section.

Pose Decoding

Fig. 3.4 illustrates the four pose decoding variants investigated in our work. The most straightforward and commonly-used approach is Fig. 3.4(a), where the 3D joint locations are directly and simultaneously regressed from the decoder. It however contains redundant parameters, and does not follow the kinematics law that dictates the 3D articulations of the body skeleton. Alternatively, the Fig. 3.4(b) variant incorporates Lie algebraic representation. The decoder here contains two vectors, skeletal Lie algebraic values $\hat{\mathbf{p}}^t_{\mathrm{Lie}}$, and global root position $\hat{\mathbf{J}}_{0,t}$. The final 3D joints are produced by *forward kinematics* (see Sec. 3.2.1). Though working well for many motion scenarios, it encounters issues when local body movements and global motions are highly correlated. Take action *walk* for example, the instantaneous velocity of walking is significantly affected by the movement of *legs*; independently generating global and local body motions is observed to lead to e.g. sliding-feet phenomenon, as depicted in Fig. 3.11.

Global and local movement integration. Existing efforts in motion forecasting or generation usually predict *only* relative body joint positions, this is, relative to the root joint, at the cost of neglecting the global motion all together [147, 57, 138, 137]. In other words, the root joint of human full-body is fixed to coordinate origin during the entire motion sequence. Recently, [203] consider global motion by directly enforcing MSE loss between predicted and ground-truth root joint locations, which is similar to the Fig. 3.4(a) variant.

Intuitively, the transition between two consecutive poses, measured by the displacement of the root joint in the two frames, is highly correlated to the body gesture of these two poses. Consider a person who is walking on a flat ground, his walking pace depends upon how wide his legs span. This inspires us to propose a *global and local movement integration unit* (**GLMI**) which, rather than predicting global transition and local joints concurrently, will first generate relative poses, then infer global motion from consecutive local poses, as illustrated in Fig. 3.4(c). Here $\hat{\mathbf{p}}^t_{\text{Lie}}$ is the Lie parameter vector produced by the generator, which is then transformed to 3D joint locations $\hat{\mathbf{p}}^o_t$ through forward kinematics; $\mathbf{p}^o_{t-1}$ is the offset value of 3D coordinates of previous pose; $\mathbf{h}^o_t$ is a hidden vector containing upstream information. The three vectors are fed into a fully connected layer, MLP, which then produces the velocity (i.e. relative translation) $\hat{\mathbf{V}}_{0,t}$ at time t. Finally, the 3D global position $\hat{\mathbf{p}}_t$ could be obtained by summation of the three components: root position of previous pose $\mathbf{J}_{0,t-1}$, estimated velocity $\hat{\mathbf{V}}_{0,t}$, and the current local pose $\hat{\mathbf{p}}^o_t$. Mathematically, this process is expressed as

$$\begin{aligned}
\left(\hat{\mathbf{p}}^t_{\text{Lie}}, \mathbf{h}^o_t\right) &= \mathrm{D_e}(\mathbf{h}^\theta_t) \\
\hat{\mathbf{p}}^o_t &= \mathbf{\Gamma}(\hat{\mathbf{p}}^t_{\text{Lie}}) \\
\hat{\mathbf{V}}_{0,t} &= \mathrm{MLP}(\hat{\mathbf{p}}^o_t, \mathbf{p}^o_{t-1}, \mathbf{h}^o_t) \\
\hat{\mathbf{p}}_t &= \hat{\mathbf{p}}^o_t + \mathbf{J}_{0,t-1} + \hat{\mathbf{V}}_{0,t}.
\end{aligned} \tag{3.9}$$

To further capture the temporal dependency of a global trajectory, another version of GLMI is also proposed, with the backbone of MLP replaced by recurrent units, GRU, as presented in Fig. 3.4(d). Besides, a trajectory alignment loss between the predicted velocities $\hat{\mathbf{V}}_{0,t}$ and real velocities $\mathbf{V}_{0,t}$ is also introduced, to encourage accurate velocity estimation. Among these variants, the GLMI-M variant is found to produce the overall best results, and is utilized in our approach by default.

Final Objective

To summarize, our final objective function becomes

$$\begin{aligned}\mathcal{L}_{\theta,\phi,\psi} = -\sum_{t=1}^{T}\Bigg[&\mathbb{E}_{q_\phi(\mathbf{z}_t|\mathbf{p}_{1:t},\mathbf{a},c_t)} \log p_\theta(\mathbf{p}_t|\mathbf{p}_{1:t-1},\mathbf{z}_{1:t},\mathbf{a},c_t) \\ &- \lambda_{kl} D_{\mathrm{KL}}\left(q_\phi\left(\mathbf{z}_t|\mathbf{p}_{1:t},\mathbf{a},c_t\right) \parallel p_\psi\left(\mathbf{z}_t|\mathbf{z}_{1:t-1},\mathbf{a},c_t\right)\right) \\ &- \lambda_{align}\|\mathbf{V}_{0,t} - \hat{\mathbf{V}}_{0,t}\|_2\Bigg],\end{aligned} \tag{3.10}$$

where λ_{kl} and λ_{align} are two tuning parameters to trade-off among reconstruction error $\mathcal{L}_{rec}$, KL-divergence, and trajectory alignment loss. Empirically, a larger λ_{kl} is observed to enhance the quality of generated motions but may decrease their diversity; and vice versa for a smaller λ_{kl}.

For the reconstruction error (the first term in Eq. (3.10)), the per-joint loss suggested in [204] is considered, as

$$\mathcal{L}_{rec}(\mathbf{p}_t, \hat{\mathbf{p}}_t) = \sum_{k=1}^{N+1} \|\mathbf{J}_{k,t} - \hat{\mathbf{J}}_{k,t}\|_2. \tag{3.11}$$

Here $N+1$ denotes the number of skeletal joints.

In our work, the trajectory alignment loss is only used in the methods of Fig. 3.4(c) and (d), where the models are trained with the re-parameterization trick of [149].

Training Strategy

One common issue in sequence modeling is the discrepancy of information exposure during training vs. testing phases. For example, in a RNN model, a *ground-truth* pose is taken as input to generate next pose in training; while in testing phase, a *generated* pose is used instead to produce next pose. To mitigate the issue, a mixed training strategy is adopted here, that chooses whether to use (or not to use) *teacher forcing* [205] by randomly draws from a Bernoulli distribution, $V \sim \mathrm{Bernoulli}(p_{\mathrm{tf}})$. In particular, teacher forcing is chosen for the entire sequence $\mathbf{p}_{1:T}$ if V is 1, and not if otherwise.

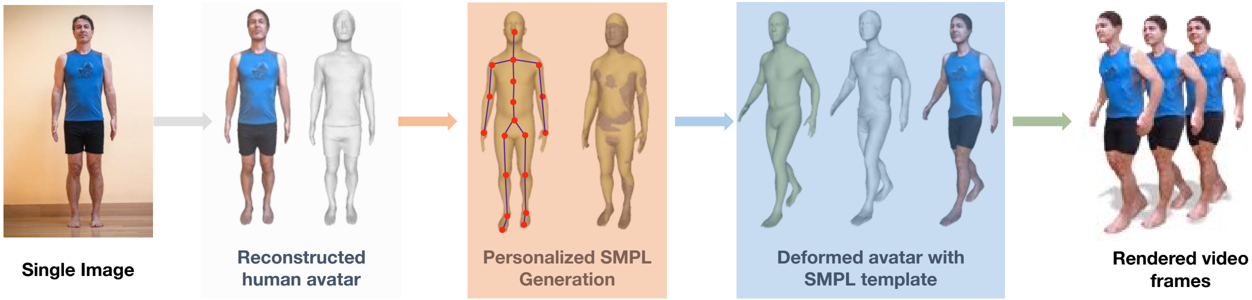

Single Image
Reconstructed human avatar
Personalized SMPL Generation
Deformed avatar with SMPL template
Rendered video frames

Figure 3.5: Illustration of the motion2video process. Shapes and textures of 3D human characters are extracted from single 2D images, that are rigged, animated with motions generated from the action2motion step, and rendered to produce final videos.

As a boundary condition in generating the initial pose $\hat{\mathbf{p}}_1$, its previous pose input $\mathbf{p}_0$ for the prior network (q_ψ) is a zero vector. In addition, *curriculum learning* [206] is used in the training phase that is to progressively increase the value of λ_{kl}.

3.2.2 Step Two: Motion2Video

Recall in step one of our approach, action2motion, diverse motions are generated from prescribed action categories. At this point, a motion is shown as a sequence of 3D skeletal articulations. To produce videos, it remains to settle the full-body shapes and textures of the involved human characters. This is addressed in step two, motion2video, where a specific setup is conceived: a reference person image is presented as input, from which 3D shape and texture of the person are extracted; this is followed by rigging and animating the characters with synthesized motions from the action2motion step, and rendering to generate final 2D videos. Unlike existing motion transfer methods [44, 8, 45] that emphasize in 2D space, our work advocates a fully 3D approach, and we claim our 3D-enabled modelling choice helps to preserve the geometric and appearance aspects in the final video production. Fig. 3.5 illustrates the components in our motion2video process that is to be detailed in the following subsections.

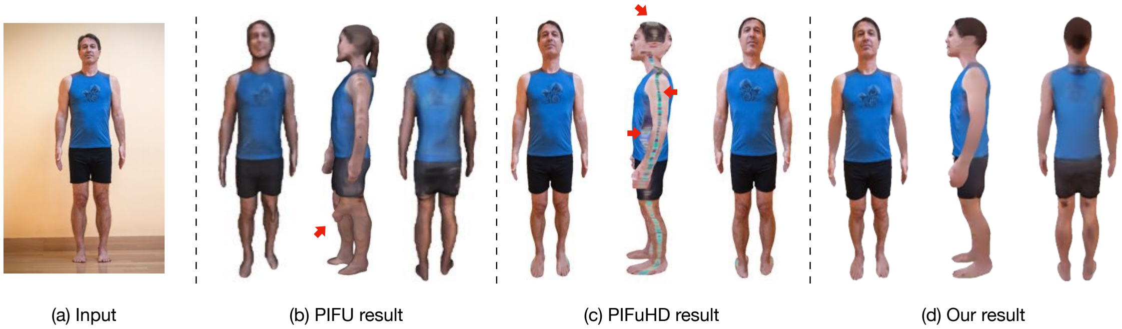


Figure 3.6: A comparison of reconstructing 3D characters from single images by the original methods of PIFu, PIFuHD, and our improved variant. Each 3D reconstruction result is shown in front, side, and back views. Salient errors are pointed by the red arrows. See text for details.

Human Shape Reconstruction from a Single 2D Image

From a single 2D image, a 3D human character is extracted to preserve sufficient geometric and textural details consistent with the input. PIFu [200] and PIFuHD [201] are the two state-of-the-art methods on single-image based human shape recovery that have their unique pros and cons. The 3D shapes and textures extracted by both methods are reasonably adhere to their 2D image inputs. Meanwhile, the texture map extracted by PIFu [200] has relatively low resolution and accuracy, see e.g. the protruded knee pointed by the red arrow in Fig. 3.6(b). Although PIFuHD produces high-resolution 3D human geometry construction, notable errors are introduced at the unseen side by the symmetric assumption. As e.g. shown by the red arrows in Fig. 3.6(c), the frontal human face is also erroneously synthesized at the back side of the 3D character head.

Aiming at refining the reconstruction results, our improved variant takes advantage of PIFuHD in better estimating 3D geometry and camera-view appearance, as well as PIFu in better inpainting of texture for the unseen views. Moreover, we also adopt a heuristic in producing smooth transition near the boundary of visible and occluded surface regions, as follows: to detect the stitching boundary, we project the character (facing Z_+ direction) onto XY plane and match the edge of 2D silhouette with the 3D character; for a point x in the transition region or inside the occluded region O with color c_x, its color c_x is expected to be close to the color c_x^{p} of the corresponding point on PIFu surface; at the same time, c_x should also be close to those of its neighbors, $\mathcal{N}_x$. This is formulated as the following convex objective function,

$$\min \sum_{x \in O} \left[\|c_x - c_x^{\mathrm{p}}\|_2 + \lambda_{nn} \frac{1}{|\mathcal{N}_x|} \sum_{x' \in \mathcal{N}_x} \|c_x - c_{x'}\|_2 \right]. \tag{3.12}$$

In practice, the vertex colors c_x in O are iteratively updated until a consistent convergence. For transition near the boundaries, only the second term of Eq. (3.12) is considered. As shown in Fig. 3.6(d), our result is able to leverage the benefits of of both PIFu and PIFuHD methods, and produces a more natural transition near the boundary regions.

Rigging, Animation, and Rendering

Fitting SMPL for extracted 3D shape. The SMPL human shape, a generative 3D human representation controlled by pose and shape parameters, is used to facilitate the follow-up rigging and animation process. This requires to fit SMPL as close as possible to the reconstructed 3D human shape that amounts to estimating the pose ($\boldsymbol{\theta}$) and shape ($\boldsymbol{\beta}$) parameters by minimizing the following composite objective,

$$\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \mathcal{L}_{\text{surface}}(\boldsymbol{\beta}, \boldsymbol{\theta}) + \lambda_j \mathcal{L}_{\text{joints}}(\boldsymbol{\beta}, \boldsymbol{\theta}) + \lambda_r \mathcal{L}_{\text{reg}}(\boldsymbol{\theta}). \tag{3.13}$$

The joints fitting term $\mathcal{L}_{joints}$ enforces the joints location of the SMPL shape to match with the predicted 3D joints from 2D image. Here, the initial 3D joints prediction $\hat{J}_c$ is obtained by regressing 2D joints from input image with OpenPose [207], and by inverse projection into the reconstructed 3D human shape. Denote $f(\cdot)$ a transformation function of specific joint from initial position to current position following skeleton kinematics chain. Denote $\rho(\cdot)$ a differentiable Geman-McClure penalty function [208], and w the confidence of 2D joint prediction. We have,

$$\mathcal{L}_{\text{joints}}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \sum_{i \in |J|} \omega_i \rho \left(f\left(J(\boldsymbol{\beta})_i, \boldsymbol{\theta}\right) - \hat{J}_{c,i} \right). \tag{3.14}$$

Then the surface fitting term $\mathcal{L}_{\text{surface}}$ is applied to minimize distance between vertex S^i of the reconstructed human shape S and its nearest vertex v of the SMPL shape $\mathcal{M}(\boldsymbol{\beta}, \boldsymbol{\theta})$,

$$\mathcal{L}_{\text{surface}}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \sum_{i \in |S|} \min_{v \in \mathcal{M}(\boldsymbol{\beta}, \boldsymbol{\theta})} \left\| S^i - v \right\|_2 . \tag{3.15}$$

Finally, the pose regularization term $\mathcal{L}_{\text{reg}}(\boldsymbol{\theta})$ penalizes unusual poses through the learned Gaussian mixture model from CMU dataset [189]. Following [209], it is of the form

$$\mathcal{L}_{\text{reg}}(\boldsymbol{\theta}) = -\log \sum_i (g_i N(\boldsymbol{\theta}; \mu_{\boldsymbol{\theta},i}, \Sigma_{\boldsymbol{\theta},i})), \tag{3.16}$$

where $N(\boldsymbol{\theta}; \mu_{\boldsymbol{\theta},i}, \Sigma_{\boldsymbol{\theta},i})$ is a Gaussian distribution with its mean $\mu_{\boldsymbol{\theta},i}$ and variance $\Sigma_{\boldsymbol{\theta},i}$, and g_i are weights of mixture Gaussian model.

In practice, to minimize the above objective function, during the first two iterations we only consider the joints and the pose regularization constraints for quick convergence; the surface constraint is then incorporated during the rest iterations.

3D model deformation and animation. After obtaining the above optimized SMPL model that closely fits to the reconstructed 3D human mesh model, the SMPL model is used as an anchor to deform the 3D models to new poses. To start with, the vertex-level correspondences between the SMPL surface and the 3D human model are established by nearest neighbor search. In addition, body part information is used to eliminate possible mismatched pairs, especially these around the inter-joint of arms and torso. Specifically, the body parts information of reference image could be obtained using DensePose [210], which then are back-projected to the surface of the 3D shape. As SMPL shape has pre-defined body segmentation, this could be utilized to filter out vertex pairs coming from different body parts. Next, we compute a displacement map from the optimized SMPL mesh to their correspondences on the 3D human model,

$$S^j = \mathcal{M}_i(\boldsymbol{\beta}^*, \boldsymbol{\theta}^*) + d_{i \to j}. \tag{3.17}$$

where $\boldsymbol{\beta}^*$ and $\boldsymbol{\theta}^*$ are the optimized shape and pose parameters of the SMPL model. S_j and $\mathcal{M}_i(\boldsymbol{\beta}^*, \boldsymbol{\theta}^*)$ are the correspondences and $d_{i\to j}$ is the displacement from optimized SMPL model to reconstructed 3D human model.

Intuitively, to repose the human shape, we could acquire the target positions S^* of shape vertices by applying the displacement map to the reposed SMPL as in Eq.(3.17). However, this will lead to imperfections due to free-form deformation. Following [211], we instead utilize the vertices of S^* as control points to deform the 3D human model as rigid as possible, by enforcing a local rigidity constraint. The locally rigid deformation $\boldsymbol{R}$ and the deformed human model $\hat{S}$ are obtained by minimizing the following objective,

$$\begin{aligned}\mathcal{L}_{def}(\boldsymbol{R}, \hat{S}) = & \sum_{i\in|S|}\sum_{j\in\mathcal{N}_i} k_{ij} \left\|(\hat{S}^i - \hat{S}^j) - \boldsymbol{R}_i(S^i - S^j)\right\|_2 \\ & + \sum_{l\in|S|} \left\|\hat{S}^l - S^{*,l}\right\|_2 .\end{aligned} \tag{3.18}$$

Here $\mathcal{N}_i$ is the set of the neighboring vertices of S^i; k_{ij} is the corresponding weights of neighboring vertices. $\boldsymbol{R}_i$ is a rotation matrix. The above objective function is optimized by iteratively solving the rotation matrix R and the deformed mesh $\hat{S}$ [212].

Rendering. The target 3D shape are deformed and driven by the generated pose sequences frame-by-frame, which are subsequently fed into 3D game engine (Unity3D) to integrate physical conditions such as illuminations and shadows and produce the final videos. Specifically, spot light and directional light are used to illuminate the character from top. Four cameras, fixed at half height of the 3D character, are aimed at the subject to record the *front, back, left side* and *right side* views, respectively.

3.3 Empirical Evaluations

A comprehensive set of experiments are conducted to systematically evaluate the performance of our action2video approach, which consists of the two-step pipeline of action2motion and motion2video.

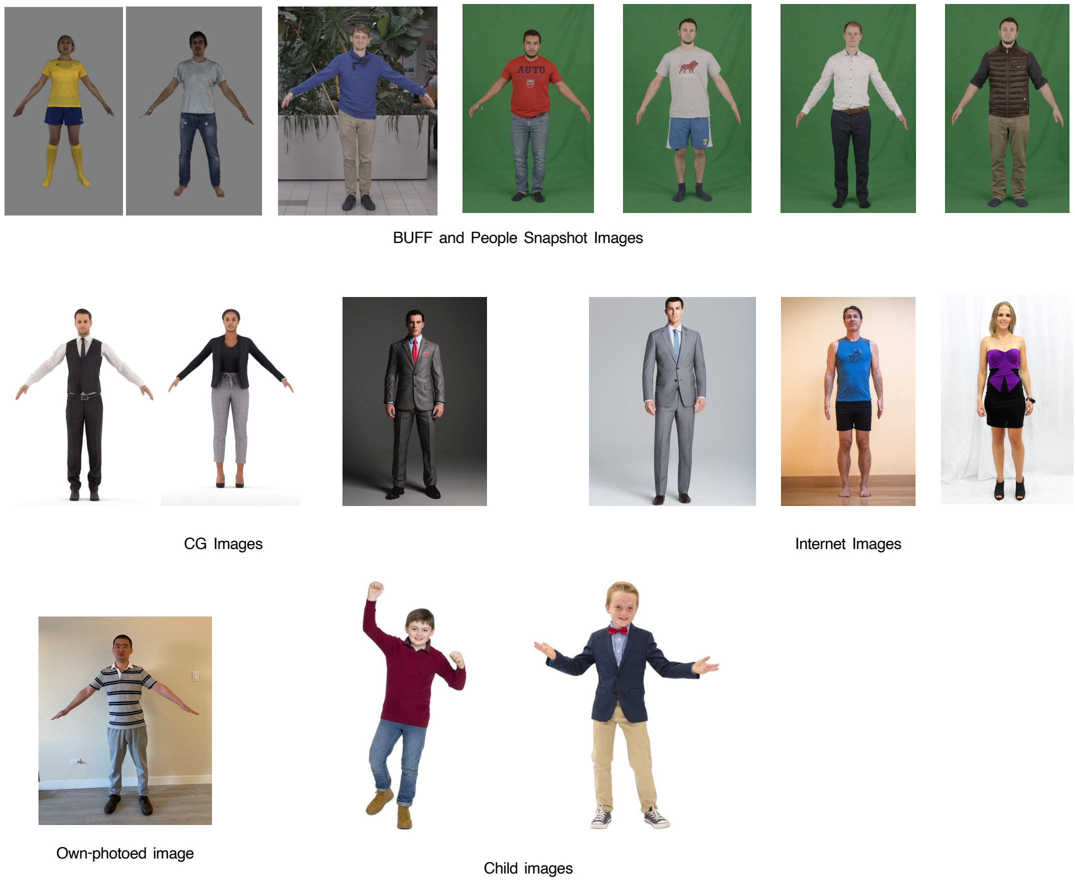


Figure 3.7: Input images used in our experiments are from different sources, including (a) BUFF dataset [213], (b) People Snapshot dataset [214], (c) internet images, (d) CG image, and (e) our in-house captured images. See text for details.

We start by introducing the related datasets, and our implementation details. This is followed by a detailed examination of our action2motion process at Sec. 3.3.1, and comparisons for our motion2video with related efforts at Sec.3.3.2. Finally, Sec. 3.3.3 provides a holistic evaluation of our full pipeline, action2video.

Datasets. Ideally, we expect to work with motion datasets that contain considerable amount of distinct motion clips of various action categories, and with proper 3D pose annotations. In practice, we achieve this by postprocessing existing popular datasets, including re-annotating 3D positions of

NTU-RGBD [42] and action categories of CMU MoCap [189]. We also curate an in-house dataset, HumanAct12. In these three datasets, all human poses are uniformly annotated into 3D joints connected into 5 kinematics chains, with pelvis being the root joint.

- **NTU-RGBD** is a large-scale 3D human motion dataset containing nearly one million motion sequences of 120 action types. Its pose annotation (i.e. 3D joint positions) is from MS Kinect readout, which is known unreliable and temporally unstable. In our experiments, the state-of-art video 3D shape estimation method [215] is employed to re-estimate the 3D poses from video feeds. Note in our scenario, it's sufficient for these poses to appear realistic, and they are not necessarily matched perfectly with the true poses. A subset of 13 distinct actions are further selected in our empirical evaluation, such as *cheer up, pick up, salute*, consisting of 3,900 motion clips. Each pose is represented by 18 joints (i.e. 17 bones).

- **CMU MoCap** is dataset accurately annotated by motion capture markers, with 2,605 pose sequences. However, the dataset is not originally organized by action types. We identify 8 distinct actions based on their motion captions, including *running, walking, climbing, jumping.* In the end, 1,088 motions are re-organized by action type, with each skeleton constituting 22 3D joints (i.e. 21 bones). In implementation, these pose sequences are down-sampled from 100 HZ to a frequency of 12 HZ.

- **HumanAct12** is our in-house dataset that comes with proper annotations. It consists of 1,191 motion clips and 90,099 frames in total, which are categorized into 12 coarse-grained action categories, including e.g. *warm up, lift dumbbell*, and 34 fine-grained action types such as *warm up (Leg pressing), lift dumbbell (with right hand).* The fine-grained annotations give more specific and dedicated information of the motions. We test our model on both coarse- and fine-grained annotations. Our dataset, HumanAct12, contains more accurate and stable 3D position annotations compared to NTU-RGBD; and has more well-organized action annotations than CMU MoCap. Note each body pose contains 24 joints (i.e. 23 bones). The detailed distribution over action categories can be find in table Sec. 3.3.

To showcase that our pipeline could work with wide range of applications, input images from

Coarse-grained Label	Fine-grained Label	Number of Motions	Total Number
Warm up	Warm_up_wristankle	25	215
	Warm_up_pectoral	49	
	Warm_up_eblowback	43	
	Warm_up_bodylean_right_arm	26	
	Warm_up_bodylean_left_arm	24	
	Warm_up_bow_right	24	
	Warm_up_bow_left	24	
Walk	Walk	47	47
Run	Run	50	50
Jump	Jump_handsup	54	94
	Jump_vertical	40	
Drink	Drink_bottle_righthand	27	88
	Drink_bottle_lefthand	43	
	Drink_cup_righthand	11	
	Drink_cup_lefthand	3	
	Drink_both_hands	4	
Lift_dumbbell	Lift_dumbbell_righthand	45	218
	Lift_dumbbell_lefthand	45	
	Lift_dumbbell_bothhands	47	
	Lift_dumbbell_overhead	43	
	Lift_dumbbell_bothhands_bend_legs	38	
Sit	Sit	54	54
Eat	Eat_righthand	33	77
	Eat_lefthand	25	
	Eat_pie/burger	19	
Turn_steering_wheel	Turn_steering_wheel	56	56
Phone	Take out phone, call and put back	28	61
	Call with left hand	33	
Boxing	Boxing_left_right	26	140
	Boxing_left_upwards	39	
	Boxing_right_upwards	41	
	Boxing_right_left	34	
Throw	Throw_right_hand	53	91
	Throw_both_hand	38	
Entire Dataset	-	-	**1191**

Table 3.1: Statistics of dataset HumanAct12.

myriad sources are considered in our experiments, as displayed in Fig. 3.7. They include images from the BUFF dataset [213], People Snapshot dataset [214], as well internet images, computer-generated (CG) images [1], and our in-house captured images. BUFF dataset provides 26 4D human sequences with different cloth styles and performing different actions. We then render 2D images from these human shapes. People Snapshot dataset contains 12 subjects and 24 video sequences with different backgrounds.

Implementation Details. Our action2motion pipeline is implemented by PyTorch. For all

[1] https://renderpeople.com/3D-people/

encoder layers, the output size is set to 128. One-layer GRU is used for prior network, posterior network and pose decoding module, while generator uses two-layer GRU. The hidden unit size of GRU is 128. And the noise vector $\mathbf{z}$ and $\mathbf{h}_t^o$ has the dimension of 30 and 20 respectively. The Adam optimizer is applied for training throughout all experiments, with learning rate of 0.0002, weight decaying of 0.00001, and default parameter values including $\beta_1 = 0.9$, $\beta_2 = 0.999$. Our model is trained with mini-batch size of 128. To stabilize the training process, *teacher forcing rate* p_{tf} is set to 0.6. The values of aforementioned hyper-parameters are fixed throughout our empirical experiments across all datasets.

Afterwards, we generate motions with length of 60, 100 and 60 on NTU-RGBD, CMU MoCap and HumanAct12, respectively. The hyper-parameter λ_{kl} is a trade-off between reconstruction constraints and KL-divergence penalty. During training, the value of λ_{kl} for all datasets are initialized with 0.001 and linearly increased to 0.1, 0.1 and 0.01 at the end for above datasets respectively. During training, the value of λ_{align} is set to 10 throughout these experiments. By default, the action2motion GLMI-M variant is utilized in our approach.

To extract 3D shape from single image, λ_{nn} and 10 neighbors are used in Eq. (3.12) for occluded region. The values of λ_j and λ_r in Eq. (3.13) are set to 2.0 and 0.2, respectively.

3.3.1 Step1: Action2Motion

Thorough evaluations of the action2motion step are carried out in this section. They include both quantitative and qualitative reports of motion generation results, and fine-grained analysis of the locomotion generation module; We also provide demonstrations of specific action2motion applications such as motion interpolation in the latent space, motion transition, and 3D motion outpainting. By default, the action2motion GLMI-M variant is utilized in our approach.

Evaluations

We start by introducing a tally of evaluation metrics and baseline methods used throughout this section, which is followed by a series of qualitative and quantitative evaluations.

Evaluation Metrics. We aim to evaluate the generated motions from the aspects of being *natural* and *diverse*. To achieve this, the three metrics in [53] are adopted in our evaluations: *Frechet Inception Distance(FID)* to characterize the visually realistic aspect, *Diversity* and *Multimodality* to quantify the diverse levels. The *action recognition accuracy* is additionally used to gauge the similarity between generated motions and real-life motions, as well as the degree of generated motions belonging to the prescribed action.

FID is perhaps the most important indicator in our scenario. A *lower* FID suggests a better result. For multimodality and diversity, a result is claimed better only if its diversity and multimodality scores are **closer** to their respective values obtained from real motions. To calculate these metrics, we rely on a feature extractor to obtain the high-level features of motions. Since there is no standard implementation of such motion feature extraction, a vanilla RNN action recognition classifier is trained for each dataset; and the final layer of classifier is used as the motion feature extractor.

We elaborate these four metrics as below:

- **Frechet Inception Distance**(FID): FID is an effective metric to evaluate the overall quality in motion generation. A large amount (in our case, 3,000) of generated motions and real motions are sampled and then are transformed to two sets of features. For real motion, we sample from test set with replacement. Then, FID is measured by computing the distance between the feature distribution of generated motions and that of the real motions.
- **Recognition Accuracy**: Recognition accuracy is calculated as the accuracy of applying a pre-trained RNN action recognition classifier to the motion of interest.
- **Diversity**: Diversity indicates the variance of the motions across *all* action types. Specifically, a large set of motions are sampled from all varieties of action types, from which two subsets are randomly sampled with the same size S_d. The corresponding sets of motion feature vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_{S_d}\}$ and $\{\mathbf{v}'_1, \ldots, \mathbf{v}'_{S_d}\}$ are extracted respectively. Then, the diversity

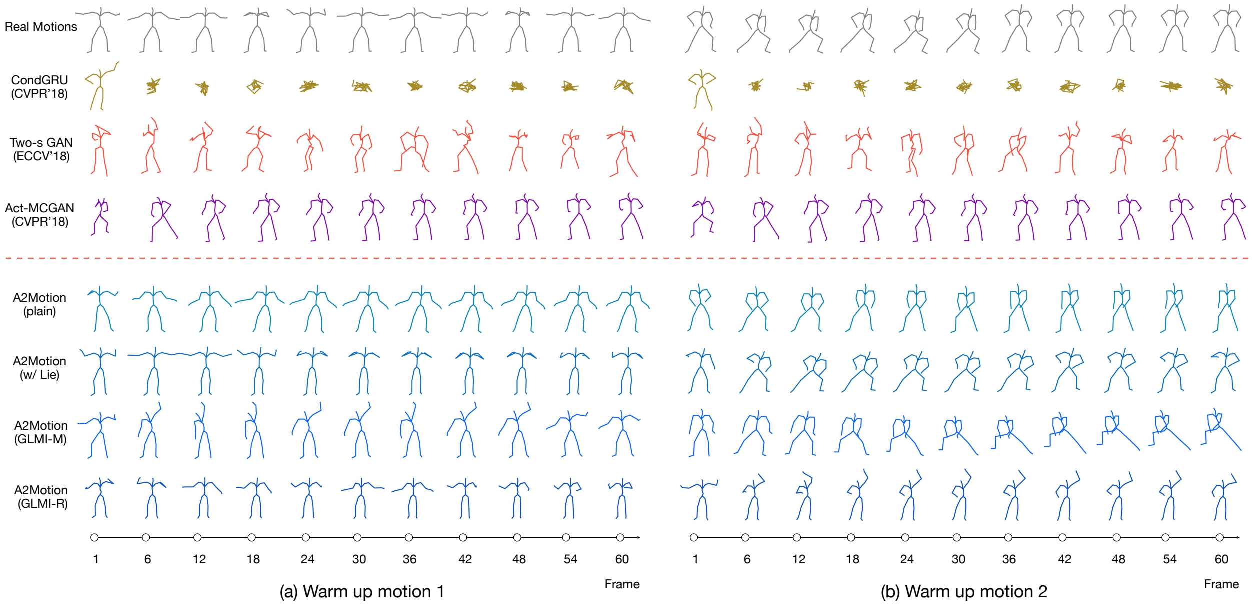


Figure 3.8: Visual comparison of motions generated by the baseline methods and our four action2motion variants. Two *warm up* motion sequences are sampled for each of the comparison methods. Every 6th frame is shown. See text for details.

of this set of motions is evaluated by

$$\text{Diversity} = \frac{1}{S_d} \sum_{i=1}^{S_d} \| \mathbf{v}_i - \mathbf{v}'_i \|_2, \tag{3.19}$$

where $S_d = 200$ is used throughout our experiments.

- **Multimoldality**: Different from diversity, multimodality indicates how much the sampled motions vary within *each* action category. Suppose there are C action types in the set of motion sequences. For the c-th action, two subsets with same size S_m are randomly sampled, which are then transformed to two subset of feature vectors $\{\mathbf{v}_{c,1}, \dots, \mathbf{v}_{c,S_m}\}$ and $\{\mathbf{v}'_{c,1}, \dots, \mathbf{v}'_{c,S_m}\}$. The multimodality is defined as

$$\text{Multimodality} = \frac{1}{C \times S_m} \sum_{c=1}^{C} \sum_{i=1}^{S_m} \left\| \mathbf{v}_{c,i} - \mathbf{v}'_{c,i} \right\|_2, \tag{3.20}$$

where $S_m = 20$ is used in our experiments.

Baseline methods. Since the problem of action2motion, aka action-conditioned 3D human motion generation, is relatively new, there are few existing methods to compare with. We thus adapt the state-of-art methods from related areas to our context, as follows:

- **CondGRU**. Condition GRU is used as a deterministic baseline in our setting, which is also the principal model for audio-to-motion translation in [51] and text-to-motion generation in [216, 217]. Here, a small modification of the model is made that the input is the concatenation of condition vector and pose vector at present step and the output is the pose vector for next step.
- **Two-stage GAN**. [33] propose a two-stage GAN method for 2D human motion generation based on action types. In particular, a Wasserstein GAN [218] is first trained as the pose generator. After that, the motion generator is learned to produce input latent vector for pose generator to synthesize pose at each time. By using adversarial training, the entire generated pose sequences are judged by a motion discriminator. We adapt this method for 3D human motion generation through necessary modifications.
- **Act-MoCoGAN**. MoCoGAN [21] is a widely used method for both conditional and unconditional video generation. While generating a video, the input noise vector are composed of two parts: one is a shared vector over time, another is a instinct noise vector sampled at each time. These two inputs are expected to map to the stationary content and dynamic motions in videos. In our experiment, to generate 3D human dynamics, we keep the original architecture and replace the video and image discriminators to motion and pose discriminators, respectively.
- **Dancing2Music**. Dancing2Music [53] generates 2D dancing motion sequences from audio signals, which consists of two main stages, decomposition and composition. During decomposition, a motion sequence is segmented into short motion snippets, with dance unit VAE (DU-VAE) model being trained to generate these motion snippets given the latent vectors of motion content and an initial frame; during composition, a music-to-movement GAN (MM-GAN) is trained to generate latent vectors of motion snippet contents conditioned on the given music signals. To make a meaningful comparison, the official implementation is adapted by

replacing the music signals with action categories.

- **LatentTransition**. [147] consider a two-stage GAN [33], with a Bi-LSTM being employed to produce input latent vectors for pose generation. An additional auxiliary action classifier further ensures the action-awareness of the generative model.

- **Action2Motion (plain)**. Oue action2motion variant by adopting the pose decoding module of Fig. 3.4(a), where the 3D position of joints are directly produced from generator.

- **Action2Motion (w/ Lie)**. Our action2motion variant with the pose decoding module of Fig. 3.4(b), where the Lie algebra parameters and root joint locations are generated independently.

- **Action2Motion (GLMI-M)**. Our action2motion variant with the pose decoding module of Fig. 3.4(c), where both the Lie algebra and GLMI are used, and GLMI is implemented by MLP.

- **Action2Motion (GLMI-R)**. Our action2motion variant with the pose decoding module of Fig. 3.4(d), where both the Lie algebra and GLMI are used, and GLMI is implemented by GRU network instead.

Visual comparisons. Fig. 3.8 provides qualitative comparisons of skeletal motions generated from different methods: given an action category of *warm up*, two motions of length 60 are sampled, with every 6th frame being displayed.

Conditional GRU [51] requires as input an initial ground-truth pose to kick-start its generation process. Unfortunately the generated poses often collapse into a cloud of 3D points near the root joint. Two-stage GAN [33] produces better results, which however are still perceptually not satisfactory. The skeletal sequence result of Act-MoCoGAN by [21] is visually the best among these three methods. The generated poses nonetheless often froze to a fixed posture quickly. Dancing2Music [53] shows capability of yielding natural poses and motions. Meanwhile, a single such motion usually contains multiple actions, with the motion context deviating from the prescribed action type. For instance, in the left column of Fig. 3.8, the stick man first performs *lift dumbbell*

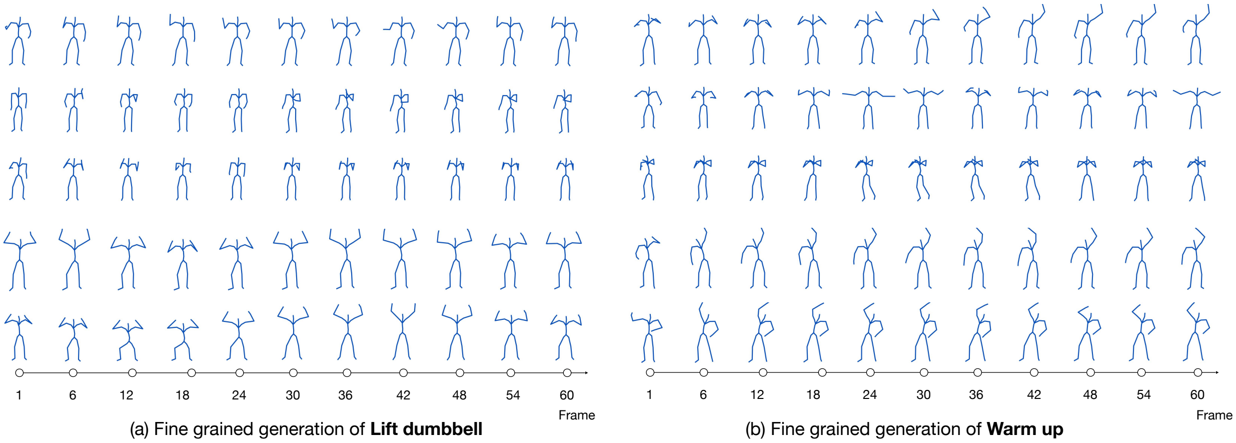

1
6
12
18
24
30
36
42
48
54
60
Frame
(a) Fine grained generation of **Lift dumbbell**
1
6
12
18
24
30
36
42
48
54
60
Frame
(b) Fine grained generation of **Warm up**

Figure 3.9: Motion examples of fine-grained action categories generated by our action2motion (GLMI-M). Every 6th frame is shown. (a) Lift dumbbell with (from top to bottom) *right hand, left hand, both hand, both hand over head*, and *both hand over head and squat.* (b) Warm up with (from top to bottom) *alt chest expansion, chest expansion, wrist circles, left side reach* and *right side reach.*

(from $t = 1$ to $t = 18$), then a short-time *warm up* (from $t = 24$ to $t = 36$), and finally drifting into *drinking*. On the other hand, LatentTransition [147] always starts with natural poses, then struggles with proper modeling of long-term motion dependencies, which typically deteriorates to unrecognizable movements. These results are in sharp contrast to that of our four action2motion variants, whose results are in general visually more appealing. Here, the action2motion (plain) variant sometimes generate visual defects noticeable to human eyes. For example, in the left column of Fig. 3.8, the arm bone lengths of the same individual abnormally vary from $t = 1$ to $t = 24$. This is due to the intrinsic 3D-coordinate skeletal representation adopted by the plain variant that does not obey the underlying skeletal kinematics. Skeletal motions generated by the other action2motion variants are typically more faithfully resemble to real-life motions, which we attribute to their adherence to kinematics by their use of Lie group/algebraic skeletal representations.

Diversity is another important evaluation criteria. In Fig. 3.8, motions generated from conditional GRU tends to be visually least appealing; this is followed by those of two-stage GAN and LatentTransition; the results of Act-MoCoGAN often suffers from the *mode collapsing* issue, with similar results popping up after multiple separate runs; In comparison, Dancing2Music is capable of producing diverse motions by transiting between different short motion snippets. However, the

Methods	HumanAct12 (Coarse-grained)				HumanAct12 (Fine-grained)			
	FID↓	Accuracy↑	Diversity→	MModality→	FID↓	Accuracy↑	Diversity→	MModality→
Real motions	$0.092^{\pm.007}$	$0.997^{\pm.001}$	$6.853^{\pm.053}$	$2.449^{\pm.038}$	$0.133^{\pm.004}$	$0.991^{\pm.001}$	$7.001^{\pm.018}$	$2.666^{\pm.012}$
CondGRU	$40.61^{\pm.144}$	$0.080^{\pm.002}$	$2.381^{\pm.020}$	$\mathbf{2.341}^{\pm.036}$	$33.91^{\pm.059}$	$0.034^{\pm.001}$	$3.779^{\pm.034}$	$3.469^{\pm.026}$
Two-stage GAN	$10.48^{\pm.089}$	$0.421^{\pm.006}$	$5.960^{\pm.049}$	$\underline{2.805}^{\pm.036}$	$6.956^{\pm.038}$	$0.397^{\pm.002}$	$6.151^{\pm.017}$	$\mathbf{2.694}^{\pm.008}$
Act-MoCoGAN	$5.610^{\pm.113}$	$0.793^{\pm.004}$	$\mathbf{6.752}^{\pm.071}$	$1.055^{\pm.017}$	$2.468^{\pm.026}$	$\mathbf{0.832}^{\pm.002}$	$\underline{6.891}^{\pm.023}$	$0.878^{\pm.003}$
Dancing2Music	$3.832^{\pm.103}$	$0.145^{\pm.003}$	$6.523^{\pm.096}$	$6.313^{\pm.035}$	$3.484^{\pm.085}$	$0.029^{\pm.001}$	$6.567^{\pm.106}$	$6.406^{\pm.026}$
LatentTransition	$3.553^{\pm.093}$	$0.471^{\pm.005}$	$6.580^{\pm.110}$	$4.387^{\pm.039}$	$2.123^{\pm.044}$	$0.397^{\pm.004}$	$6.640^{\pm.082}$	$4.590^{\pm.027}$
Action2Motion (plain)	$3.299^{\pm.079}$	$0.656^{\pm.005}$	$\underline{6.742}^{\pm.046}$	$4.248^{\pm.037}$	$1.329^{\pm.021}$	$0.560^{\pm.002}$	$6.756^{\pm.015}$	$4.487^{\pm.015}$
Action2Motion (w/ Lie)	$2.458^{\pm.079}$	$\mathbf{0.923}^{\pm.002}$	$7.032^{\pm.038}$	$2.870^{\pm.037}$	$1.000^{\pm.016}$	$0.776^{\pm.001}$	$6.783^{\pm.015}$	$3.508^{\pm.011}$
Action2Motion (GLMI-M)	$\mathbf{2.157}^{\pm.052}$	$\underline{0.835}^{\pm.005}$	$6.986^{\pm.028}$	$3.633^{\pm.031}$	$\mathbf{0.739}^{\pm.015}$	$\underline{0.787}^{\pm.002}$	$6.783^{\pm.015}$	$\underline{3.301}^{\pm.009}$
Action2Motion (GLMI-R)	$\underline{2.349}^{\pm.057}$	$0.831^{\pm.002}$	$7.001^{\pm.023}$	$3.607^{\pm.037}$	$\underline{0.957}^{\pm.017}$	$0.767^{\pm.001}$	$\mathbf{6.924}^{\pm.019}$	$3.303^{\pm.012}$

Table 3.2: Performance evaluation on HumanAct12 benchmark on coarse-grained and fine-grained action categories, respectively. ± indicates 95% confidence interval. ↑ (or ↓) is higher (or lower) the better; → means closer to real motion scores the better. For performance, **bold** face specifies the best method, with underscore referring to the second best.

Methods	CMU MoCap				NTU-RGBD			
	FID↓	Accuracy↑	Diversity→	MModality→	FID↓	Accuracy↑	Diversity→	MModality→
Real motions	$0.064^{\pm.006}$	$0.936^{\pm.002}$	$6.130^{\pm.079}$	$2.726^{\pm.066}$	$0.031^{\pm.004}$	$0.999^{\pm.001}$	$7.108^{\pm.048}$	$2.194^{\pm.025}$
CondGRU	$51.72^{\pm.123}$	$0.093^{\pm.001}$	$0.792^{\pm.011}$	$0.752^{\pm.016}$	$28.31^{\pm.138}$	$0.078^{\pm.001}$	$3.663^{\pm.024}$	$3.578^{\pm.027}$
Two-stage GAN	$14.34^{\pm.107}$	$0.179^{\pm.003}$	$4.419^{\pm.064}$	$\mathbf{1.623}^{\pm.024}$	$13.86^{\pm.091}$	$0.202^{\pm.003}$	$5.328^{\pm.039}$	$3.490^{\pm.027}$
Act-MoCoGAN	$11.15^{\pm.074}$	$0.445^{\pm.005}$	$5.280^{\pm.069}$	$\underline{1.516}^{\pm.022}$	$2.723^{\pm.019}$	$\mathbf{0.997}^{\pm.001}$	$6.920^{\pm.061}$	$0.907^{\pm.009}$
Dancing2Music	$6.882^{\pm.127}$	$0.138^{\pm.003}$	$4.772^{\pm.104}$	$4.289^{\pm.012}$	$3.461^{\pm.077}$	$0.075^{\pm.002}$	$6.562^{\pm.114}$	$6.556^{\pm.045}$
LatentTransition	$12.85^{\pm.181}$	$0.389^{\pm.003}$	$5.856^{\pm.143}$	$4.639^{\pm.053}$	$6.882^{\pm.127}$	$0.138^{\pm.003}$	$4.772^{\pm.105}$	$4.289^{\pm.049}$
Action2Motion (plain)	$2.994^{\pm.052}$	$0.378^{\pm.004}$	$\underline{5.791}^{\pm.044}$	$5.006^{\pm.045}$	$\underline{0.540}^{\pm.047}$	$0.832^{\pm.004}$	$\underline{6.926}^{\pm.049}$	$\underline{3.443}^{\pm.052}$
Action2Motion (w/ Lie)	$2.885^{\pm.116}$	$\mathbf{0.686}^{\pm.003}$	$6.509^{\pm.061}$	$4.126^{\pm.056}$	$\mathbf{0.330}^{\pm.008}$	$\underline{0.949}^{\pm.001}$	$\mathbf{7.065}^{\pm.043}$	$\mathbf{2.052}^{\pm.030}$
Action2Motion (GLMI-M)	$\mathbf{2.448}^{\pm.031}$	$0.665^{\pm.001}$	$\mathbf{6.374}^{\pm.022}$	$4.093^{\pm.019}$	-	-	-	-
Action2Motion (GLMI-R)	$\underline{2.519}^{\pm.029}$	$\underline{0.675}^{\pm.001}$	$6.484^{\pm.028}$	$4.073^{\pm.029}$	-	-	-	-

Table 3.3: Performance evaluation on CMU MoCap and NTU-RGBD Dataset. ± indicates 95% confidence interval. As NTU-RGBD dataset does not have global motion trajectory annotations available, our GLMI-M & GLMI-R variants that could not be fairly evaluated here.

generated motions could not be faithfully aligned to the prescribed action type; On the contrary, our action2motion variants are shown to be capable of generating both diverse and consistent motions.

Moreover, our action2motion framework is also capable of producing motions from fine-grained action categories, as showcased in Fig. 3.9. The motions generated by our action2motion (GLMI-M) variant faithfully assemble the subtle characteristics of local motions (e.g. leg pressing and chest expansion), and body parts (e.g. left hand and right hand) from a range of fine-grained action types.

Quantitative comparisons. Quantitative evaluations are conducted on a range of datasets. Specifically, Table 3.2 displays results on our in-house HumanAct12 dataset, where coarse-grained and fine-grained action annotations are both considered; Table 3.3 presents comparison results on the popular benchmarks of CMU MoCap and NTU-RGBD. Considering the stochastic nature of motion generation, each experiment is repeated 20 times, a statistical confidence interval of 95% is reported in both tables. Note action2motion (GLMI) is however not applicable to the post-processed NTU-RGBD dataset, since the re-estimated pose sequences from videos does not contain global trajectory information.

Among the four evaluation metrics in both tables, FID is perhaps the most important indicator, as it evaluates the overall quality of the generated motions. Recognition accuracy quantifies how well a generated motion fits into an action category. Diversity and multimodality (i.e. MModality) are metrics quantifying the diversity aspects of the generated motions. Note the values of FID (or accuracy) is lower (or higher) the better; for Diversity and MModality though the values are as close to the real motion scores the better. From Table 3.2 and Table 3.3, we have the following observations. As a deterministic method, conditional GRU fails to generate diverse motions that is essentially an one-to-many mapping problem. GAN models such as two-stage GAN, Act-MoCoGAN and Latent-Transition have improved upon conditional GRU in both metrics of FID and recognition accuracy. The considerably high accuracy obtained by Act-MoCoGAN may be attributed to its use of action classifier during training. A sharp drop of FID is observed in Dancing2Music, which however comes at the price of much lower accuracy. Meanwhile, our action2motion clearly outperforms the rest on FID, and the GLMI-M variant consistently excels among the four action2motion variants. The success could be partly attributed to the incorporation of Lie algebraic pose representation.

Given substantial performance on FID and perhaps also accuracy scores, the scores of diversity and multimodality are also important indicators for the model capacity of producing diverse motions. Note for diversity and multimodality, the higher values do not necessarily reflect better performance; instead the values are best to be close to those from the real motions, denoted as → in Tables 3.2 and 3.3. Act-MoCoGAN generates motions with severely limited diversity. Overall, our action2motion variants, while performing best on FID and accuracy, also maintain a considerable extent of diversity and multimodality.

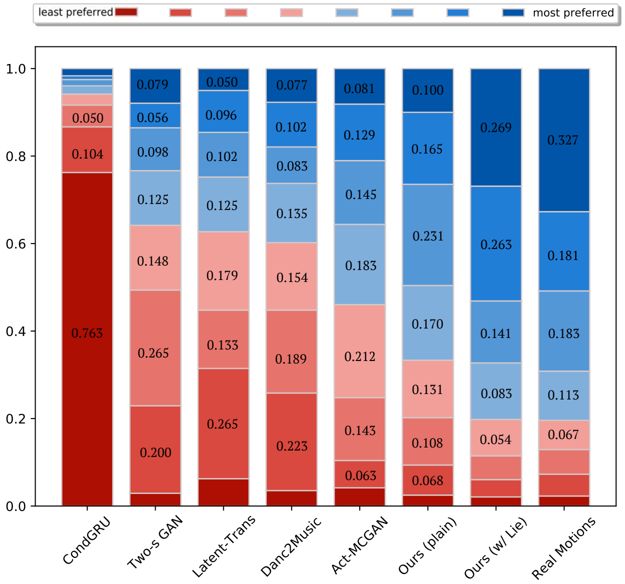


Figure 3.10: Crowd-sourced subjective assessment results of motions generated by comparison methods. For each method, there is a bar of different colors (from red to blue) indicating the percentage of corresponding preference levels (least to most preferred). See text for details.

Crowd-sourced Subjective Evaluation. In addition to the aforementioned objective experiments, two user studies are conducted on Amazon Mechanical Turk. The principal criteria used in these two user surveys are the visual perceptual quality of the motion, and the magnitude it is adhere to the intended action categories. 40 users who possess hit approval rate higher than 97% and 1000 completed hits are considered.

The first user study is illustrated in Fig. 3.10, which compares the first two action2motion variants, ours (plain) and ours (w/ Lie), with baseline methods. Here, same amount (i.e. 36) of motions are generated by different methods. The users are then asked to rank their preferences of these motions evenly sampled over all action categories. Our action2motion variants receive the highest user ratings. Contrarily, conditional RNN, two-stage GAN and LatentTransition are the three least performed methods. Dancing2Music and Act-MoCoGAN rank somewhere in-between. More positive feedback is observed in our action2motion *plain* variant, with 10% motions being graded the first by users. By adopting the Lie algebraic representation, our ours w/ Lie variant further narrows the gap to real motions, with 54% generated motions being secured at the top-2 spots by user ratings.

The second user study compares bewteen our two action2motion variants: ours (GLMI) and ours (w/ Lie). As GLMI-M outperforms GLMI-R in most cases, we focus on the evaluation of GLMI-M in this survey. Here the motions are generated following the same protocol conceived in the first study. As shown in table 3.4, ours with GLMI earns more appreciation from users when compared with ours (w/ Lie), with over a half motion sequences (i.e. 54.4 %) being preferred by

Preference	Percentage
Ours (GLMI-M) Over *Ours (w/ Lie)*	0.544
Ours (w/ Lie) Over *Real Motions*	0.462
Ours (GLMI-M) Over *Real Motions*	0.501

Table 3.4: Crowd-sourced subjective assessment to compare motions sampled from **Ours (GLMI-M)**, **Ours (w/ Lie)**, and real motions.

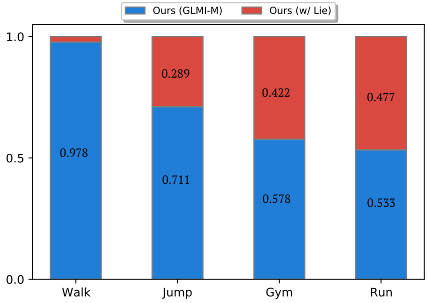


Table 3.5: Crowd-sourced subjective assessment to compare generated motions together with their global displacements from **Ours (GLMI-M)** and **Ours (w/ Lie)**.

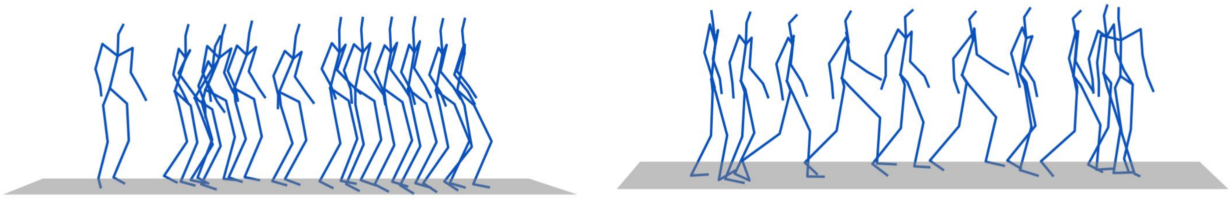


Figure 3.11: Examples of locomotion generated without GLMI (top) vs. with GLMI (bottom). Note the *ghosting* manoeuvre patterns when without GLMI.

users. When comparing to real motions, samples generated by ours (w/ Lie) are slightly inferior to real-life human motions, with 46.2% being preferred. Meanwhile ours (GLMI-M) is almost indistinguishable to the real motions. The results suggest the potentials of applying our algorithm to more interesting VR/AR applications.

We further investigate the global displacement aspect of the generated motions. As demonstrated in Fig. 3.5, motions generated from ours (GLMI) are always more preferred by users than those from ours (w/ Lie) over all these four action categories.

In summary, our GLMI-M variant, i.e. ours (GLMI), delivers overall best results among our four action2motion variants, which are often indistinguishable from real-life human motions.

Locomotion Generation Analysis

Methods	Walk			Jump Forward		
	FID↓	Accuracy↑	Diversity→	FID↓	Accuracy↑	Diversity→
Real motions	$0.148^{\pm.007}$	$0.999^{\pm.001}$	$2.618^{\pm.013}$	$0.135^{\pm.006}$	$0.999^{\pm.001}$	$2.711^{\pm.015}$
Action2Motion (plain)	$6.659^{\pm.119}$	$0.755^{\pm.002}$	$4.379^{\pm.026}$	$13.14^{\pm.104}$	$0.226^{\pm.004}$	$5.412^{\pm.018}$
Action2Motion (w/ Lie)	$5.392^{\pm.069}$	$0.786^{\pm.003}$	$4.200^{\pm.031}$	$7.233^{\pm.124}$	$0.523^{\pm.004}$	$5.398^{\pm.018}$
Action2Motion (GLMI-R)	$2.096^{\pm.057}$	$0.930^{\pm.002}$	$3.471^{\pm.020}$	**3.796**$^{\pm.083}$	**0.749**$^{\pm.018}$	**4.662**$^{\pm.031}$
Action2Motion (GLMI-M)	**1.183**$^{\pm.028}$	**0.967**$^{\pm.001}$	**3.059**$^{\pm.022}$	$4.443^{\pm.146}$	$0.715^{\pm.005}$	$4.747^{\pm.031}$

Table 3.6: Performance evaluation over CMU MoCap dataset on two locomotion action types. ± indicates 95% confidence interval. ↑ (or ↓) is higher (or lower) the better; → means closer to real motion scores the better. For performance, Bold face specifies the best method, with underscore referring to the second best.

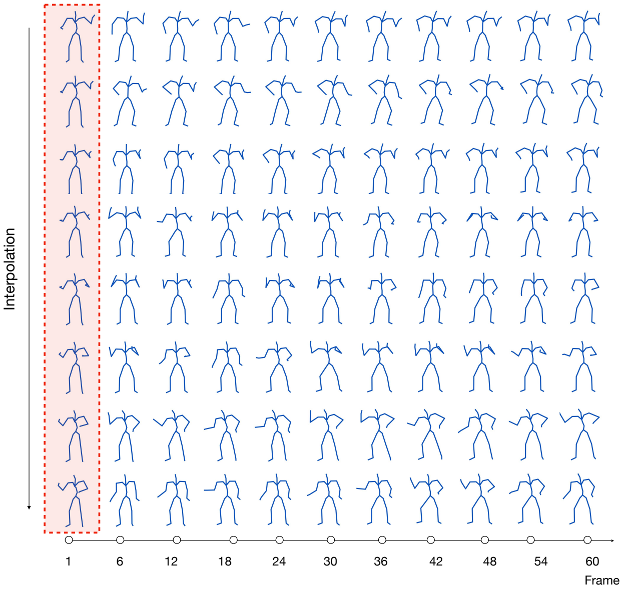


Figure 3.12: Examples of motion interpolation in *lift dumbbell*. Every 6th frame is shown. See text for details.

Locomotions (e.g. walking) are the most common activities in our daily life, which typically involve full-body displacements. Fig. 3.11 visually compares walking motions produced with vs. without our global local movement integration (GLMI) module. When without, the walking motions appear surreal like *ghost* haunting on the ground, with arm and leg local movements not tuned to its global motion trajectory. By contrast, our proposed GLMI module significantly mitigates these issues. For example, the waving patterns of left (or right) arm is now synchronized with the right (or left) leg; the local-part movements are also well in agreement with the full-body motion trajectories.

Table 3.6 quantitatively evaluates the effects of incorporating GLMI module for locomotion generation on CMU MoCap dataset. The same evaluation metrics of Section 3.3.1 are considered here. The number of motion sampling is set to 500. Overall, ours with GLMI variants perform best over all the three metrics. In contrast, ours (plain) attains worst results, which we attribute to the missing modules of Lie algebraic representation and GLMI. Moreover, GLMI-M , i.e. GLMI with MLP implementation, works best in generating *Walking* motions, while GLMI-R takes the lead in

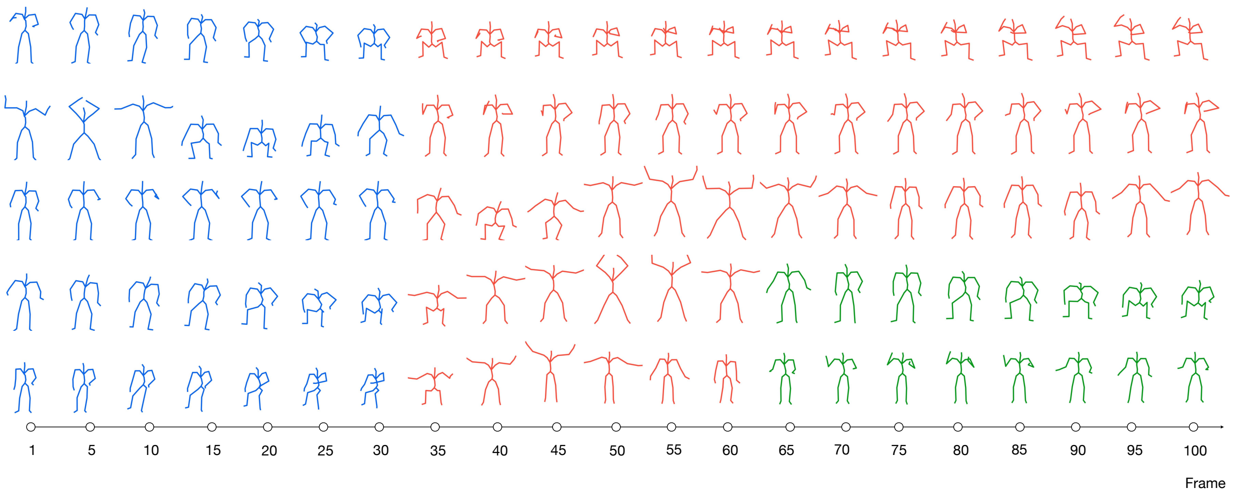


Figure 3.13: Action transition examples. Every 5th frame is shown. The top three rows show transition between two actions. from top to bottom, they are *sit-drink*, *jump up-lift dumbbell*, *lift dumbbell-jump up*, respectively. The bottom two rows display transition of three actions, which are (from top to bottom) *sit-jump up-sit* and *sit-jump up-lift dumbbell*, respectively.

Jump Forward.

Interpolation in Latent Space

Generative models could be regarded as a function mapping between points in a latent space and those in the real data space. Meanwhile, similar to the concept of well-posed problems, a well-learned generative model is expected to behave smoothly from a small perturbation in the latent space. In other words, when we perform interpolations between two distinct latent codes, their generated motions are supposed to transit smoothly. It is thus of interest to examine how interpolations in the latent space would change the motion generation behaviors of our action2motion. It also demonstrates the model capability in producing non-existent samples.

The task is a bit complicated in our situation, as our model generates motion sequences instead of single images. Alternatively, we use the first poses as anchors to perform interpolation between two motions. Specifically, the first poses of two pose sequences are selected. Then, a series of points can be created on the linear path between the latent vectors (i.e. noise vectors) of these two poses. After that, these points are input as initial latent vectors into our model to kick-start the

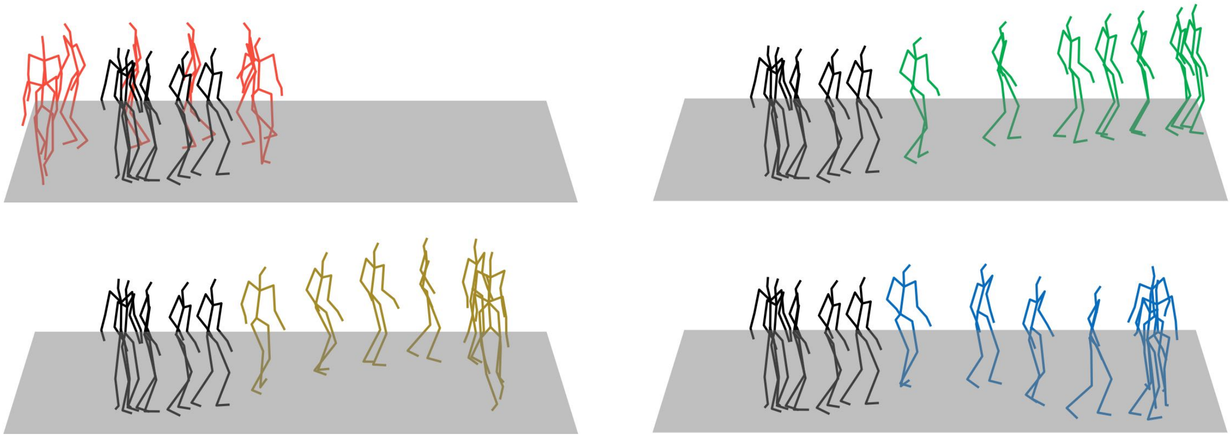


Figure 3.14: Examples of motion outpainting of *Walking*. Provided several initial poses (in black), our method completes the rest motion sequence with multiple plausible outcomes.

generation of rest poses.

Fig. 3.12 considers *lift dumbbell* action. Here two pose sequences are deliberately selected from motions generated by action2motion (GLMI-M), where the first poses of the two sequences are a person lifting with the left (and the right) hand, respectively. We have the following observations. 1) As demonstrated in the first column, transition from the *left hand* pose to the *right hand* pose is realistic at the first poses, by gradually putting one hand down and lifting another hand up. 2) From each of these initial interpolated poses, a visually natural motion sequence is generated. 3) Interestingly the interpolation leads to the generation of a novel motion, *lift dumbbell with both hands*.

Action Transition

To showcase the flexibility of our motion synthesis process, *action transition* is explored by switching the action categories during sequence generation. Exemplar results are presented in Fig. 3.13. To our surprise, our action2motion model is able to produce unseen motions through action transition. In the first row of Fig. 3.13, after switching from *sit down* to *drink*, the character starts to open the bottle and drink with a sitting pose. However, all drinking motions in our training set are performed in *standing* poses. As shown in these examples, the resulting motion sequences are rather realistic and with natural transitions which is well maintained in transitions of not only two actions, but also three actions. This experiment clearly demonstrates the capacity of our approach in synthesizing unseen motions that goes beyond merely memorizing training examples.

Motion Outpainting

Our method could also serve as a motion outpainting tool: provided the initial few poses, apply our method to complete the rest of the motion sequence. This is realized by simply fixing the beginning poses, and generating the rest. Executing multiple independent runs usually creates distinct yet plausible outcomes. Fig. 3.14 illustrates such an example. Here black poses denote the fixed initial poses of *Walk*. This is completed by our model with visually plausible walking motions of distinct velocities and directions. This also suggests the necessity of modeling motion forecasting and generation in a non-deterministic manner.

3.3.2 Step2: Motion2Video

A comprehensive set of experiments are conducted to systematically evaluate the performance of our motion2video approach. Side-by-side evaluations are performed in terms of reconstructing 3D human shapes & textures from single images in Sec. 3.3.2, and animation in Sec. 3.3.2. Sec. 3.3.3 also provides a holistic evaluation of our full pipeline, action2video.

Datasets. To showcase that our pipeline could work with wide range of applications, input images from myriad sources are considered in our experiments, as displayed in Fig. 3.7. They include images from the BUFF dataset [213], People Snapshot dataset [214], as well internet images, computer-generated (CG) images [2], and our in-house captured images. BUFF dataset provides 26 4D human sequences with different cloth styles and performing different actions. We then render 2D images from these human shapes. People Snapshot dataset contains 12 subjects and 24 video sequences with different backgrounds.

[2] https://renderpeople.com/3D-people/

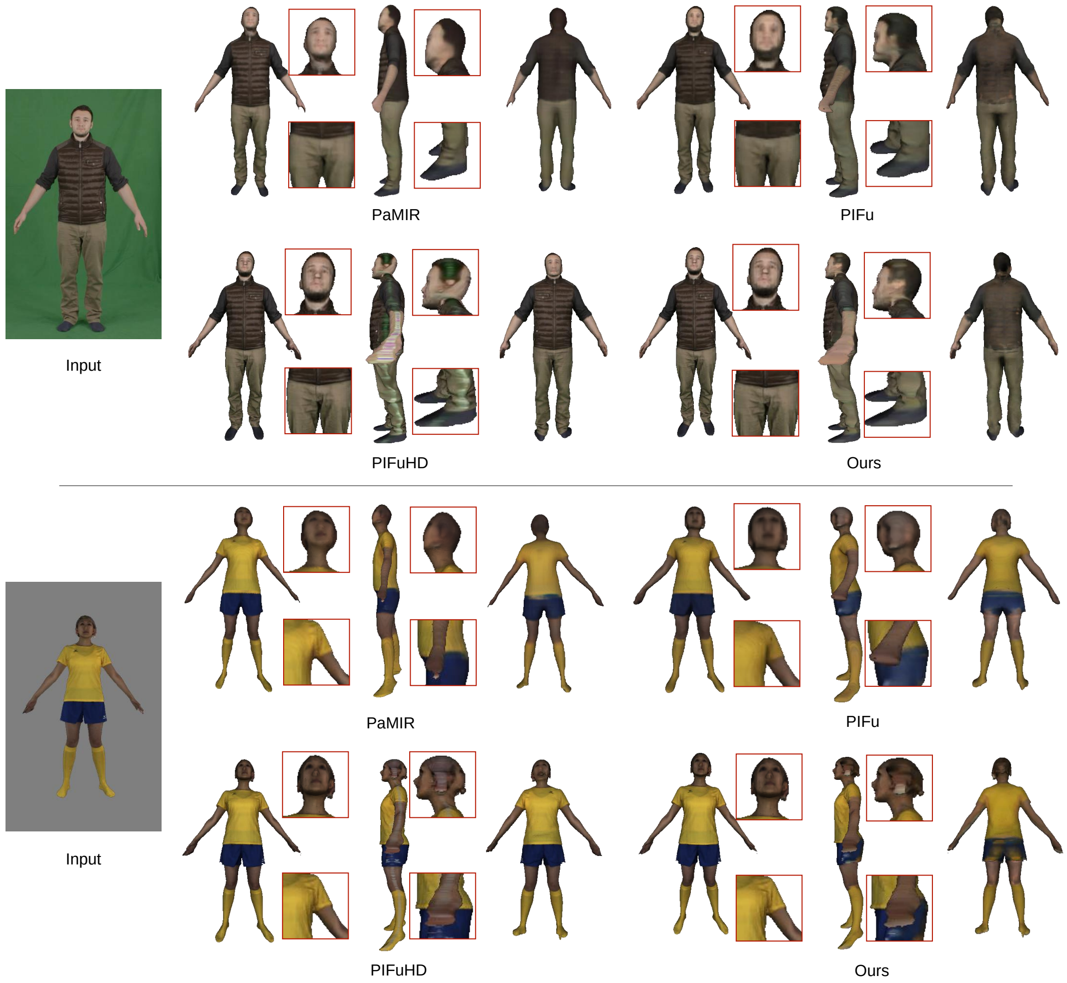


Figure 3.15: A qualitatively comparison of reconstructing 3D human shape & texture from single image, where the top image is from People Snapshot dataset [214] and the bottom one from BUFF dataset [213].

To extract 3D shape from single image, λ_{nn} and 10 neighbors are used in Eq. (3.12) for occluded region. The values of λ_j and λ_r in Eq. (3.13) are set to 2.0 and 0.2, respectively.

3D Shape and Texture Reconstruction

Here we focus on the evaluation of reconstructing 3D human shape & texture from single images, where the respective part of our approach is compared side-by-side with the state-of-the-arts, namely PaMIR [202], PIFu [200] and PIFuHD [201]. PaMIR [202] combines parametric SMPL body model with deep implicit function for robust 3D shape reconstruction. In our comparison, 30

Method	Average Rank↓
PIFuHD [201]	3.60
PIFu [200]	2.36
PaMIR [202]	2.26
Ours	1.77

Table 3.7: Quantitative comparison of reconstructing 3D human shape & texture from single images. The numbers are averaged user preference ranks, with ↓ meaning the numbers are lower the better.

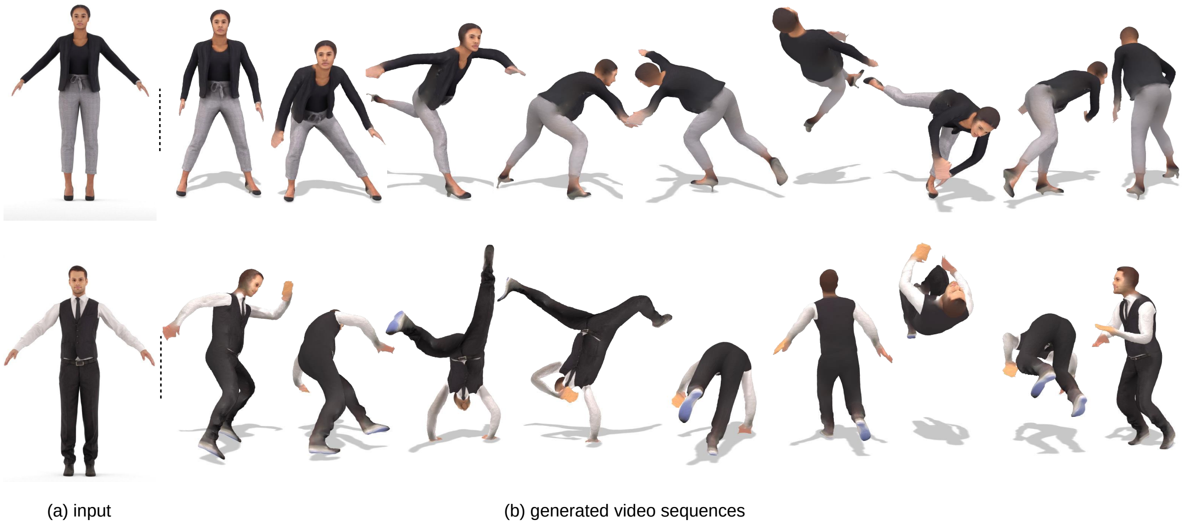


Figure 3.16: Two animation results of our method. Given single images of frontal view of individuals shown on the left, their 3D shapes are reconstructed, 2D videos are obtained, using prescribed off-the shelf motion sequences. The videos produced by our method are visually plausible.

images are obtained from a wide variety of sources, including the BUFF dataset [213], the People Snapshot dataset [214], internet images, CG image, and our in-house captured images. Following the network architectures, the input resolution of PaMIR and PIFU is 512×512, whereas the input image resolution is 1024×1024 for PIFuHD and our approach.

Exemplar results of reconstructed textured shapes from single input images are shown in Fig. 3.15. The shapes and textures extracted by PaMIR and PIFu commonly lack details, and are oftentimes inaccurate. For example, the 3D shape of lady produced by PaMIR is overly slim, together with an smooth face that lack geometric details which is noticeable especially from side-views.

PIFuHD is capable of recovering 3D shapes with better facial geometry and in high-resolution, yet the texture is often visually unpleasantly wrong, especially when viewing from the back. In contrast, our method maintains a delicate balance of shape and texture, thus stands at a better position in facilitating the follow-up animation and realistic rendering processes in our pipeline.

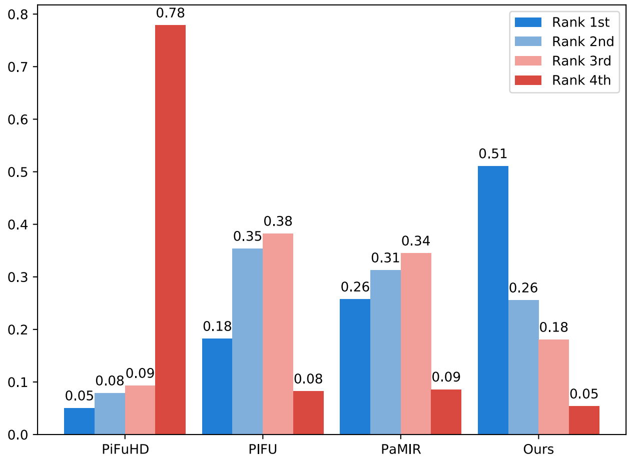


Figure 3.17: User preference distributions of reconstructing 3D human shape & texture from single images.

For quantitative evaluation, user study is further conducted to measure the perceptual quality of the comparison methods. For each input image, 20 Amazon mechanical turk Workers are enrolled to rank their preferences over the shapes reconstructed by their corresponding comparison methods. Table 3.7 displays the average rank of each method, with more detailed rank distributions presented in Fig. 3.17. Our method clearly stands out with the most appreciations from users, where almost half (i.e. 51%) results are ranked the first. By contract, PIFuHD is the least preferred one, of which 78% results are placed as least favorable. In-between are PIFu with the second lowest average rank, and PaMIR that receives considerable more positive feedback compared to PIFuHD.

Motion2Video Animation

In Fig. 3.16, We present two single image animation showcases using our method. 3D shape and texture are predicted from input images, which are driven by two challenging motions, cartwheel, from Adobe Mixamo [3]. As shown, our method could obtain accurate shape and texture predictions from all views, as well as plausible animations with provided motions.

In what follows, we elaborate the comparisons between our method and other three state-of-

[3] www.mixamo.com

Preference	Percentage
Ours Over [8]	0.843
Ours Over [9]	0.593
Ours Over [10]	0.703

Table 3.8: Crowd-sourced subjective assessment to compare the videos animated with the same image and motion, produced by **Ours**, **Liquid Warping GAN** [8], **ARCH** [9] and **[10]**.

the-art image animation methods [8, 9, 10]. For quantitative evaluation, we conduct user study on Mechanical Turk which pairs the videos animated with the same image and motion from our and comparison method, and request the workers to determine which one that is ”more realistic”. For each animation, 50 workers with Hit approval rate higher than 97% are enrolled for perceptual assessment.

Comparison with Liquid Warping GAN [8]. Liquid Warping GAN [8] is a learning based motion transfer method in pseudo-3D space, where 3D SMPL model estimated from reference video frames are used to re-pose the person in source image. Fig. 3.18 presents the animated videos by our method (bottom) and Liquid Warping GAN (top), when feeding with the same input image and motion. While successfully modeling the motion dynamics, the individual images obtained by Liquid Warping GAN are very blurring such that the characteristic personal landmarks of face or T-shirt logo are nearly unrecognizable. In contrast, the animation results of our method are of high-resolution and high quality.

A user study is performed for quantitative evaluation, based on 22 animations from Liquid Warping GAN and our method covering a variety of input images and motion sequences, including composed of 9 Mixamo motions and 13 motions generated by our action2motion step.

As shown in Table 3.8, 84.3% of our animations are preferred by users.

Comparison with ARCH [9]. ARCH [9] uses a semantic deformation field to produce 3D rigged full-body human avatars from a single image, which is already animatable. However, the implementation and pre-trained model of ARCH has not been released yet. We managed to obtain 3 animated 3D model sequences from the authors with our provided images and Mixamo motions. We render video frames of these 3D model sequences in Unity3D with the same environment setting

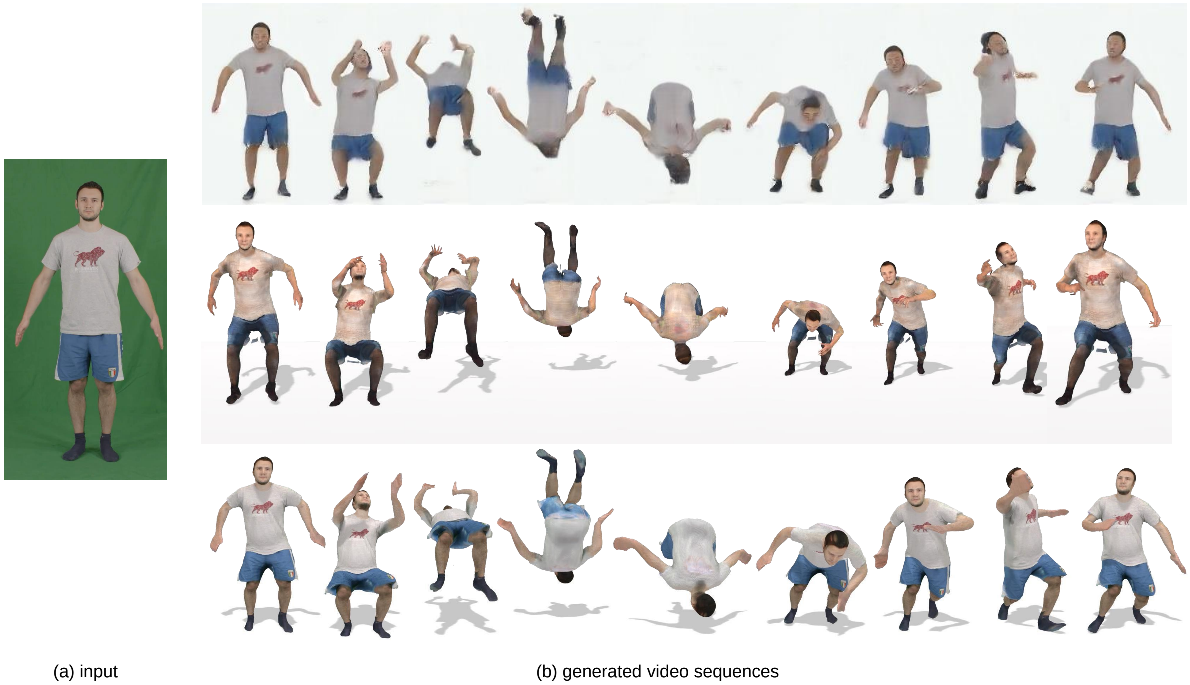


Figure 3.18: Comparing our method (bottom) with Liquid Warping GAN [8] (top) and ARCH [9] (middle), animated using the same input image and motion sequence. Results are displayed by pairing the corresponding video frames.

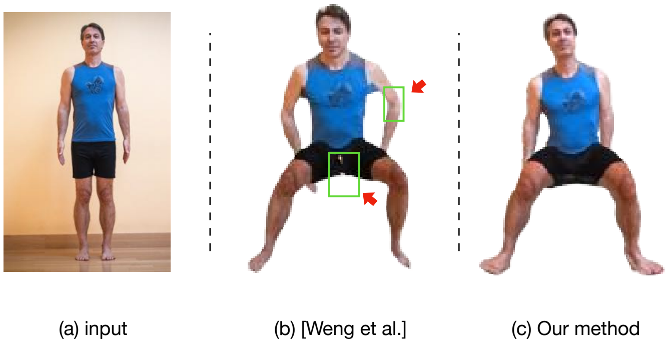


Figure 3.20: Comparing our method with [10] by animating sitting motions.

(e.g. light, camera) as ours. Fig. 3.18 presents a visual comparison between ARCH's result (middle) and our result (bottom). Though ARCH shows capability of generating reasonable rendering, the person appearance is yet to be realistic. For example, the pants comes with several blue debris; the two feet of the man are in wrong color (black); and the texture of T-shirt is overly bright. A user study is again conducted regarding the 3 animations from ARCH and our method. As given in Table 3.8, our method earns more preference (i.e. 59.3%) from users. Please refer to the supplementary video for more visual comparisons.

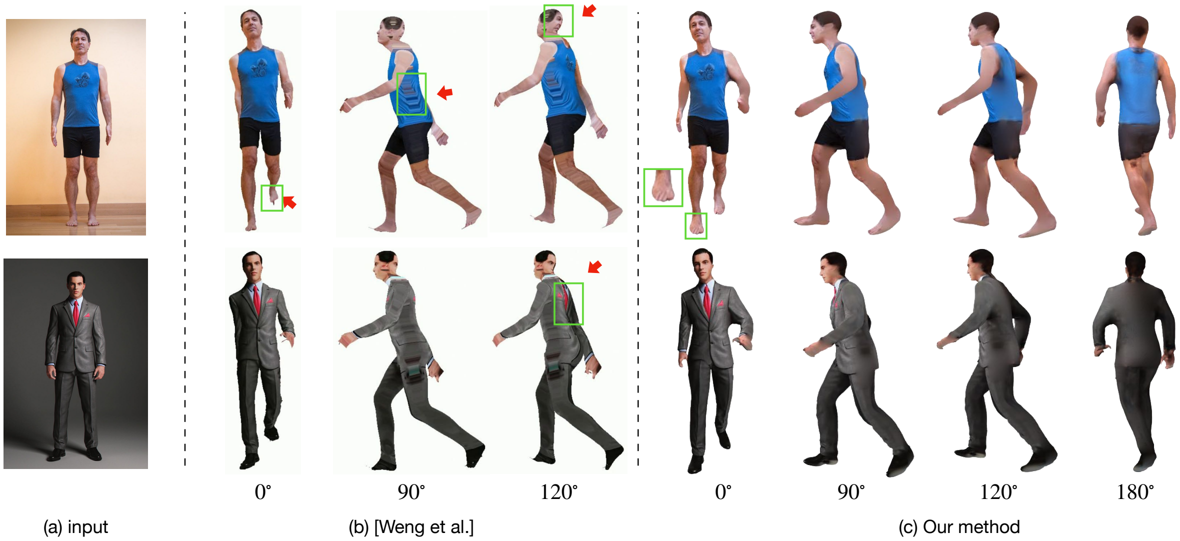


Figure 3.19: Comparing our action2video with [10] by animating walking motions. For each given image on the left, we show the results of [10] (middle column) and ours (right column) from different views. [10] fail to build an intact 3D texture model (e.g. incomplete feet), and the appearance of unseen part is distorted. Our method could generate plausible animation from all angles.

Comparison with [10]. the work of [10] is also closely related to part of our motion2video step, where a 3D character is extracted out of a single image and is further animated to form videos. Their implementation is unfortunately not publicly available, instead we obtain from the authors of [10] two animated action sequences (i.e. sit and walk) from the two input images provided by us. Note that the motions involved in [10] are *real* MoCap motion sequences, while our motions are *generated* by ourselves. For an easy side-by-side visual comparison, we hand pick two of our generated motions that resemble the animations used [10]. The *walk* and *sit* visual results are displayed and compared in Figs. 3.19 and 3.20, respectively. When viewing from frontal view, the results of [10] possess incomplete and distorted errors including the incomplete feet (Fig. 3.19(b)), over-slim arms, and torn pants (Fig. 3.20(b)), as highlighted by red arrows. These artifacts come from the fact that the textures are directly copied and pasted from the 2D input image, which is inadequate to maintain intact appearance in 3D geometry. In comparison, our results are noticeably better at preserving detailed structure and appearance, e.g. around the feet.

When inspecting from the side and back views of the extracted 3D characters that are not directly visible from the input image view, the textured results of [10] are simply mirrored from the frontal region, as shown in the back side of head and torso - the visual results are thus significantly deteriorated to being funny. In contrast, our results preserve reasonable 3D shape and consistent appearance across multiple views including the frontal view. Moreover, a similar user study is conducted among the two set of generated videos. As in Table 3.8, our method is 70.3% more preferred over [10].

3.3.3 The Full Action2Video Pipeline

This section is devoted to the examination of our full action2video pipeline. We start by comparing with state-of-the-art 2D-based human video generation results. Further experiments also demonstrate the capacity of our action2video approach in accommodating input images from different sources.

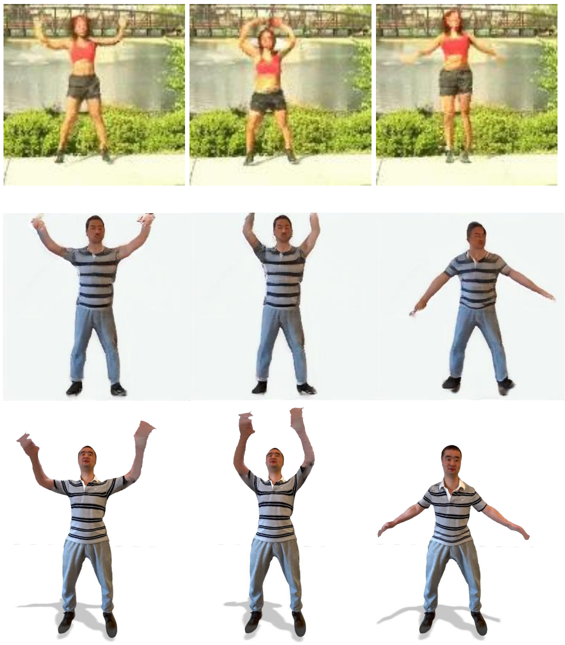


Figure 3.21: Visual comparison of three methods: (top) a state-of-the-art 2D-based method [11], (middle) Liquid Warping GAN [8], and (bottom) ours.

Comparison with existing methods. The work of [11] is state-of-art in generating human motion videos, which is 2D-based and relies on large-scale training set of videos. Fig. 3.21 presents a comparison of their results and ours that share in common similar poses and views. Compared with our results, the frames of [11] is of low resolution (128x128). Moreover, there are visible lack of details of face, hands & clothes, and unrealistic shape deformations, which we attribute to their innate 2D based limitations. For example, lengths of legs and arms in [11] of the same lady character vary over time. Moreover, as presented in the middle row of Fig. 3.21, the exemplar video result generated by engaging Liquid Warping GAN based on the same motion generated by our action2motion step,

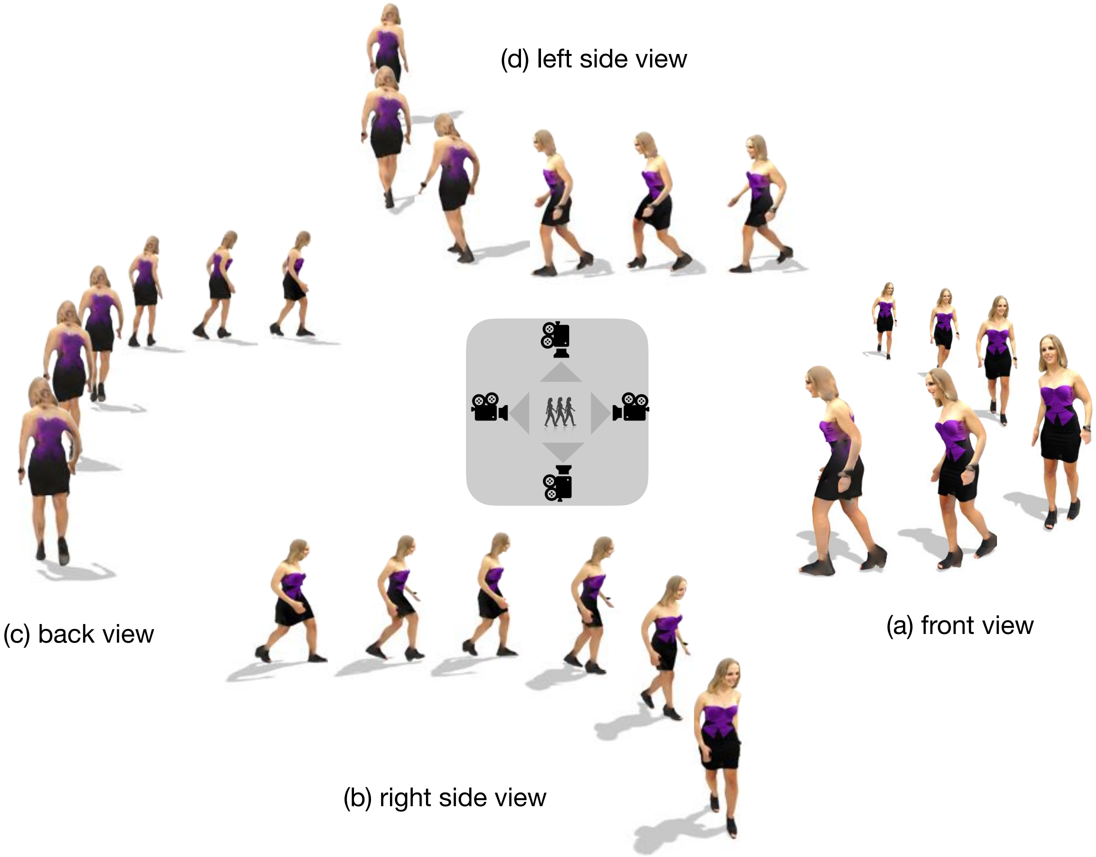


Figure 3.22: An generated walking video from the following views: (a) front, (b) right-side, (c) back, and (d) left-side.

where edges and facial details are very foggy and fuzzy, when comparing to our results shown at the bottom row.

Diverse input image sources. This experiment is to evaluate the flexibility of our action2video pipeline in accommodating input images from varied sources. Fig. 3.23 presents our action2video results based on BUFF images (e.g. 1st row), People Snapshot images (e.g. 2nd row), Internet images (e.g. 4th row), these captured by our mobile-phone (e.g. 3rd row) as the input images. Overall our approach is able to adapt to these different applications, and to produce videos of visually pleasing quality.

Multiple camera views. Fig. 3.22 displays an exemplar video sequence generated by our approach, that is inspected from four different views. It demonstrates 1) our extracted 3D shape and clothing texture are reasonably realistic when examined in different rendered views, and 2) compared to the popular 2D-based methods, our generated videos are consistent among distinct views.

3.4 Conclusion and Discussion

Conclusion. We propose an action2video approach to tackle the exciting and challenging problem of generating natural and diverse 3D motions & videos of human actions. This is accomplished in this chapter by a 2-step pipeline: action2motion focuses on generating 3D human motions, which are then turned into videos by motion2video. Empirical studies demonstrate the effectiveness of our approach.

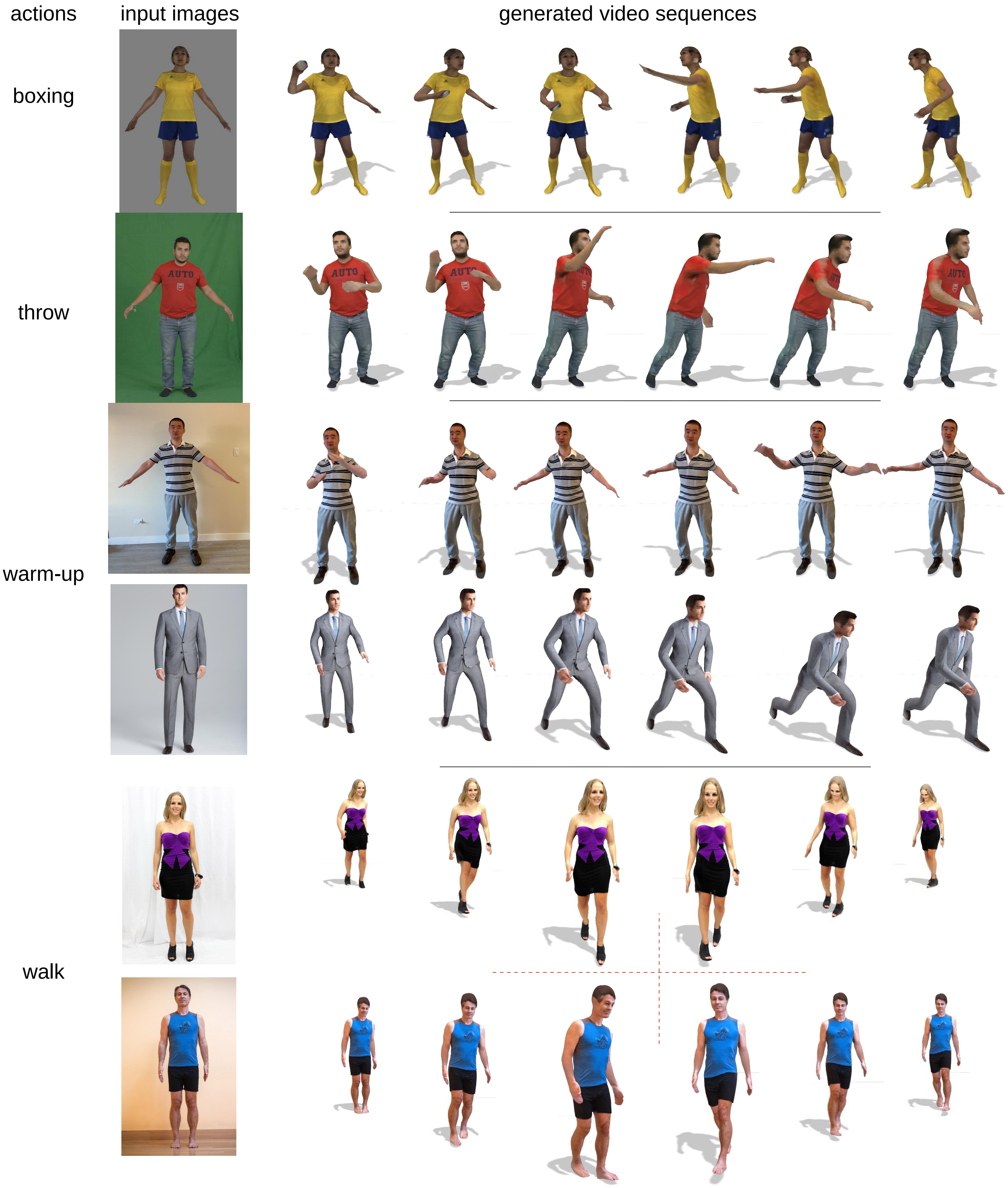


Figure 3.23: Exemplar videos produced by our action2video pipeline.

Chapter 4

Text based 3D Human Motion Generation

As illustrated in Fig. 4.1, given an input script, our goal is to generate a diverse set of natural 3D human motion sequences following precisely the text. Automated generation of 3D human motions from text is a challenging problem. The generated motions are expected to be sufficiently *diverse* to explore the text-grounded motion space, and more importantly, *accurately* depicting the content in prescribed text descriptions. Here we tackle this problem with a two-stage approach: text2length sampling and text2motion generation. Text2length involves sampling from the learned distribution function of motion lengths conditioned on the input text. This is followed by our text2motion module using a temporal variational autoencoder to synthesize a diverse set of human motions of the sampled lengths. Moreover, a large-scale dataset of scripted 3D Human motions, HumanML3D, is constructed, consisting of 14,616 motion clips and 44,970 text descriptions. This chapter has been published as [3]. The related data, pretrained model, and codes are publicly available:https://ericguo5513.github.io/text-to-motion/.

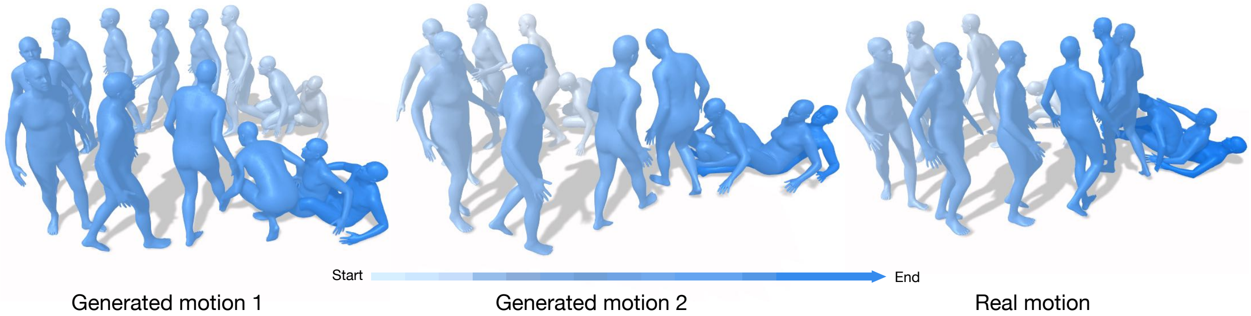


Figure 4.1: Taken as input the text description, *"the figure rises from a lying position and walks in a counterclockwise circle, and then lays back down the ground"*, our approach generates multiple distinct 3D human motions (e.g. the left and middle panels) that are faithful to the prescribed textual content. The real motion is also presented at the right panel for reference.

4.1 Introduction

Given a short textual description of a character's movement as for example, an excerpt from a novel or a script, we are capable of visualizing the motions in our minds or even in drawings. The question is, how to automate this process by a machine, or in paraphrase, to generate realistic 3D human motions from text? This is the problem we tackle with in this chapter. As illustrated in Fig. 4.1, given the input feed of "*the figure rises from a lying position and walks in a counterclockwise circle, and then lays back down the ground*", our goal is to generate a diverse set of plausible 3D human motion dynamics following precisely the action types, directions, speeds, timing and styles as prescribed by the text.This automation process could bring a broad range of application impacts in AR/VR content creation, gaming, robotics, and human-machine interaction, to name a few.

Meanwhile, existing efforts in generating 3D human motions from descriptions [12, 37, 38, 39, 219] are sporadic and the results are far from being satisfactory. Several common shortfalls are observed: the input text is usually one short sentence; the task is invariably formulated as deterministic sequence-to-sequence generation, with the synthesized motions tending to be stationary and lifeless; moreover, the generated motions are restricted to have the same length; finally, the sole dataset relied on by existing methods, KIT Motion-Language (KIT-ML) [43], consists of only 3,010 motion sequences focusing on locomotion actions. In particular, there are three inherit challenges yet to be addressed. First, motions generated from text by the same model are expected to possess variable lengths. Second, there are usually multiple ways for a character to behave following the same textual description. Third, from natural language perspective, the input descriptions may have a wide range of forms, from being short & simple to very long & complex.

To address the aforementioned shortfalls and challenges, we propose a two-stage pipeline consisting of text2length sampling and text2motion generation. Text2length estimates the distribution function of visual motion length grounded on the input text. The role of text2motion is to generate distinct 3D motions from the input text and the sampled motion length; this is realized by engaging the temporal variational autoencoder (VAE) framework in its triplet form of prior, posterior, and generator networks; moreover, motion snippet code is introduced as the internal representation in VAE code and throughout our pipeline to characterize the temporal motion semantics, with its role empirically examined in later ablation studies. Finally, a dedicated dataset (HumanML3D) is constructed, consisting of 44,970 textual descriptions for 14,616 3D human motions. It covers a wide range of action types including but not limited to locomotive actions. Empirical evaluations on both HumanML3D and KIT-ML datasets demonstrate the superior performance of our approach over existing methods.

Our key contributions are summarized as follows. First, this work is to our knowledge the first in stochastically generating 3D motions from text, capable of generating diverse 3D human motions of variable lengths that are realistic-looking and faithful to the text input. Second, our approach is flexible to work with input text ranging from simple to complex forms. This is made possible by the text2length & text2motion modules, and the proposed motion snippet codes that are to be detailed in later sections. Finally, a large-scale human motion dataset is constructed. It contains a wide range of actions, with each motion sequence paired with three textual descriptions.

4.2 Our Approach

From a text description of M words, $X = (x_1, ..., x_M)$, our goal is to generate a 3D pose sequence, $P = (\mathbf{p}_1, ..., \mathbf{p}_{T'})$, with its length T' determined at test time. As shown in Fig. 4.2, we start by a preprocessing step to train a motion autoencoder. This is followed by settling a reasonable motion

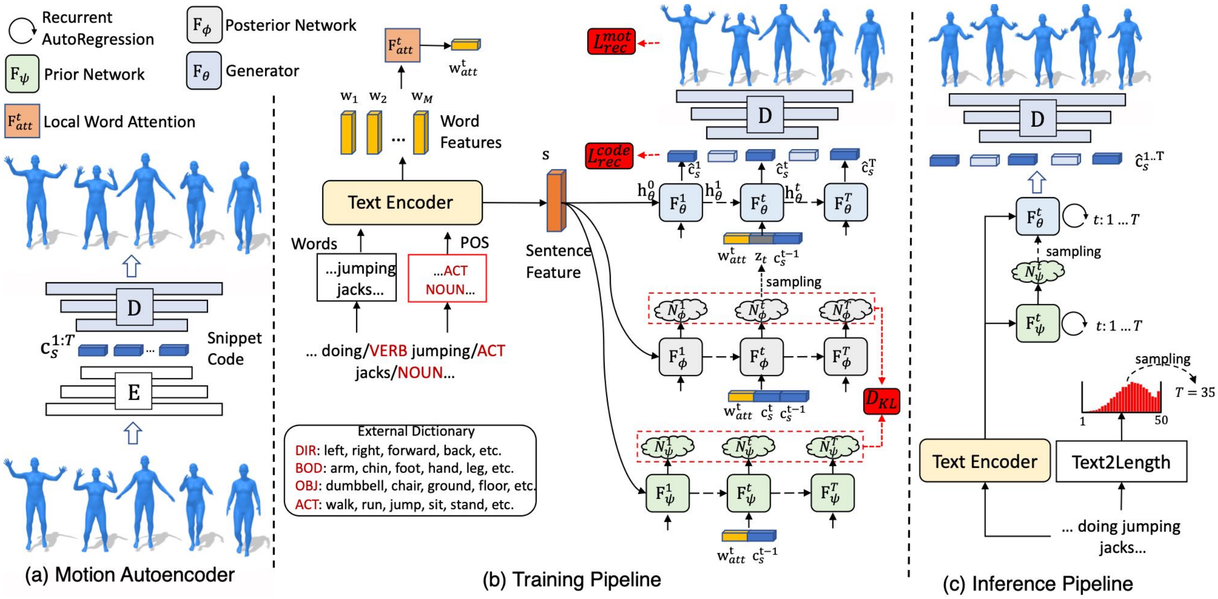

Recurrent AutoRegression
F_ϕ Posterior Network
F_ψ Prior Network
F_θ Generator
F^t_{att} Local Word Attention
D
$c_s^{1:T}$
Snippet Code
E
(a) Motion Autoencoder
F^t_{att}
w^t_{att}
w_1 w_2 w_M
Word Features
Text Encoder
Words
POS
...jumping jacks...
...ACT NOUN...
... doing/VERB jumping/ACT jacks/NOUN...
External Dictionary
DIR: left, right, forward, back, etc.
BOD: arm, chin, foot, hand, leg, etc.
OBJ: dumbbell, chair, ground, floor, etc.
ACT: walk, run, jump, sit, stand, etc.
s
Sentence Feature
$\mathcal{L}^{mot}_{rec}$
$\mathcal{L}^{code}_{rec}$
D
$\hat{c}^1_s$ $\hat{c}^t_s$ $\hat{c}^T_s$
h^0_θ h^1_θ h_θ
F^1_θ F^t_θ F^T_θ
w^t_{att} z_t c_s^{t-1}
sampling
N^1_ϕ N^t_ϕ N^T_ϕ
F^1_ϕ F^t_ϕ F^T_ϕ
w^t_{att} c^t_s c_s^{t-1}
D_{KL}
N^1_ψ N^t_ψ N^T_ψ
F^1_ψ F^t_ψ F^T_ψ
w^t_{att} c_s^{t-1}
(b) Training Pipeline
D
$\hat{c}_s^{1..T}$
F^t_θ $t: 1 \dots T$
sampling
N^t_ψ
F^t_ψ $t: 1 \dots T$
sampling
$T = 35$
1
50
Text Encoder
Text2Length
... doing jumping jacks...
(c) Inference Pipeline

Figure 4.2: **Approach overview.** (a) As a preprocessing step, a dedicated motion autoencoder is trained on our training motion data to encode a motion sequence into a stream of motion snippet codes, which then could be decoded back into motions. (b) Our training pipeline. Through text encoder, the attentive word features ($\mathbf{w}_{att}$) are used by VAE networks as illustrated in Fig. 4.3. The triplet structure of temporal VAE involving the prior, posterior, and generator networks is employed to process the motion snippet codes ($\mathbf{c}_s$) and the reconstructed ones ($\hat{\mathbf{c}}_s$). This leads to the loss terms evaluating the reconstructed pose sequence ($\mathcal{L}^{mot}_{rec}$) and the reconstructed code sequence ($\mathcal{L}^{code}_{rec}$), respectively. Due to lack of space, some key ingredients are deferred to be presented in Fig. 4.3. (c) Our inference pipeline. From the input text, text2length module is activated to sample an intended motion length. Text features extracted through the text encoder are then fed to the prior network, yielding a prior distribution. Generator samples latent vectors from the prior distribution and produces a series of motion snippet codes ($\hat{\mathbf{c}}_s$). The pose sequence is finally obtained by decoding the snippet codes from the motion decoder pre-trained in (a).

length from text (Sec. 4.2.2), and subsequently synthesizing motions conditioned on the input text and the sampled motion length (Sec. 4.2.3), by introducing an internal motion representation – motion snippet codes (Sec. 4.2.1).

4.2.1 Motion Autoencoder

As the preprocessing step described in Fig. 4.2(a), an encoder E transforms the pose sequence $P = (\mathbf{p}_1, ..., \mathbf{p}_{T'})$ to a motion snippet code sequence, $C_s = \left(\mathbf{c}_s^1, ..., \mathbf{c}_s^T\right)$, achieved by applying 1-D convolutions over temporal line; $\hat{P}$ is then reconstructed with a deconvolutional decoder, D.

Mathematically, this process is formulated as

$$C_s = \mathrm{E}(P), \quad \hat{P} = \mathrm{D}(C_s). \tag{4.1}$$

To avoid foot sliding, our decoder D additionally predict foot contacts at each frame which are not given to the encoder E. The foot contact label is a 4D binary vector that indicates the contact status of heels and toes, obtained by thresholding the velocity of foot joints. During training, this foot contact vector is concatenated with other features, forming the pose vector $\mathbf{p}$. It is also necessary to constrain the snippet code values and the differences of consecutive codes to encourage sparsity and temporal smoothness. The final objective function becomes

$$\mathcal{L}_{E,D} = \sum_{t'} \|\hat{\mathbf{p}}_{t'} - \mathbf{p}_{t'}\|_1 + \lambda_{spr} \sum_{t} \|\mathbf{c}_s^t\|_1 + \lambda_{smt} \sum_{t} \|\mathbf{c}_s^t - \mathbf{c}_s^{t-1}\|_1. \tag{4.2}$$

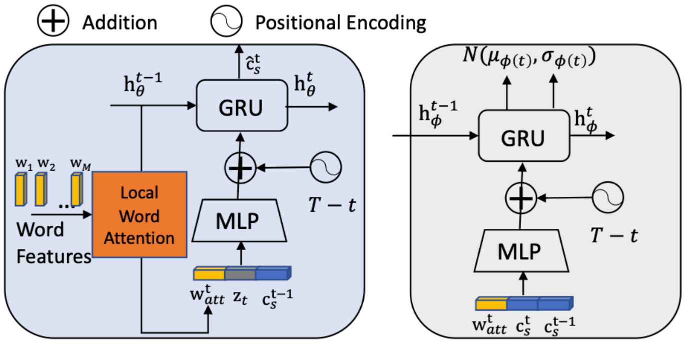
The autoencoder consists of two-layer convolutions with filter size of 4 and stride 2. As a result, a motion snippet code $\mathbf{c}_s^t$ has a 8-frame receptive field, amounting to around 0.5 second for 20 frame-per-second (fps) pose streaming; it also leads to a more compact internal code sequence with $T = \frac{T'}{4}$. Compared to individual poses, snippet code captures temporal semantic information that is crucial in smooth and faithful motion generation.

4.2.2 Text2length Sampling

As shown in Fig. 4.2(c), the purpose of our text2length sampling module is to approximate the probability distribution of discrete motion length T conditioned on text, such that at inference stage, a discrete time length T can be obtained by sampling from this learned distribution function, $p(T|x_1, ..., x_M)$ given an input text. This module thus enables our approach in generating motions of distinct lengths.

This is a typical density estimation problem with many practical options, among them we adopt the neural network scheme of pixelCNN [220]. Since a motion sequence is internally represented in our work as a series of snippet codes, our aim specifically boils to deciding the length of snippet codes. In inference, a text encoder extracts sentence-level features from the input text, which are then fed into an MLP layer with softmax activation, producing a multinomial distribution over discrete length indices $\{1, 2, ..., T_{max}\}$. Here an increment of 1 corresponds to 4 pose frames, and setting $T_{max} = 50$ corresponds to 200 frames, amounting to 10 seconds for a 20 fps video. Its training objective is defined by the cross entropy loss.

4.2.3 Text2motion Generation



(a) Generator F_θ^t (b) Posterior Network F_ϕ^t

Figure 4.3: Structure of our temporal VAE for text2motion generation: (a) generator F_θ^t, and (b) posterior network F_ϕ^t. Prior network F_ϕ^t has the same architecture as F_ϕ^t except different inputs.

Our text2motion generator contains a text encoder, and a temporal VAE model consisting a triplet networks of generator F_θ, posterior F_ϕ and prior F_ψ, as in Fig. 4.2(b). The text encoder extracts both the word-level $\mathbf{w}_{1:M}$ and sentence-level $\mathbf{s}$ features from input text; our VAE generates motion snippet codes $\mathbf{c}_s^{1:T}$ one by one with a recurrent architecture: at time t, our posterior network F_ϕ approximates the posterior distribution $q_\phi\left(\mathbf{z}_t | \mathbf{c}_s^{1:t}, \mathbf{c}\right)$ conditioned on partial code sequence $\mathbf{c}_s^{1:t}$ as well as word and sentence features $\mathbf{c} = (\mathbf{w}_{1:M}, \mathbf{s}, ...)$. Instead of relating the posterior distribution to a prior normal distribution $\mathcal{N}(0, I)$ as used by the literature, here it is related to a learned prior distribution $p_\psi(\mathbf{z}_t | \mathbf{c}_s^{1:t-1}, \mathbf{c})$, which is obtained by our prior network F_ψ, based on the previous state $\mathbf{c}_s^{1:t-1}$ and conditions $\mathbf{c}$. Overall, our VAE is trained by maximizing the following variational lower bound,

$$\log p(C_s) \geq \sum_{t=1}^{T} \Big[\mathbb{E}_{q_\phi(\mathbf{z}_t|\mathbf{c}_s^{1:t},\mathbf{c})} \log p_\theta(\mathbf{c}_s^t|\mathbf{c}_s^{1:t-1}, \mathbf{z}_{1:t}, \mathbf{c}) \quad -\lambda_{KL} D_{\mathrm{KL}} \left(q_\phi(\mathbf{z}_t|\mathbf{c}_s^{1:t}, \mathbf{c}) \parallel p_\psi(\mathbf{z}_t|\mathbf{c}_s^{1:t-1}, \mathbf{c}) \right) \Big]. \tag{4.3}$$

The first term is to reduce reconstruction error $\mathcal{L}_{rec}$, while the second term penalizes the KL-divergence $\mathcal{L}_{KL}$ between the posterior and the prior distributions.

Text Encoder. In addition to the word embeddings, we propose to incorporate the part-of-speech (POS) tags of words into text encoder. POS tag explicitly indicates the word categories, thus facilitates the localization of important words in a sentence. Furthermore, as in Fig. 4.2(b), an external dictionary is manually constructed to collect motion-related words and categorize them into four types: direction, body part, object and action. These one-hot word tags are fed into an embedding layer and added to word embedding vectors. Our text encoder is realized in the form of bi-directional GRUs, which take these embedding vectors as inputs and produces both sentence feature s and word features $\mathrm{w}_{1:M}$.The former provides global contextual information and is used to initialize the hidden units of VAE; the latter serves as partial inputs at each time step in the form of local word attention, to be discussed next.

Local Word Attention (F_{att}). Attentions assigned to each word may vary in the process of predicting motions from text. This is addressed by our local word attention unit F_{att} that engages and interacts word features $\mathrm{w}_{1:M}$ with motion context memory $\mathbf{h}_\theta$ (i.e. generator hidden unit) as depicted in Fig. 4.3. The process of local word attention can be described as

$$\begin{aligned} \mathbf{Q} = \mathbf{h}_\theta^{t-1}\mathbf{W}^Q, \mathbf{K} = \mathbf{w}_{1:M}\mathbf{W}^K, \mathbf{V} = \mathbf{w}_{1:M}\mathbf{W}^V, \\ \mathbf{w}_{att}^t = \mathrm{softmax}\left(\frac{\mathbf{QK}^T}{\sqrt{d_{att}}}\right)\mathbf{V}, \end{aligned} \tag{4.4}$$

where $\mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d_w \times d_{att}}$ and $\mathbf{W}^Q \in \mathbb{R}^{d_h \times d_{att}}$ are trainable weights, with d_h, d_w and d_{att} the number of channels in generator hidden unit $\mathbf{h}_\theta^{t-1}$, word features $\mathrm{w}_{1:M}$ and attentive layer respectively. $\mathbf{w}_{att}^t$ is the multi-modal attentive vector obtained as time t.

Time-to-Arrival Positional Encoding. In generating motions of variable lengths, it is important to aware *where we are* and *how far to go*. This motivates us to encode the time-to-arrival information $T - t$ with positional encoding at each time step, as in Fig. 4.3. It is formulated as

$$\begin{aligned} \mathrm{PE}_{T-t,2i} &= \sin\left(\frac{T-t}{10000^{\frac{2i}{d}}}\right), \\ \mathrm{PE}_{T-t,2i+1} &= \cos\left(\frac{T-t}{10000^{\frac{2i}{d}}}\right), \end{aligned} \tag{4.5}$$

where the second subscript of the vector PE denotes the dimension index; d is the dimensionality of input embedding.

Architecture of Temporal VAE. Fig. 4.2(b) illustrates the overall architecture of our temporal VAE for text2motion generation; this is followed by Fig. 4.3, which brings a zoom-in view of the generator and posterior network structures.

At time t, word features first interacts with generator memory unit h_θ^{t-1} to yield the attentive vector $\mathbf{w}_{att}^t$. Now concatenating the present and previous snippet codes ($\mathbf{c}_s^t$ & $\mathbf{c}_s^{t-1}$), and the attentive vector $\mathbf{w}_{att}^t$ to form an input vector, which is fed into a multi-layer perceptron (MLP); its output is summed with time-to-arrival positional encoding PE_{T-t}, which then passes through a GRU layer to produce the posterior distribution $\mathcal{N}(\mu_\phi(t), \sigma_\phi(t))$. Yielding prior distribution $\mathcal{N}(\mu_\psi(t), \sigma_\psi(t))$ follows the same process, except not taking $\mathbf{c}_s^t$ as input. In training, the generator learns to reconstruct current snippet code $\hat{\mathbf{c}}_s^t$ from the input of $\mathbf{c}_s^{t-1}$, $\mathbf{w}_{att}^t$, and a noise vector $\mathbf{z}_t$ sampled from the posterior distribution. In testing, as the $\mathbf{c}_s^t$ from real data is unavailable, $\mathbf{z}_t$ is instead sampled from the estimated prior distribution $p_\psi(\mathbf{z}_t|\mathbf{c}_s^{1:t-1}, \mathbf{c})$ (Fig. 4.2(c)). Finally, the output pose sequence $\hat{\mathbf{p}}_{1:T'}$ is produced by decoding the internal snippet code sequence $\mathbf{c}_s^{1:T}$ with the pre-trained motion decoder D (Sec. 4.2.1). In text2motion, motion decoder D is fine-tuned with the rest networks.

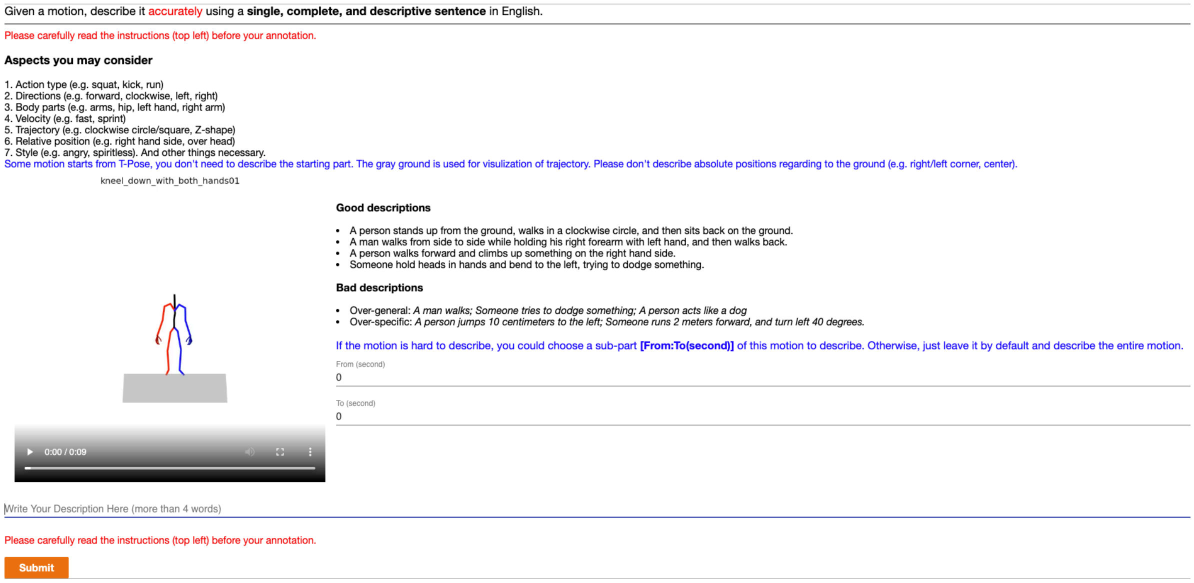
Final Objective. Our final objective function for text2motion generation becomes

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{rec}^{code} + \lambda_{mot}\mathcal{L}_{rec}^{mot} + \lambda_{KL}\mathcal{L}_{KL}, \quad \text{with} \\ \mathcal{L}_{rec}^{code} &= \sum_{t} \|\hat{\mathbf{c}}_s^t - \mathbf{c}_s^t\|_1, \\ \mathcal{L}_{rec}^{mot} &= \sum_{t'} \|\hat{\mathbf{p}}_{t'} - \mathbf{p}_{t'}\|_1, \\ \mathcal{L}_{KL} &= \sum_{t} \mathrm{KL}(\mathcal{N}(\mu_\phi(\mathrm{t}), \sigma_\phi(\mathrm{t}))\|\mathcal{N}(\mu_\psi(\mathrm{t}), \sigma_\psi(\mathrm{t}))). \end{aligned} \tag{4.6}$$

Training Scheme. To address the variable length sequence-to-sequence generation task, our training process utilizes both curriculum learning [206] and scheduled sampling [205] strategies, as follows. Starting from aiming to generate first T_{cur} snippet codes in sequence, we optimize our model on training data that owns snippet code lengths equal or longer than T_{cur}. As long as the reconstruction loss on the validation starts raising, then we move on to the next stage by appending one more snippet code in the target sequence ; the complexity of the task is progressively increased at every stage till the maximum time step T_{max} of prediction is reached (i.e, $T_{cur} = T_{max}$). In addition, to bridge the gap of training and inference for sequence prediction, *teacher forcing* is applied for the entire target snippet code sequence $\mathbf{c}_s^{1:T}$ with probability of p_{tf}, which means the *ground-truth* snippet code is taken as input for the generation at next step. Accordingly, the *generated* snippet code will instead serve as the input with probability $1 - p_{tf}$. As a boundary condition, $\mathbf{c}_s^0$ is a constant vector that encodes mean poses using motion encoder E.

4.3 Our HumanML3D Dataset

Our HumanML3D dataset originates from a amalgamation of motion sequences from the HumanAct12 [1] and AMASS [194] datasets, two large-scale datasets of 3D human motion captures that are publicly accessible. They contains motions from a variety of human actions, such as daily activities (e.g., 'walking', 'jumping'), sports (e.g, 'swimming', 'karate'), acrobatics (e.g, 'cartwheel') and artistry (e.g, 'dancing'). Unfortunately, these datasets come without textual descriptions of



Given a motion, describe it accurately using a **single, complete, and descriptive sentence** in English.

Please carefully read the instructions (top left) before your annotation.

Aspects you may consider

1. Action type (e.g. squat, kick, run)
2. Directions (e.g. forward, clockwise, left, right)
3. Body parts (e.g. arms, hip, left hand, right arm)
4. Velocity (e.g. fast, sprint)
5. Trajectory (e.g. clockwise circle/square, Z-shape)
6. Relative position (e.g. right hand side, over head)
7. Style (e.g. angry, spiritless). And other things necessary.

Some motion starts from T-Pose, you don't need to describe the starting part. The gray ground is used for visulization of trajectory. Please don't describe absolute positions regarding to the ground (e.g. right/left corner, center).

kneel_down_with_both_hands01

Good descriptions

- A person stands up from the ground, walks in a clockwise circle, and then sits back on the ground.
- A man walks from side to side while holding his right forearm with left hand, and then walks back.
- A person walks forward and climbs up something on the right hand side.
- Someone hold heads in hands and bend to the left, trying to dodge something.

Bad descriptions

- Over-general: *A man walks; Someone tries to dodge something; A person acts like a dog*
- Over-specific: *A person jumps 10 centimeters to the left; Someone runs 2 meters forward, and turn left 40 degrees.*

If the motion is hard to describe, you could choose a sub-part **[From:To(second)]** of this motion to describe. Otherwise, just leave it by default and describe the entire motion.

From (second)
0

To (second)
0

Write Your Description Here (more than 4 words)

Please carefully read the instructions (top left) before your annotation.

Figure 4.4: HumanML3D annotation interface on Amazon Mechanical Turk.

the motions.

Several processing steps take place for data normalization, as follows. Motions are scaled to 20 FPS, and those longer than 10 seconds are randomly cropped to 10-second ones; they are then retargeted to a default human skeletal template and properly rotated to face Z+ direction initially. This is followed by a textual annotation process via the Amazon Mechanical Turk (AMT), where native English-speaking turkers with average work approval rating above 92% are hired and asked to describe a motion with at least 5 words. We collect 3 text descriptions for each motion clip from distinct workers. A manual postprocessing step ensues to filter away abnormal textual descriptions.

As a result, our HumanML3D dataset becomes to our knowledge the largest and most diverse collection of scripted human motions, consisting of 14,616 motions and 44,970 descriptions composed by 5,371 distinct words. The total length of motions amounts to 28.59 hours, in which the average motion length is 7.1 seconds. The minimum and maximum duration are 2s and 10s respectively. In terms of the textual descriptions, their average and median lengths are 12 and 10, respectively. A tabular comparison of our HumanML3D versus the only existing motion-text dataset, KIT Motion-Language [43] is presented in Table 4.1.

Fig. 4.4 presents the interface of annotating our HumanML3D dataset on Amazon Mechanical

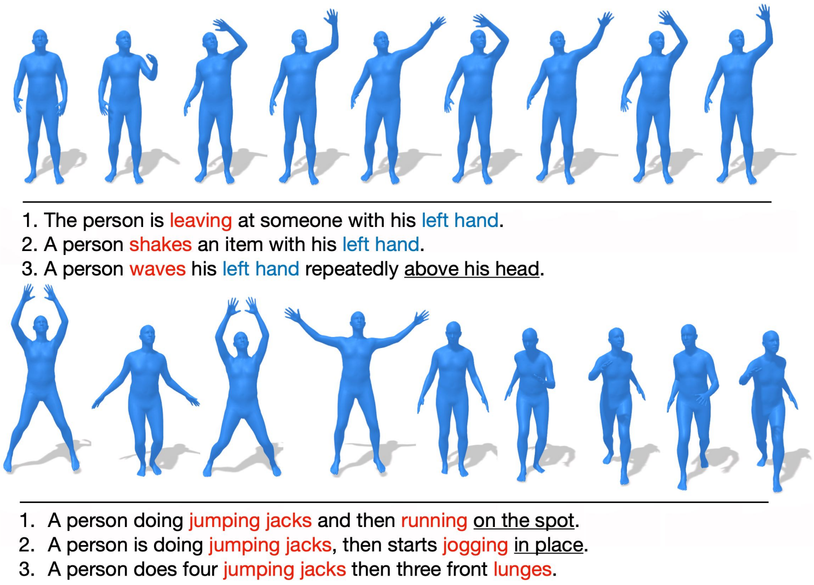

1. The person is leaving at someone with his left hand.
2. A person shakes an item with his left hand.
3. A person waves his left hand repeatedly above his head.
1. A person doing jumping jacks and then running on the spot.
2. A person is doing jumping jacks, then starts jogging in place.
3. A person does four jumping jacks then three front lunges.

Figure 4.5: **Examples in HumanML3D dataset**. Two annotation examples in our HumanML3D dataset are shown. Each motion sequence comes with 3 distinct script descriptions.

Dataset	#Motions	#texts	Duration	Vocab.
HumanML3D	14,616	44,970	28.59h	5,371
KIT-ML[43]	3,911	6,278	10.33h	1,623

Table 4.1: Comparisons of 3D human motion-language datasets.

Turk. Given an animation, users are encouraged to convey the information of *action type*, *direction*, *body parts*, *velocity*, *trajectory*, *relative position* and *style* in text descriptions. If a motion is too complicated to be described, we also provide a channel for users to describe a sub-interval of presented animation, and give the start- and end- time point. Some exemplar good and bad descriptions help workers form a clearer picture. We ask users to avoid **over-general** descriptions (e.g. a man walks) and **over-specific** descriptions (e.g. absolute distance, angles, positions). Unqualified descriptions are manually rejected. Fig. 4.5 presents two scripted motions in our HumanML3D dataset. Each motion is described by 3 distinct worker on Amazon Mechanical Turker, thus resulting in diversified descriptions.

4.4 Experiments

Empirical evaluations are carried on both the in-house HumanML3D and KIT-ML [43] datasets. We augment both datasets by mirroring motions and properly replacing certain keywords in the descriptions (e.g. 'left'→ 'right'). Both datasets are split to training, test and validation sets with 0.8 : 0.15 : 0.05 ratio. In training, all motions are trimmed such that numbers of frames are multiples of 4. We apply the same pose processing steps as in [221].

Pose Representation. A pose $\mathbf{p}$ in our work is defined by a tuple of $(\dot{r}^a, \dot{r}^x, \dot{r}^z, r^y, \mathbf{j}^p, \mathbf{j}^v, \mathbf{j}^r, \mathbf{c}^f)$, where $\dot{r}^a \in \mathbb{R}$ is root angular velocity along Y-axis; $(\dot{r}^x, \dot{r}^z \in \mathbb{R})$ are root linear velocities on XZ-plane; $r^y \in \mathbb{R}$ is root height; $\mathbf{j}^p \in \mathbb{R}^{3j}, \mathbf{j}^v \in \mathbb{R}^{3j}$ and $\mathbf{j}^r \in \mathbb{R}^{6j}$ are the local joints positions, velocities and rotations in root space, with j denoting the number of joints; $\mathbf{c}^f \in \mathbb{R}^4$ is binary features obtained by thresholding the heel and toe joint velocities to emphasize the foot ground contacts. In particular, the 6D continuous rotation representation of [222] is adopted. Motions in HumanML3D dataset follows the skeleton structure of SMPL [47] with 22 joints. Poses have 21 joints in KIT-ML.

Implementation Details. Our framework is implemented by PyTorch. Dimensions of pose vector for HumanML3D and KIT-ML dataset are 263 and 251 respectively. The motion snippet codes $\mathbf{c}_s$ are 512-dimensional vectors. Word features are 300-dimensional embedding obtained via GloVe [223]. The bi-directional GRU of our text encoder is with hidden size of 512. The triplet of generator, posterior and prior networks in our VAE are 1-layer GRUs with hidden size 1,024. The size of the noise vector $\mathbf{z}$ and the attention vector $\mathbf{w}_{att}$ are 128 and 512, respectively. Values of $\lambda_{spr}, \lambda_{smt}, \lambda_{mot}$ are set to 0.001, 0.001, and 1, respectively. λ_{KL} is set to 0.01 for HumanML3D, and to 0.05 for KIT-ML dataset. Teacher forcing rate p_{tf} is 0.4 throughout all experiments. We use Adam optimizer with learning rate of $2e^{-4}$. In text2motion training, the learning rate of motion decoder D is particularly set to $2e^{-5}$, 10 times smaller than other networks. T_{max} and T_{cur} are 50 and 8 respectively.

Z-score normalization is applied to both training and testing data. And we scale the magnitude of features $(\dot{r}^a, \dot{r}^x, \dot{r}^z, r^y, \mathbf{c}^f)$ by a value of 5 to amplify their importance. In addition, to enhance the robustness of our method, we introduce randomness to training data feed by cutting off the last 4 frames of each input pose sequence with probability of coin flipping.

4.4.1 Evaluation Metrics and Baselines

Evaluation Metrics from [1] are adopted here, which include Frechet Inception Distance (FID), diversity and multimodality. For quantitative evaluation, a motion feature extractor and text feature extractor is trained under contrastive loss to produce geometrically close feature vectors for matched text-motion pairs, and vice versa. In addition, the *R-precision* and *MultiModal distance* are proposed in this chapter as complementary metrics, as follows. Consider R-precision: for each generated motion, its ground-truth text description and 31 randomly selected mismatched descriptions from the test set form a description pool. This is followed by calculating and ranking the Euclidean distances between the motion feature and the text feature of each description in the pool. We then count the average accuracy at top-1, top-2 and top-3 places. The ground truth entry falling into the top-k candidates is treated as successful retrieval, otherwise it fails. Meanwhile, MultiModal distance is computed as the average Euclidean distance between the motion feature of each generated motion and the text feature of its corresponding description in test set.

Baseline Methods. We compare our work to three state-of-the-art methods: Seq2Seq [37], Language2Pose [12] and Text2Gesture [219]. As with all existing methods, they are deterministic methods. Considering the stochastic nature of our task, we adapt two non-deterministic methods from related fields for more fair and thorough evaluations: MoCoGAN [21] and Dance2Music [53]. The former is widely used for conditioned video synthesis, and the latter produces 2D dancing motion sequences from audio signals. Proper changes are made to allow these methods generating 3D motions from text.

4.4.2 Quantitative Evaluation

Table 4.2 and Table 4.3 present the quantitative results on HumanML3D and KIT-ML datasets, respectively. For fair comparison, each experiment is repeated 20 times, and a statistical interval

Methods	R Precision↑			FID↓	MultiModal Dist↓	Diversity→	MultiModality↑
	Top 1	Top 2	Top 3				
Real motions	0.511$^{\pm.003}$	0.703$^{\pm.003}$	0.797$^{\pm.002}$	0.002$^{\pm.000}$	2.974$^{\pm.008}$	9.503$^{\pm.065}$	-
Seq2Seq[37]	0.180$^{\pm.002}$	0.300$^{\pm.002}$	0.396$^{\pm.002}$	11.75$^{\pm.035}$	5.529$^{\pm.007}$	6.223$^{\pm.061}$	-
Language2Pose[12]	0.246$^{\pm.002}$	0.387$^{\pm.002}$	0.486$^{\pm.002}$	11.02$^{\pm.046}$	5.296$^{\pm.008}$	7.676$^{\pm.058}$	-
Text2Gesture[219]	0.165$^{\pm.001}$	0.267$^{\pm.002}$	0.345$^{\pm.002}$	7.664$^{\pm.030}$	6.030$^{\pm.008}$	6.409$^{\pm.071}$	-
MoCoGAN[21]	0.037$^{\pm.000}$	0.072$^{\pm.001}$	0.106$^{\pm.001}$	94.41$^{\pm.021}$	9.643$^{\pm.006}$	0.462$^{\pm.008}$	0.019$^{\pm.000}$
Dance2Music[53]	0.033$^{\pm.000}$	0.065$^{\pm.001}$	0.097$^{\pm.001}$	66.98$^{\pm.016}$	8.116$^{\pm.006}$	0.725$^{\pm.011}$	0.043$^{\pm.001}$
Ours w/ real length	**0.457**$^{\pm.002}$	**0.639**$^{\pm.003}$	**0.740**$^{\pm.003}$	**1.067**$^{\pm.002}$	**3.340**$^{\pm.008}$	**9.188**$^{\pm.002}$	2.090$^{\pm.083}$
Ours	0.455$^{\pm.003}$	0.636$^{\pm.003}$	0.736$^{\pm.002}$	1.087$^{\pm.021}$	3.347$^{\pm.008}$	9.175$^{\pm.083}$	**2.219**$^{\pm.074}$

Table 4.2: Quantitative evaluation on the HumanML3D test set. All baselines directly use real motion lengths, while our approach (Ours) instead resorts to the sequence length sampled from the text2length module. ± indicates 95% confidence interval, and → means the closer to Real motions the better. **Bold** face indicates the best result, while underscore refers to the second best.

Methods	R Precision↑			FID↓	MultiModal Dist↓	Diversity→	MultiModality↑
	Top 1	Top 2	Top 3				
Real motions	0.424$^{\pm.005}$	0.649$^{\pm.006}$	0.779$^{\pm.006}$	0.031$^{\pm.004}$	2.788$^{\pm.012}$	11.08$^{\pm.097}$	-
Seq2Seq[37]	0.103$^{\pm.003}$	0.178$^{\pm.005}$	0.241$^{\pm.006}$	24.86$^{\pm.348}$	7.960$^{\pm.031}$	6.744$^{\pm.106}$	-
Language2Pose[12]	0.221$^{\pm.005}$	0.373$^{\pm.004}$	0.483$^{\pm.005}$	6.545$^{\pm.072}$	5.147$^{\pm.030}$	9.073$^{\pm.100}$	-
Text2Gesture[219]	0.156$^{\pm.004}$	0.255$^{\pm.004}$	0.338$^{\pm.005}$	12.12$^{\pm.183}$	6.964$^{\pm.029}$	9.334$^{\pm.079}$	-
MoCoGAN[21]	0.022$^{\pm.002}$	0.042$^{\pm.003}$	0.063$^{\pm.003}$	82.69$^{\pm.242}$	10.47$^{\pm.012}$	3.091$^{\pm.043}$	0.250$^{\pm.009}$
Dance2Music[53]	0.031$^{\pm.002}$	0.058$^{\pm.002}$	0.086$^{\pm.003}$	115.4$^{\pm.240}$	10.40$^{\pm.016}$	0.241$^{\pm.004}$	0.062$^{\pm.002}$
Ours w/ real length	**0.370**$^{\pm.005}$	**0.569**$^{\pm.007}$	**0.693**$^{\pm.007}$	**2.770**$^{\pm.109}$	**3.401**$^{\pm.008}$	**10.91**$^{\pm.119}$	1.482$^{\pm.065}$
Ours	0.361$^{\pm.006}$	0.559$^{\pm.007}$	0.681$^{\pm.007}$	3.022$^{\pm.107}$	3.488$^{\pm.028}$	10.72$^{\pm.145}$	**2.052**$^{\pm.107}$

Table 4.3: Quantitative evaluation on the KIT-ML test set. All baselines directly use real motion lengths, while our approach (Ours) instead resorts to the sequence length sampled from the text2length module. ± indicates 95% confidence interval, and → means the closer to the real motion the better.

with 95% confidence is reported. Since all baseline methods directly use the ground-truth motion length in generating a new motion, for fair comparison, we also consider a variant of our approach by removing the text2length sampling module (i.e. *ours w/ real length*). The high R precision of real motions evidences the reliability of the proposed R-precision metric, which sets a upper performance limit for all methods. Overall, we have the following observations from Table 4.2 and Table 4.3. First, our approach clearly outperform all comparison methods by a significant margin, over all metrics and on both datasets. Seq2Seq [37] and Text2Gesture [219] directly map textual data to human dynamics by their neural machine translation architecture of encoder-decoder and transformer; they however find difficulty in retaining the sense of realistic motions during their processes. This results in low motion-based text retrieval precision, and high FID values.

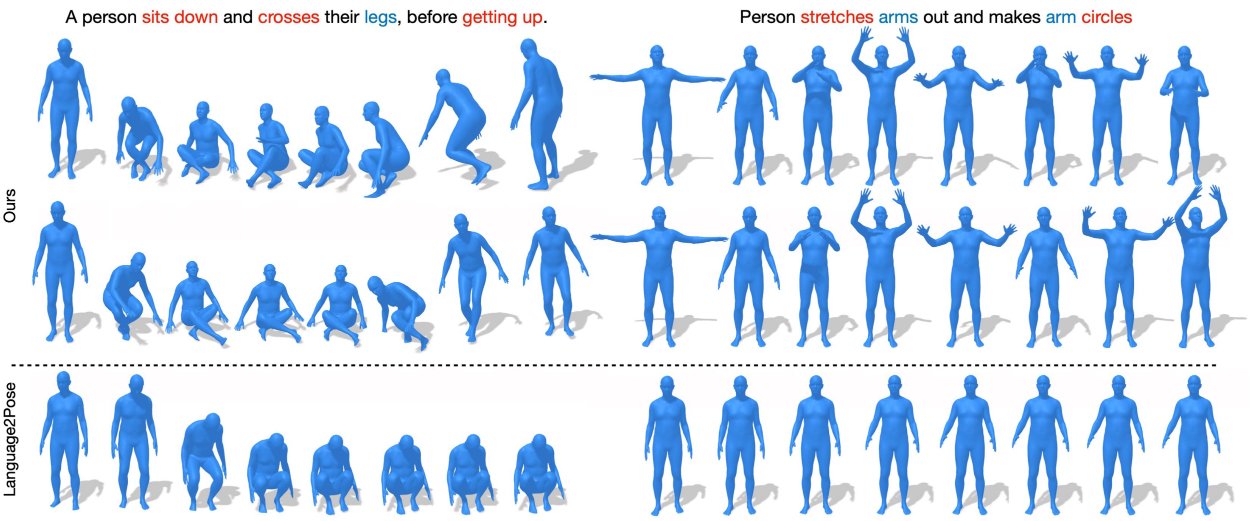

A person sits down and crosses their legs, before getting up.
Person stretches arms out and makes arm circles
Ours
Language2Pose

Figure 4.6: **Visual results** of our approach vs. those of Language2Pose[12]. Given each input description, we show two generated motions from our approach, and one motion from Language2Pose (since it is a deterministic method). As our generated motions are of variable length, only key frames from each sequence are displayed.

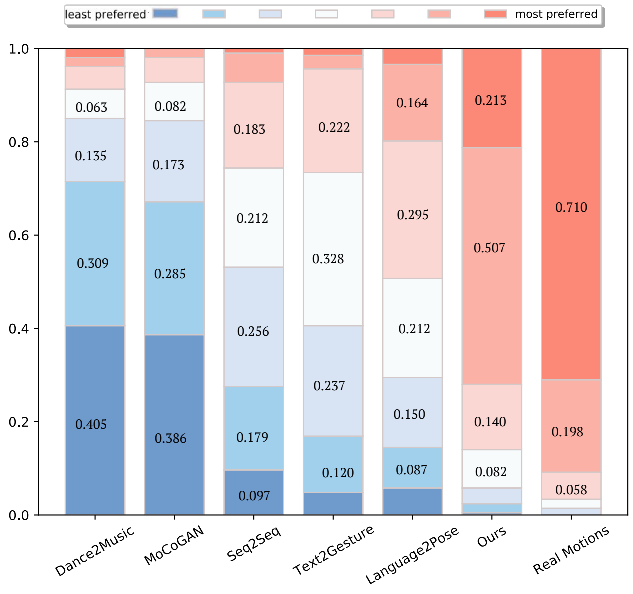


Figure 4.7: Quantitative evaluation of user preference among the generated motions. For each comparison method, a color bar (from blue to red) indicates the percentage of its preference levels (from least to most preferred).

Language2Pose [12] performs better on generation quality by incorporating a co-embedding space, yet the results are very far from real motions. The motions generated by non-deterministic methods of MoCoGAN [21] and Dance2Music [52] are unfortunately of severely low quality, as manifested by their low diversity and multimodality scores – a result of being unfaithful to the input text. On the contrary, the variant of our approach directly using real motion length (Ours w/ real length) achieves the optimal performance on almost all metrics. Our default approach that uses text2length sampling (Ours) possesses a comparable performance in R-precision and FID scores, yet it is more capable of synthesizing diverse motions, as reflected especially in the diversity

& multimodality scores.

User Study In addition to the aforementioned objective evaluations, a Crowd-sourced subjective evaluation via Amazon Mechanical Turk is conducted concerning the visual perceptual quality of the generated motions. For each comparison method, motions are generated using 50 descriptions randomly selected from the test set. For each description, the results of different methods are shown to 5 AMT users, who are asked to rank their preference over these motions based on the motion realism and the magnitude they are aligned to the intended text descriptions. Only AMT users with *master* recognition are considered.

The preference results are shown in Fig. 4.7. Overall our approach is most preferred by the users; meanwhile, two non-deterministic methods are least preferred, as their motions exhibit severely distortions; Seq2Seq and Text2Gesture gain comparably more positive scores from users; Language2Pose becomes the second most preferred. Moreover, a significant portion (around 72%) of motions generated by our approach are considered at top-2 by users, i.e. being on par with or only next to the real human motions. This user study brings strong evidence of our approach capable of synthesizing visually realistic motions.

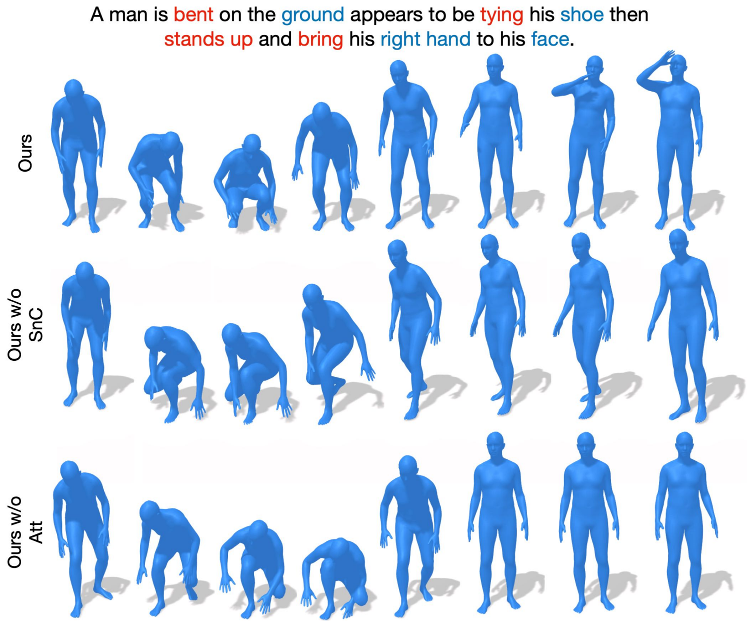

A man is bent on the ground appears to be tying his shoe then stands up and bring his right hand to his face.
Ours
Ours w/o SnC
Ours w/o Att

Figure 4.8: Visual comparison of motion results generated by ours, ours w/o SnC, and our w/o Att, all provided with the same description.

Methods	R Precision↑			FID↓
	Top 1	Top 2	Top 3	
Ours	$\mathbf{0.455}^{\pm.003}$	$\mathbf{0.636}^{\pm.003}$	$\mathbf{0.736}^{\pm.002}$	$\mathbf{1.087}^{\pm.021}$
w/o SnC	$0.370^{\pm.002}$	$0.538^{\pm.003}$	$0.642^{\pm.003}$	$1.200^{\pm.027}$
w/o Att	$0.396^{\pm.002}$	$0.570^{\pm.002}$	$0.674^{\pm.003}$	$1.833^{\pm.032}$
w/o PoS	$0.443^{\pm.003}$	$0.622^{\pm.003}$	$0.723^{\pm.003}$	$1.157^{\pm.016}$
w/o PoE	$0.444^{\pm.005}$	$0.627^{\pm.003}$	$0.729^{\pm.002}$	$1.229^{\pm.020}$

Table 4.4: Ablation study on the HumanML3D dataset, with SnC denoting *motion snippet code*, Att the *local word attention*, PoS the *Part-of-Speech tag*, and PoE the *Positional Encoding*.

4.4.3 Qualitative Evaluation

Fig. 4.6 displays qualitative comparisons of our approach vs. Language2Pose [12], the best-performing baseline. Motions from other comparison methods are too distorted to be rendered with the SMPL human shapes [47]. Language2Pose sometimes captures partial concepts (*e.g.,* sit down) in the input text. It however fails to understand the global textual information. Moreover, the generated motions tend to be frozen after a short while. In contrast, our approach is capable of generating visually appealing motions which accurately reflect the fine details in text descriptions, in terms of the *gesture*, *actions*, *body parts* and *timing*. Furthermore, from the same input text, our generated motions are sufficiently diverse.

4.4.4 Ablation Analysis

Table 4.4 quantifies the effects of different components in our approach on HumanML3D dataset. A sharp drop of performance is observed when *snippet code* (i.e. SnC) or *word attentions* (i.e. Att) is removed, with a decreasing R-Precision of over 6%. On the contrary, the influence of *positional encoding* (i.e. PoE) and *part-of-speech* (i.e. POS) are relatively less significant, given a drop of R-precision around 2%. In Fig. 4.8, a visual comparison of synthesized motions from ours, ours w/o SnC, and ours w/o Att from the same input text is shown. While snippet codes are not applied, the resulting motion appears to be visually plausible and context-aware at the beginning; it however fails to faithfully follow the text description as time goes on. This may be attributed to the lack of characterization in temporal dependencies. Similar phenomenon is observed in motions from ours w/o word attentions. On the other hand, the result of our approach aligns sufficiently with the

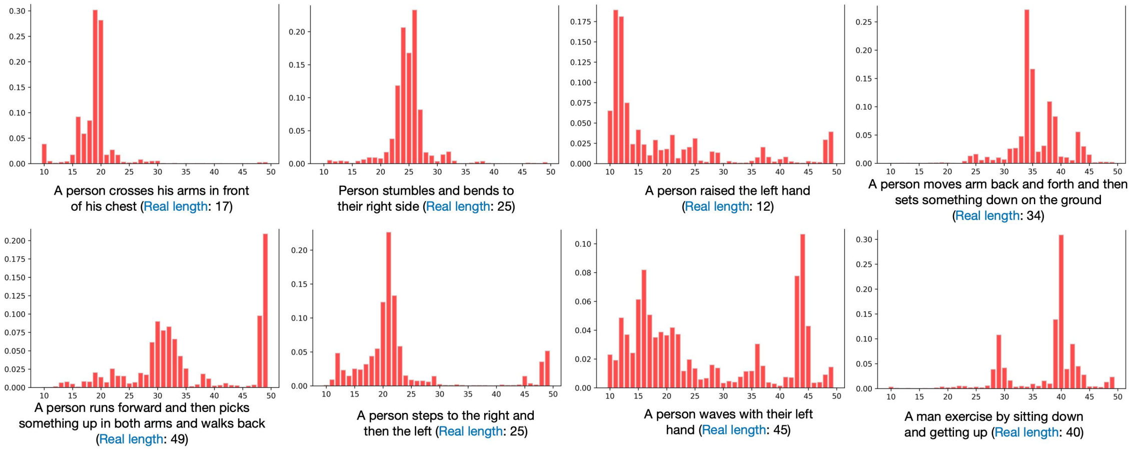


Figure 4.9: **Exemplar results of text2length.** For each subplot, given one text description (bottom), the estimated probability density of snippet code length is visualized in histogram. The corresponding real lengths are highlighted in blue. Length of motion is 4 times of the snippet code.

textual concepts throughout.

4.4.5 Text2length sampling

In Figure 4.9, we gives some examples of estimated snippet code length distribution from our text2length provided with text descriptions. Note the corresponding motion lengths are 4 times of the presented number. During training, motions with less than 40 frames (2s) are discarded. Therefore, here we produce the distribution over discrete values from 10 to 50. As shown, our method yields probability densities in which the values with high confidence are reasonably close to the corresponding ground truth value. In addition, cyclic motions (e.g., *a person waves with their left hand*) have relatively flatter probability density compared to non-cyclic motions (e.g., *Person stumbles and bends to their right side*). During infer-

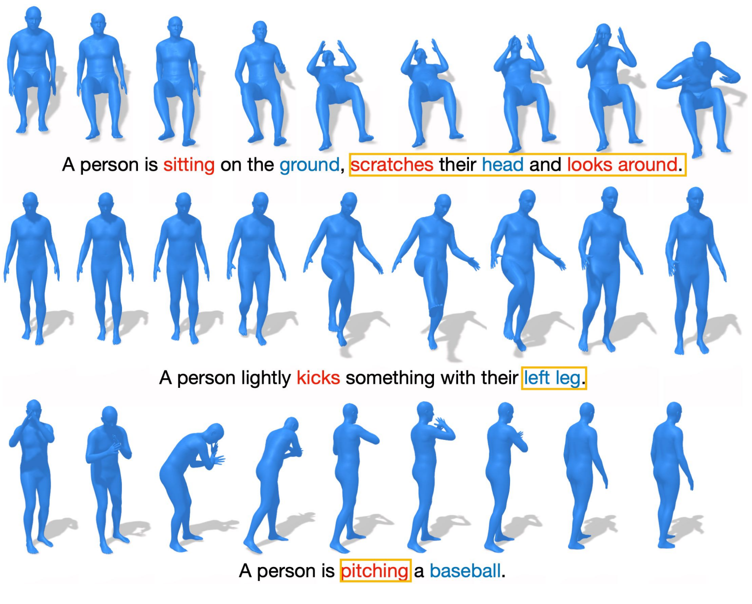

A person is sitting on the ground, scratches their head and looks around.
A person lightly kicks something with their left leg.
A person is pitching a baseball.

Figure 4.10: **Examples of failure cases.**

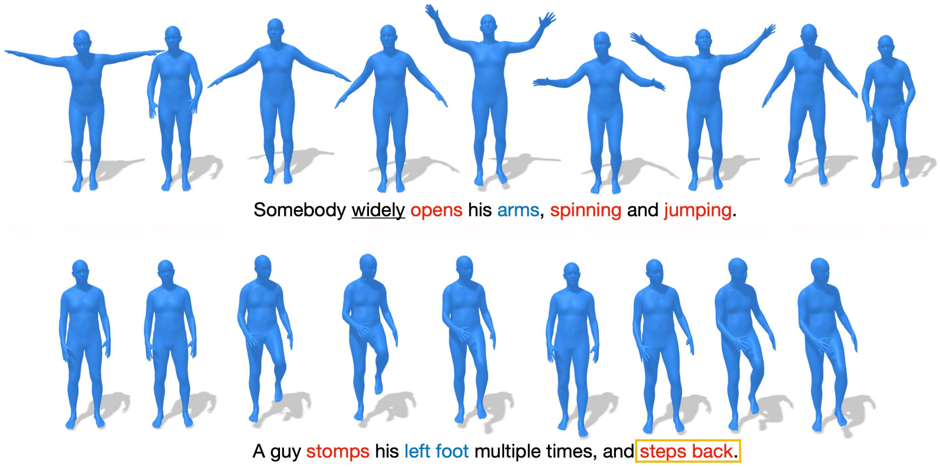

Somebody widely opens his arms, spinning and jumping.
A guy stomps his left foot multiple times, and steps back.

Figure 4.11: **Out-of-the-dataset results.** Key frames are displayed. Yellow bound box indicates the parts of description that generated motions fail to present.

ence, the target sequence length will be randomly sampled from the estimated probability density, which further increases variety of generated results.

4.4.6 Failures and Limitations

Figure 4.11 presents two results generated from out-of-the-dataset descriptions. In other words, the descriptions are collected independently to our dataset. Our method are able to demonstrate the overall content in the text descriptions. Nonetheless, the model may fail in descriptions involving rare actions (e.g., *'stomp'*). In the second pose sequence, '*step back*' is unfaithfully missed after *'stomping foot'*.

When further looking into multimodal alignment, our method sometime fail in finer grained descriptions. We showcase some failure results from our method in Figure 4.10. Actions such as "scratch" and "pitching baseball" are too sophisticated and beyond the capability of our method. We also find our method less sensitive to fine-grained descriptions regarding to body parts, for example, *left/right leg*. There are still space for exploring broader scenarios such as, text descriptions with overlength, environment interactions.

4.5 Conclusion and Outlook

This chapter looks into an emerging research problem of generating 3D human motions grounded on natural language descriptions, where we especially emphasize on diverse and natural motion generation. It leads to our two-stage pipeline, where the text2length module sampled from the estimated motion length distribution given text description; the text2motion module generates motions of sampled motion length from input text, accomplished by our temporal VAE. A large-scale human motion-language dataset is constructed, with the expectation of facilitating the development and evaluation of new methods in the community. Extensive quantitative and qualitative experiments demonstrate the effectiveness of our approach.

Outlook. In the future, it will be interesting to explore the large-scale pre-trained language model, such as BERT [185] and CLIP [64], to accommodate more diverse and challenging textual inputs. On the other hand, text description may find difficult precisely describing motion dynamics, which is insufficient for professional use. Combining precise control signals (e.g., trajectory) and textual description for motion generation could find broader range of audience in practical use.

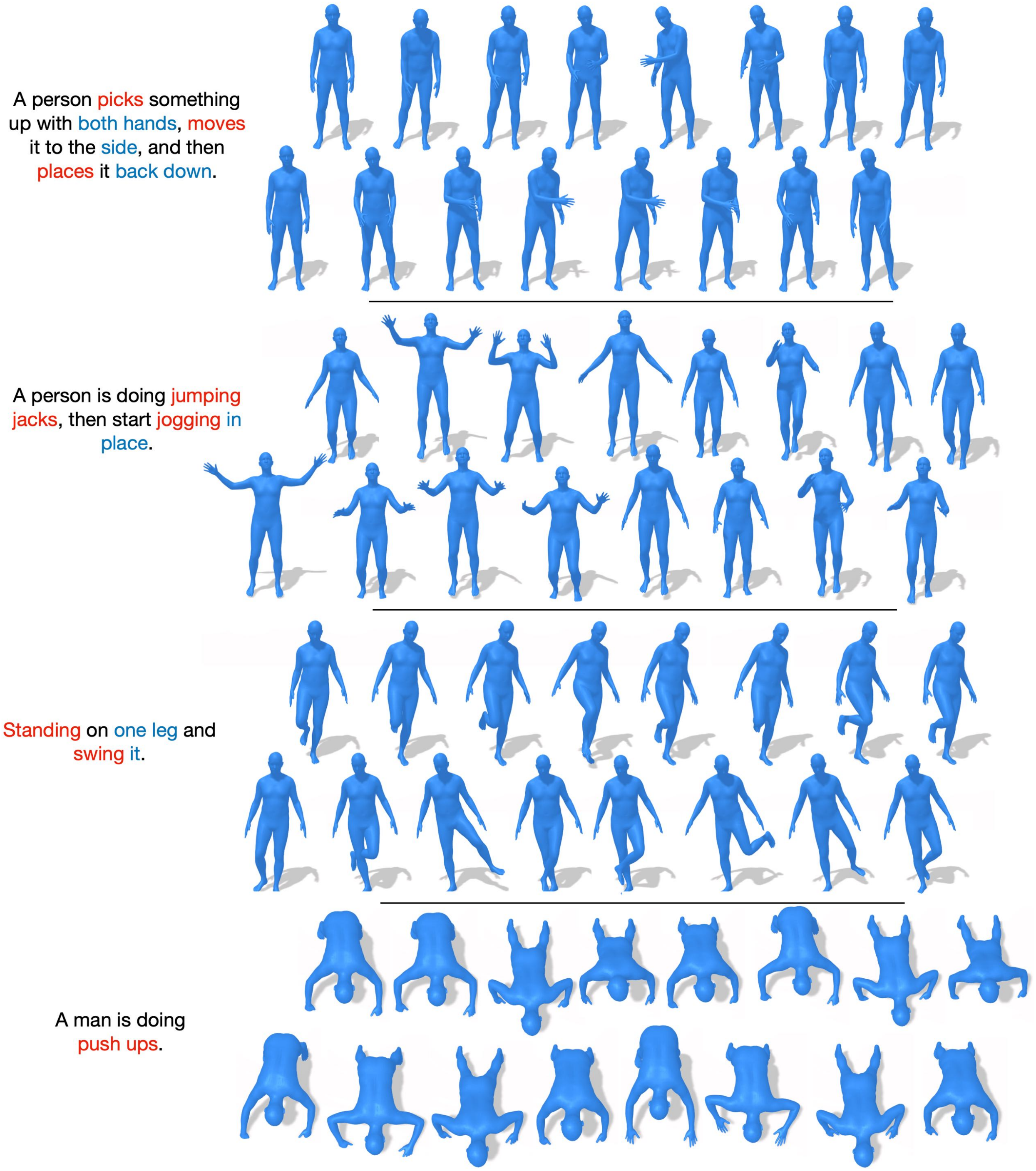

A person picks something up with both hands, moves it to the side, and then places it back down.
A person is doing jumping jacks, then start jogging in place.
Standing on one leg and swing it.
A man is doing push ups.

Figure 4.12: **Additional examples generated from our method.** For each description (left), we show two distinct synthetic motions (right). Key frames are displayed for each sequence.

Chapter 5

Generative Human Motion Stylization in Latent Space

Fig. 5.1 gives a brief view of our fourth project, *human motion stylization*, which aims to revise the style of an input motion while keeping its content unaltered. Unlike existing works that operate directly in pose space, we leverage the *latent space* of pretrained autoencoders as a more expressive and robust representation for motion extraction and infusion. Building upon this, we present a novel *generative* model that produces diverse stylization results of a single motion (latent) code. During training, a motion code is decomposed into two coding components: a deterministic content code, and a probabilistic style code adhering to a prior distribution; then a generator massages the random combination of content and style codes to reconstruct the corresponding motion codes. Our approach is versatile, allowing the learning of probabilistic style space from either style-labeled or unlabeled motions, providing notable flexibility in stylization as well. In inference, users can opt to stylize a motion using style cues from a reference motion or a label. Even in the absence of explicit style input, our model facilitates novel re-stylization by sampling from the unconditional style prior distribution. Experimental results show that our proposed stylization models, despite their lightweight design, outperform the state-of-the-art in style reenactment, content preservation, and generalization across various applications and settings. This chapter has been accepted to the

International Conference on Learning Representations [4].

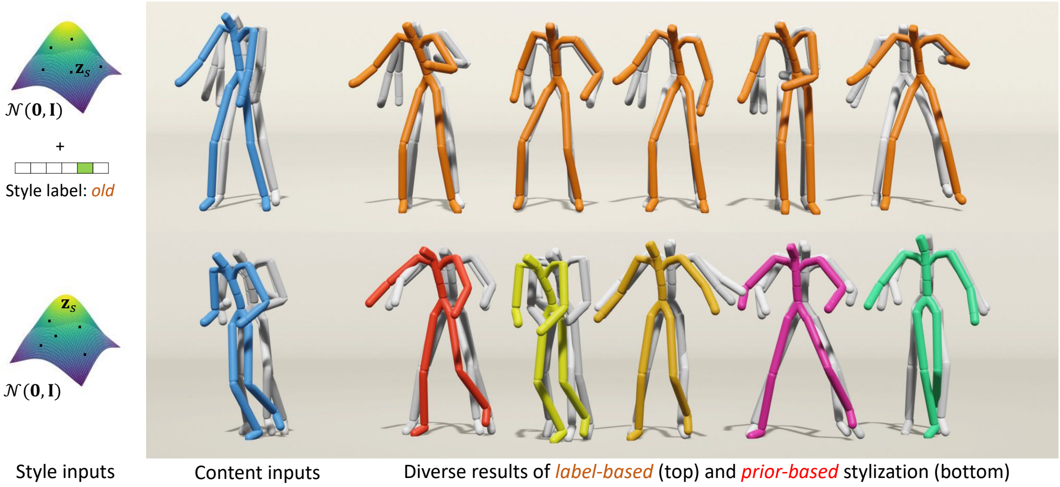

z_s
𝒩(0, I)
+
Style label: old
z_s
𝒩(0, I)
Style inputs
Content inputs
Diverse results of label-based (top) and prior-based stylization (bottom)

Figure 5.1: (**Top**) Given an input motion and target style label (*i.e., old*), our label-based stylization generates diverse results following provided label. (**Bottom**) Without any style indicators, our prior-based method randomly re-stylizes the input motion using sampled prior styles $\mathbf{z}_s$. Five distinct stylized motions from the same content are presented, with poses synchronized and history in gray. See Fig. 5.3 (b) and (d) for implementations.

5.1 Introduction

The motions of our humans are very expressive and contain a rich source of information. For example, by watching a short duration of one individual's walking movement, we could quickly recognize the person, or discern the mood, age, or occupation of the person. These distinguishable motion traits, usually thought of as styles, are therefore essential in film or game industry for realistic character animation. It is unfortunately unrealistic to acquire real-world human motions of various styles solely by motion capture. Stylizing existing motions using a reference style motion (*i.e.,* motion-based), or a preset style label (*i.e.,* label-based) thus becomes a feasible solution.

Deep learning models have recently enabled numerous data-driven methods for human motion stylization. These approaches, however, still find their shortfalls. A long line of existing works [6, 103, 109, 110] are limited to deterministic stylization outcomes. [108, 111] though allows diverse stylization, their results are far from being satisfactory, and the trained models struggle to generalize to other motion datasets. Furthermore, all of these approaches directly manipulate style within raw poses, a redundant and potentially noisy representation of motions. Meanwhile, they often possess rigid designs, allowing for only supervised or unsupervised training, with style input typically limited to either reference motions or labels, as shown in Tab. 5.1.

In this chapter, we introduce a novel *generative* stylization framework for 3D human motions. Inspired by the recent success of content synthesis in latent space [3, 183, 79, 171], we propose to use *latent* motion features (namely motion code) of pretrained convolutional autoencoders as the intermedia for motion style extraction and infusion. Compared to raw poses, the benefits are three-folds: **(i)** Motion codes are more compact and expressive, containing the most discriminative features of raw motions; **(ii)** Autoencoders can be learned once on a large dataset and reused for downstream datasets. Thanks to the inductive bias of CNN [224], the learned motion code features typically contains less noise, resulting in improved generalization, as empirically demonstrated in Tables 5.2 and 5.3; **(iii)** Practically, motion code sequences are much shorter than motions, making them more manageable in neural networks. Building on this, our latent stylization framework decomposes the motion code into two components: a temporal and deterministic *content* code, and a global probabilistic *style* code confined by a prior Gaussian distribution. The subsequent generator recombines content and style to synthesize valid motion code. During training, besides auto-encoding and decoding, we swap the contents and styles between random pairs, and the resulting motion codes are enforced to recover the source contents and styles through cycle reconstruction. To further improve content-style disentanglement, we propose a technique called *homo-style alignment*, which encourages the alignment of style spaces formed by different motion sub-clips from the same sequence. Lastly, the global velocity of resulting motions are obtained through a pre-trained global motion regressor.

Our approach offers versatile stylization capabilities (Tab. 5.1), accommodating various conditioning options during both training and inference: 1) Deterministic stylization using style from **exemplar motions**; 2) In the label conditioned setting, our model can perform diverse stylization based on provided **style labels**, as in Fig. 5.1 (top); 3) In the unconditional setting, our model can randomly sample styles from the **prior distribution** to achieve stochastic stylization, as in Fig. 5.1 (bottom). Benefiting from our latent stylization and lightweight model design, our

	Supervised (w style label)		Unsupervised (w/o style label)		Generative
	Motion-based	Label-based	Motion-based	Prior-based	
Xia et al. [7]		✓			
Holden et al. [103, 106]			✓		
Aberman et al. [6]	✓				
Park et al. [108]	✓	✓			✓
Tao et al. [110]	✓				
Jang et al. [109]			✓		
Ours	✓	✓	✓	✓	✓

Table 5.1: Our generative framework owns flexible design for training and inference.

approach achieves state-of-the-art performance while being 14 times faster than the most advanced prior workJang et al. [109], as shown in Table 5.5.

Our key contributions can be summarized as follows. Firstly, we propose a novel generative framework, using motion latent features as an advanced alternative representation, accommodating various training and inference schemes in a single framework. Secondly, through a comprehensive suite of evaluations on three benchmarks, our framework demonstrates robust and superior performance across all training and inference settings, with notable efficiency gains.

5.2 Generative Motion Stylization

An overview of our method is described in Figure 5.2. Motions are first projected into the latent space of autoencoders (Sec. 5.2.1). With this, our latent stylization framework learns to disentangle the content and style information from the input motion (latent) code (Sec. 5.2.2). During inference, our learned models support multiple applications for motion stylization (Sec. 5.2.3).

5.2.1 Motion Latent Representation

As a pre-processing step, we learn a motion autoencoder that builds the mapping between motion and latent space. More precisely, given a pose sequence $\mathbf{P} \in \mathbb{R}^{T \times D}$, where T denotes the number of poses and D pose dimension, the encoder $\mathcal{E}$ encodes $\mathbf{P}$ into a motion code $\mathbf{z} = \mathcal{E}(\mathbf{P}) \in \mathbb{R}^{T_z \times D_z}$, with T_z and D_z the temporal length and spatial dimension respectively, and then the decoder $\mathcal{D}$ recovers the input motion from the latent features, formally $\hat{\mathbf{P}} = \mathcal{D}(\mathbf{z}) = \mathcal{D}(\mathcal{E}(\mathbf{P}))$.

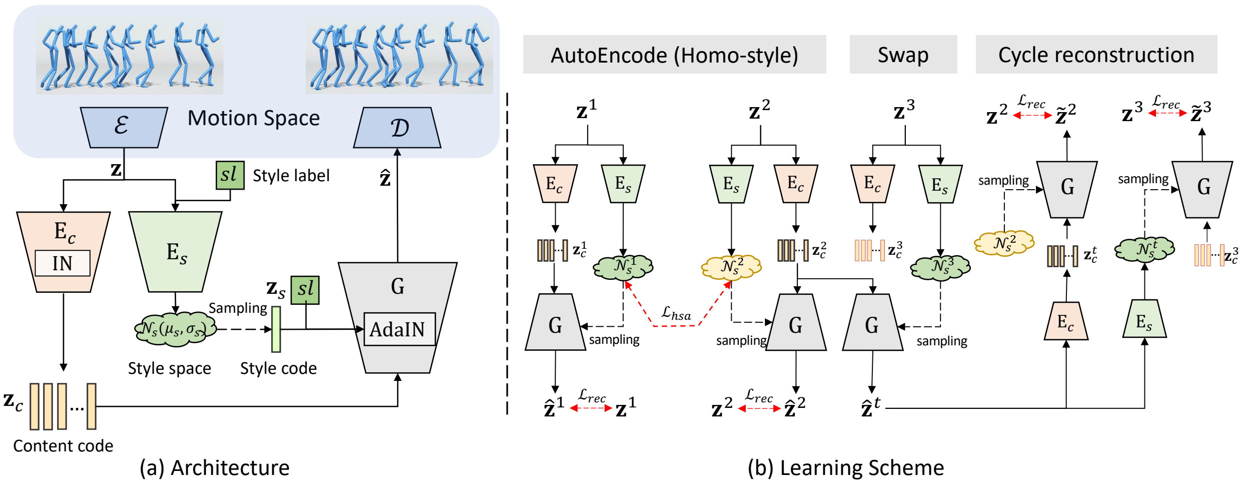
A well-learned latent space should exhibit smoothness and low variance, which allows for continuous and gradual changes in the input data to be reflected in the latent space. It encourages higher generalization ability and facilitates efficient deep learning based on this latent space. In this chapter, we experiment with two kinds of regularization methods in latent space: 1) as in VAE [149], the latent space is formed under a light KL regularization towards standard normal distribution $\mathcal{L}^{l}_{kld} = \lambda^{l}_{kld} D_{\mathrm{KL}}(\mathbf{z}||\mathcal{N}(\mathbf{0}, \mathbf{I}))$; and 2) similar to [3], we train the classical autoencoder and impose L1 penalty on the magnitude and smoothness of motion code sequences, giving $\mathcal{L}^{l}_{reg} = \lambda_{l1}\|\mathbf{z}\|_1 + \lambda_{sms}\|\mathbf{z}_{1:T_z} - \mathbf{z}_{0:T_z-1}\|_1$. Our motion encoder $\mathcal{E}$ and decoder $\mathcal{D}$ are simply 1-D convolution layers with downsampling and upsampling scale of 4 (*i.e.,* $T = 4T_z$), resulting in a more compact form of data that captures temporal semantic information.

5.2.2 Motion Latent Stylization Framework

As depicted in Figure 5.2, our latent stylization framework aims to yield a valid parametric style space, and meanwhile, preserve semantic information in content codes as much as possible. This is achieved by our specific model design and dedicated learning strategies.

Model Architecture.

There are three principal components in our framework: a content encoder E_c, a style encoder E_s and a generator G, as in Figure 5.2 (a). The content encoder converts the a motion code $\mathbf{z} \in \mathbb{R}^{T_z \times D_z}$ into a content code $\mathbf{z}_c \in \mathbb{R}^{T^c_z \times D^c_z}$ that keeps a temporal dimension T^c_z, where global statistic features (style) are erased through instance normalization (IN). For style, existing works [108, 6, 109] generate deterministic style code from motion input. In contrast, our style encoder E_s, taking $\mathbf{z}$ and style label sl as input, produces a vector Gaussian distribution $\mathcal{N}_s(\mu_s, \sigma_s)$ to formulate the style space, from which a style code $\mathbf{z}_s \in \mathbb{R}^{D^s_z}$ is sampled. The asymmetric shape of content code $\mathbf{z}_c$ and style code $\mathbf{z}_s$ are designed of purpose. We expect the former to capture local semantics while the latter encodes global features, as what style is commonly thought of. Content code is subsequently fed into the convolution-based generator G, where the mean and variance of each layer output are



(a) Architecture

(b) Learning Scheme

Figure 5.2: **Approach overview.** (a) A pre-trained autoencoder $\mathcal{E}$ and $\mathcal{D}$ (Sec. 5.2.1) builds the mappings between *motion* and *latent* spaces. Motion (latent) code $\mathbf{z}$ is further encoded into two parts: content code $\mathbf{z}_c$ from content encoder (E$_c$), and style space $\mathcal{N}_s$ from style encoder (E$_s$) that take style label sl as an additional input. The content code ($\mathbf{z}_c$) is decoded back to motion code ($\hat{\mathbf{z}}$) via generator G. Meanwhile, a style code $\mathbf{z}_s$ is sampled from style space ($\mathcal{N}_s$), together with style label (sl), which are subsequently injected to generator layers through adaptive instance normalization (AdaIN). (b) Learning scheme, where style label (sl) is omitted for simplicity. Our model is trained by autoencoding for content and style coming from the **same** input. When decoding with content from **different** input (*i.e.*, swap), we enforce the resulting motion code ($\hat{\mathbf{z}}^t$) to follow the cycle reconstruction constraint. For motion codes ($\mathbf{z}^1$, $\mathbf{z}^2$) segmented from the same sequence (homo-style), their style spaces are assumed to be close and learned with style alignment loss $\mathcal{L}_{hsa}$.

modified by an affine transformation of style information (*i.e.*, style code and label), known as adaptive instance normalization (AdaIN). The generator aims to transform valid combinations of content and style into meaningful motion codes in the latent space.

Learning Scheme

With the model mentioned above, we propose a series of strategies for learning disentangled content and style representations. Figure 5.2 (b) illustrates our learning scheme. Note the input of style label sl is omitted for simplicity. During training, for each iteration, we design three groups of inputs: $\mathbf{z}^1$, $\mathbf{z}^2$ and $\mathbf{z}^3$, where $\mathbf{z}^1$ and $\mathbf{z}^2$ are motion code segments coming from the same sequence and $\mathbf{z}^3$ can be any other segments.

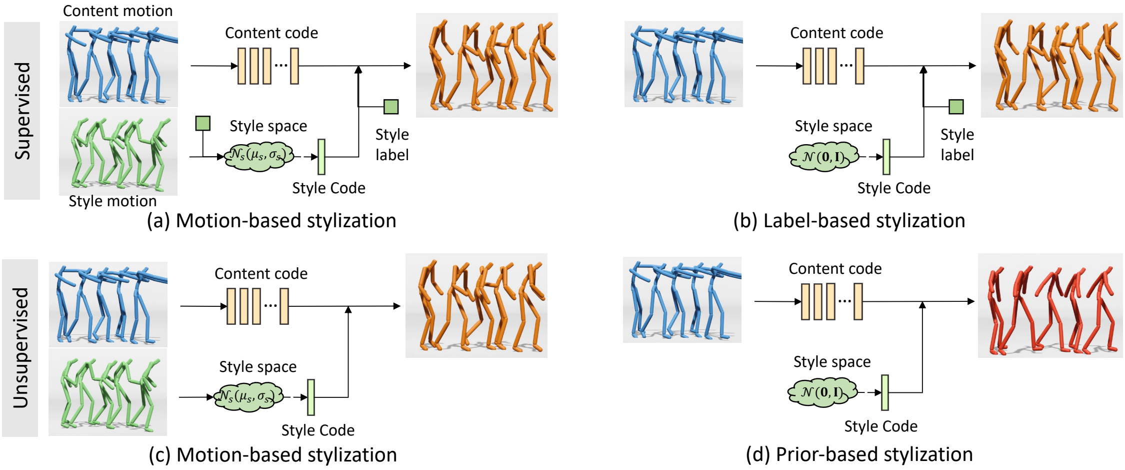

Supervised
Content motion
Content code
Style space
$\mathcal{N}_s(\mu_s, \sigma_s)$
Style label
Style Code
Style motion
(a) Motion-based stylization
Content code
Style space
$\mathcal{N}(\mathbf{0}, \mathbf{I})$
Style label
Style Code
(b) Label-based stylization
Unsupervised
Content code
Style space
$\mathcal{N}_s(\mu_s, \sigma_s)$
Style Code
(c) Motion-based stylization
Content code
Style space
$\mathcal{N}(\mathbf{0}, \mathbf{I})$
Style Code
(d) Prior-based stylization

Figure 5.3: During inference, our approach can stylize input content motions with the style cues from (a, c) motion, (b) style label and (d) unconditional style prior space.

AutoEncoding $\mathcal{L}_{rec}$. We train our latent stylization framework partly through autoencoding, that given motion latent codes, like $\mathbf{z}^1$ and $\mathbf{z}^2$, the generator learns to reconstruct the input from the corresponding encoded content and style features, formally $\hat{\mathbf{z}} = \mathrm{G}(\mathrm{E}_c(\mathbf{z}), \mathrm{E}_s(\mathbf{z}))$. For accurate reconstruction, we decode the resulting motion latent codes ($\hat{\mathbf{z}}^1$ and $\hat{\mathbf{z}}^2$) back to motion space ($\hat{\mathbf{P}}^1$ and $\hat{\mathbf{P}}^2$) through $\mathcal{D}$, and apply L1-distance reconstruction in both latent and motion space:

$$\mathcal{L}_{rec} = \sum_{i \in \{1,2\}} \|\hat{\mathbf{z}}^i - \mathbf{z}^i\|_1 + \|\hat{\mathbf{P}}^i - \mathbf{P}^i\|_1 \tag{5.1}$$

Homo-style Alignment $\mathcal{L}_{hsa}$. For the motion segments in one motion sequence, we could usually assume their styles are similar in all aspects. This is a strong supervision signal especially when style annotation is unavailable, dubbed *homo-style alignment* in our work. Since $\mathbf{z}^1$ and $\mathbf{z}^2$ belong to the same sequence, their learned style spaces are enforced to be close:

$$\mathcal{L}_{hsa} = D_{\mathrm{KL}}(\mathcal{N}_s^1(\mu_s^1, \sigma_s^1) \| \mathcal{N}_s^2(\mu_s^2, \sigma_s^2)) \tag{5.2}$$

Swap and Cycle Reconstruction $\mathcal{L}_{cyc}$. To further encourage content-style disentanglement, we adopt a cycle consistency constraint [97, 109] when content and style are swapped between different motion codes, such as $\mathbf{z}^2$ and $\mathbf{z}^3$ in Fig. 5.2. Specifically, the generator G takes as input the content from $\mathbf{z}^2$ and the style from $\mathbf{z}^3$, and then produces a new *transferred* motion code $\mathbf{z}^t$, which are supposed to preserve the content information from $\mathbf{z}^2$ and the style from $\mathbf{z}^3$. Therefore, if we re-combine $\mathbf{z}^t$'s content and $\mathbf{z}^2$'s style, the generator should be able to restore $\mathbf{z}^2$. The same to $\tilde{\mathbf{z}}^3$ that are recovered from the mix of $\mathbf{z}^t$'s style and $\mathbf{z}^3$'s content :

$$\mathcal{L}_{cyc} = \sum_{i \in \{2,3\}} \|\tilde{\mathbf{z}}^i - \mathbf{z}^i\|_1 + \|\tilde{\mathbf{P}}^i - \mathbf{P}^i\|_1 \tag{5.3}$$

To ensure smooth and samplable style spaces, we apply a KL loss regularization to all style spaces:

$$\mathcal{L}_{kl} = \sum_{i \in \{1,2,3,t\}} D_{\mathrm{KL}}(\mathcal{N}_s^i(\mu_s^i, \sigma_s^i)) \| \mathcal{N}(\mathbf{0}, \mathbf{I})) \tag{5.4}$$

Overall, our final objective is $\mathcal{L} = \mathcal{L}_{rec} + \lambda_{hsa}\mathcal{L}_{hsa} + \lambda_{cyc}\mathcal{L}_{cyc} + \lambda_{kl}\mathcal{L}_{kl}$. We also have experimented adversarial loss for autoencoding and cycle reconstruction as in [108, 6, 110], which however appears to be extremely unstable in training.

Unsupervised Scheme (*w/o* Style Label). Collecting style labeled motions is resource consuming. Our approach can simply fit in unsupervised setting with just one-line change of code during training—to drop out style label sl input.

Difference of $\mathcal{N}_s$ Learned *w* and *w/o* Style Label. While learning with style label, since both the style encoder E_s and generator G are conditioned on style label, the style space is encouraged to learn style variables other than style label as illustrated in Fig. 5.7 (d). Whereas in unsupervised setting where the networks are agnostic to style label, in order to precisely reconstruct motions, the style space is expected to cover the *holistic* style information, including style label (see Fig. 5.7 (c)).

Global Motion Prediction

Global motion (*i.e.,* root velocity) is perceptually a more sensitive element than local joint motion (e.g., foot skating). However, given one motion, transferring its global motion to another style domain is challenging without supervision of paired data. Previous works commonly calculate the target global motion directly from the content motion, or enforce them to be close in training. This may fail when the transferred motion differs a lot from the source content. In our work, we propose a simple yet effective alternative, which is a small 1D convolutional network that predicts the global motion from local joint motion, simply trained on unlabeled data using objective of mean absolute error. During inference, the global motion of output can be accurately inferred from its local motion.

5.2.3 Inference Phase

As displayed in Figure 5.3, our approach at run time can be used in multiple ways. In *supervised* setting: a) **motion-based** stylization requires the user to provide a style motion and a style label as the style references; and b) **label-based** stylization only asks for a target style label for stylization. With sampled style codes from a standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, we are able to stylize source content motion non-deterministically. In the case of *unsupervised* setting: c) motion-based stylization, which similarly, yields a style code from a reference motion; and d) **prior-based** stylization that samples random style codes from the prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Since there is no other pretext style indications, the output motion could carry any style trait in the style space.

5.3 Experiments

We adopt three datasets for comprehensive evaluation. Aberman et al. [6] is a widely used motion style dataset, which contains 16 distinct style labels including *angry*, *happy*, *Old*, etc, with total duration of 193 minute. Xia et al. [7] is much smaller motion style collection (25 mins) that is captured in 8 styles, with accurate action type annotation (8 actions). The motions are typically shorter than 3s. The other one is CMU Mocap [225], an unlabeled dataset with high diversity and quantity of motion data. All motion data is retargeted to the same 21-joint skeleton structure, with a 10% held-out subset for evaluation. Our autoencoders and global motion regressor are trained on the union of all training sets, while the latent stylization models are trained **excursively** onAberman et al. [6], using the other two for zero-shot evaluation. During evaluation, we use the styles fromAberman et al. [6] test sets to stylize the motions from one of the three test sets. Style space is learned based on motions of 160 poses (5.3s). Note our models supports stylization of arbitrary-length content motions.

Metrics in previous motion stylization works heavily rely on a sparse set of measurements, typically human evaluation and style accuracy. Here, we design a suite of metrics to comprehensively evaluate our approach. We firstly pre-train a style classifier onAberman et al. [6] train set, and use it as a style feature extractor to compute *style recognition accuracy* and *style FID*. For dataset with available action annotation (Xia et al. [7]), an action classifier is learned to extract content features and calculate *content recognition accuracy* and *content FID*. We further evaluate the content preservation using *geodesic distance* of the local joint rotations between input content motion and generated motion. *Diversity* in [53] is also employed to quantify the stochasticity in the stylization results.

Baselines. We compare our method to three state-of-the-art methods [6, 109, 108] in their respective settings. Among these, [6] and [108] are supervised methods learned within GAN framework. [108] learns per-label style space, and a mapping between Gaussian space and style space. At run time, it supports both deterministic motion-based and diverse label-based motion stylization. Note recent work [110] is not included as their approach requires action labels for supervision, which are unavailable in our context.

Implementation Details. Our models are implemented by Pytorch. Motion encoder $\mathcal{E}$ and decoder $\mathcal{D}$ consists of 2 1-D convolution layers; global motion regressor is a 3-layer 1D convolution network. The content encoder E_c and style encoder E_s are also downsampling convolutional networks, where style encoder contains a average pooling layer before the output dense layer. The values of λ_{kld}^{l}, λ_{l1} and λ_{sms} are all set to 0.001, and dimension D_z of $\mathbf{z}$ is 512. During training our latent stylization network, the value of λ_{hsa}, λ_{cyc} and λ_{kl} are (1, 0.1, 0.1) and (0.1, 1, 0.01) in supervised setting and unsupervised setting, respectively.

Setting	Methods	Aberman et al. [6]				CMU Mocap [225]			
		Style Accuracy↑	Style FID↓	Geo Dis↓	Div↑	Style Accuracy↑	Style FID↓	Geo Dis↓	Div↑
	Real Motions	$0.997^{\pm 002}$	$0.002^{\pm 000}$	-	-	$0.997^{\pm 002}$	$0.002^{\pm 000}$	-	-
Motion-based (S)	Aberman et al. [6]	$0.547^{\pm 016}$	$0.379^{\pm 018}$	$0.804^{\pm 003}$	-	$0.445^{\pm 009}$	$0.508^{\pm 011}$	$0.910^{\pm 002}$	-
	Park et al. [108]	$0.891^{\pm 007}$	$0.038^{\pm 003}$	$0.531^{\pm 001}$	-	$0.674^{\pm 014}$	$0.136^{\pm 011}$	$0.663^{\pm 003}$	-
	Ours w/o latent	$0.932^{\pm 008}$	$0.022^{\pm 002}$	$0.463^{\pm 003}$	-	$0.879^{\pm 008}$	$0.046^{\pm 004}$	$0.636^{\pm 004}$	-
	Ours (V)	$\underline{0.935}^{\pm 007}$	$\underline{0.020}^{\pm 002}$	$\underline{0.426}^{\pm 003}$	-	$\underline{0.918}^{\pm 010}$	$\mathbf{0.028}^{\pm 003}$	$\underline{0.629}^{\pm 002}$	-
	Ours (A)	$\mathbf{0.945}^{\pm 007}$	$\mathbf{0.020}^{\pm 002}$	$\mathbf{0.344}^{\pm 002}$	-	$\mathbf{0.918}^{\pm 007}$	$\underline{0.031}^{\pm 003}$	$\mathbf{0.569}^{\pm 002}$	-
Label-based (S)	Park et al. [108]	$\mathbf{0.971}^{\pm 006}$	$\mathbf{0.013}^{\pm 001}$	$0.571^{\pm 002}$	$0.146^{\pm 009}$	$0.813^{\pm 010}$	$0.065^{\pm 007}$	$0.693^{\pm 004}$	$0.229^{\pm 019}$
	Ours w/o latent	$0.933^{\pm 009}$	$0.023^{\pm 002}$	$0.447^{\pm 002}$	$\mathbf{0.174}^{\pm 017}$	$0.882^{\pm 008}$	$0.053^{\pm 003}$	$\underline{0.611}^{\pm 003}$	$\mathbf{0.266}^{\pm 021}$
	Ours (V)	$\underline{0.946}^{\pm 007}$	$0.020^{\pm 002}$	$\underline{0.427}^{\pm 003}$	$\underline{0.134}^{\pm 016}$	$\mathbf{0.923}^{\pm 007}$	$\mathbf{0.027}^{\pm 003}$	$0.614^{\pm 002}$	$\underline{0.193}^{\pm 013}$
	Ours (A)	$0.942^{\pm 006}$	$\underline{0.019}^{\pm 001}$	$\mathbf{0.344}^{\pm 003}$	$0.050^{\pm 006}$	$\underline{0.915}^{\pm 005}$	$\underline{0.031}^{\pm 003}$	$\mathbf{0.571}^{\pm 003}$	$0.067^{\pm 005}$
Motion-based (U)	Jang et al. [109]	$\underline{0.833}^{\pm 010}$	$0.047^{\pm 004}$	$0.559^{\pm 003}$	-	$0.793^{\pm 009}$	$0.058^{\pm 004}$	$0.725^{\pm 004}$	-
	Ours w/o latent	$0.780^{\pm 014}$	$0.048^{\pm 003}$	$\underline{0.466}^{\pm 004}$	-	$0.761^{\pm 009}$	$0.082^{\pm 005}$	$\mathbf{0.645}^{\pm 003}$	-
	Ours (V)	$\mathbf{0.840}^{\pm 010}$	$\mathbf{0.036}^{\pm 003}$	$0.478^{\pm 004}$	-	$\mathbf{0.828}^{\pm 010}$	$\mathbf{0.052}^{\pm 004}$	$0.672^{\pm 003}$	-
	Ours (A)	$0.804^{\pm 011}$	$\underline{0.040}^{\pm 003}$	$\mathbf{0.441}^{\pm 003}$	-	$\underline{0.799}^{\pm 009}$	$\underline{0.056}^{\pm 003}$	$\underline{0.648}^{\pm 004}$	-
Prior-based (U)	Ours w/o latent	-	-	$\underline{0.431}^{\pm 003}$	$\underline{1.169}^{\pm 030}$	-	-	$\underline{0.626}^{\pm 001}$	$\underline{1.252}^{\pm 029}$
	Ours (V)	-	-	$\mathbf{0.418}^{\pm 003}$	$1.069^{\pm 028}$	-	-	$0.611^{\pm 003}$	$0.857^{\pm 024}$
	Ours (A)	-	-	$0.436^{\pm 004}$	$\mathbf{1.187}^{\pm 029}$	-	-	$\mathbf{0.641}^{\pm 002}$	$\underline{0.949}^{\pm 022}$

Table 5.2: Quantitative results on the Aberman et al. [6] and CMU Mocap test sets. ± indicates 95% confidence interval. **Bold** face indicates the best result, while underscore refers to the second best. (S) and (U) denote *supervised* and *unsupervised* setting. (V) VAE and (A) AE represent different latent models in Sec. 5.2.1

.

Data Processing. We mostly adopt the pose processing procedure in [3]. In short, a single pose is represented by a tuple of root angular velocity, root linear velocity, root height, local joint positions, velocities, 6D rotations [222] and foot contact labels, resulting in 260-D pose representation. Meanwhile, all data is downsampled to 30 FPS, augmented by mirroring, and applied with Z-nomalization.

5.3.1 Quantitative Results

Table 5.2 and Table 5.3 present the quantitative evaluation results on the test sets of Aberman et al. [6], CMU Mocap [225] and Xia et al. [7]. Note the latter two datasets are completely unseen to our latent stylization models. We generate results using motions in these three test sets as content, and randomly sample style motions and labels from Aberman et al. [6] test set. For fair comparison, we repeat this experiment 30 times, and report the mean value with a 95% confidence interval. We also consider the variants of our approach: non-latent stylization (*ours w/o latent*), using VAE (*Ours (V)*) or AE (*Ours (A)*) as latent model (See Sec. 5.2.1).

Overall, our proposed approach consistently achieves appealing performance on a variety of

Setting	Methods	Xia et al. [7]				
		Style Acc↑	Content Acc↑	Content FID↓	Geo Dis↓	Div↑
M-based (S)	Aberman et al. [6]	$0.364^{\pm 011}$	$0.318^{\pm 008}$	$0.705^{\pm 014}$	$0.931^{\pm 003}$	-
	Park et al. [108]	$0.527^{\pm 006}$	$0.441^{\pm 009}$	$0.381^{\pm 010}$	$\underline{0.698}^{\pm 001}$	-
	Ours w/o latent	$0.851^{\pm 012}$	$\underline{0.654}^{\pm 012}$	$0.258^{\pm 007}$	$0.707^{\pm 004}$	-
	Ours (V)	$\mathbf{0.934}^{\pm 006}$	$0.579^{\pm 016}$	$\underline{0.210}^{\pm 004}$	$0.716^{\pm 003}$	-
	Ours (A)	$\underline{0.926}^{\pm 008}$	$\mathbf{0.674}^{\pm 011}$	$\mathbf{0.189}^{\pm 005}$	$\mathbf{0.680}^{\pm 003}$	-
L-based (S)	Park et al. [108]	$0.796^{\pm 007}$	$0.311^{\pm 009}$	$0.507^{\pm 011}$	$0.770^{\pm 003}$	$0.175^{\pm 014}$
	Ours w/o latent	$0.843^{\pm 012}$	$\underline{0.655}^{\pm 013}$	$0.264^{\pm 008}$	$\underline{0.691}^{\pm 003}$	$\mathbf{0.281}^{\pm 032}$
	Ours (V)	$\mathbf{0.944}^{\pm 008}$	$0.606^{\pm 013}$	$\underline{0.208}^{\pm 005}$	$0.705^{\pm 003}$	$\underline{0.228}^{\pm 023}$
	Ours (A)	$\underline{0.933}^{\pm 011}$	$\mathbf{0.668}^{\pm 014}$	$\mathbf{0.193}^{\pm 005}$	$\mathbf{0.679}^{\pm 002}$	$0.095^{\pm 013}$
M-based (U)	Jang et al. [109]	$0.658^{\pm 009}$	$0.337^{\pm 017}$	$0.380^{\pm 011}$	$0.857^{\pm 004}$	-
	Ours w/o latent	$0.734^{\pm 014}$	$\underline{0.584}^{\pm 011}$	$0.272^{\pm 008}$	$\mathbf{0.721}^{\pm 003}$	-
	Ours (V)	$\mathbf{0.860}^{\pm 010}$	$0.499^{\pm 015}$	$\underline{0.221}^{\pm 006}$	$0.747^{\pm 004}$	-
	Ours (A)	$\underline{0.814}^{\pm 011}$	$\mathbf{0.588}^{\pm 010}$	$\mathbf{0.217}^{\pm 006}$	$\underline{0.735}^{\pm 003}$	-
P-based (U)	Ours w/o latent	-	$\mathbf{0.627}^{\pm 014}$	$0.246^{\pm 007}$	$\underline{0.708}^{\pm 003}$	$\mathbf{1.193}^{\pm 029}$
	Ours (V)	-	$0.579^{\pm 013}$	$\underline{0.239}^{\pm 006}$	$\mathbf{0.704}^{\pm 002}$	$0.874^{\pm 029}$
	Ours (A)	-	$\underline{0.586}^{\pm 015}$	$\mathbf{0.227}^{\pm 006}$	$0.736^{\pm 003}$	$\underline{0.978}^{\pm 026}$

Table 5.3: Quantitative results on the Xia et al. [7] test set.

Setting	Method	Ours wins
M-based (S)	Aberman et al. [6]	78.69%
	Park et al. [108]	73.67%
	Ours w/o latent	65.98%
L-based (S)	Park et al. [108]	73.06%
M-based (U)	Jang et al. [109]	58.92%

Table 5.4: Human evaluation results.

Methods	Runtime (ms)↓
Aberman et al. [6]	16.763
Park et al. [108](M)	37.247
Park et al. [108](L)	<u>16.329</u>
Jang et al. [109]	67.563
Ours (A)(M)	**4.760**

Table 5.5: Runtime comparisons.

applications across three datasets. In supervised setting, GAN approaches, such as Aberman et al. [6] and Park et al. [108], tend to overfit on one dataset and find difficult on scaling to other motions. For example, Park et al. [108] earns the highest achievement on *style recognition* on [6], as 97.1%, while underperforms on the other two unseen datasets, with style accuracy of 81.3% and 79.6%. Furthermore, these methods usually fall short in preserving content, as evidenced by the low content accuracy (31.8% and 44.1%) in Tab. 5.3. Jang et al. [109] is shown to be a strong unsupervised baseline; it gains comparable and robust performance on different datasets, which though still suffers from content preservation. On the contrary, our supervised and unsupervised models commonly maintain high style accuracy over 90% and 80% respectively, with minimal loss on content semantics. Among all variants, *latent* stylization improves the performance on almost all aspects, including generalization ability, with slight compromise on diversity. *Ours (V)* tends to own higher success rate of style transfer, while *ours (A)* typically outperforms on maintaining content (*i.e., Geo Dis* and *Content Accuracy*).

User Study. In addition, an user study on Amazon Mechanical Turk is conducted to perceptually evaluate our motion stylization results. 50 comparison pairs (on CMU Mocap [225]) between each baseline model and our approach, in the corresponding setting, are generated and shown to 4 users, who are asked to choose their favored one regarding realism and stylization quality. Overall, we collect 992 responses from 27 AMT users who have *master* recognition. As shown in Table 5.4, our method earns more user appreciation over most of the baselines by a large margin.

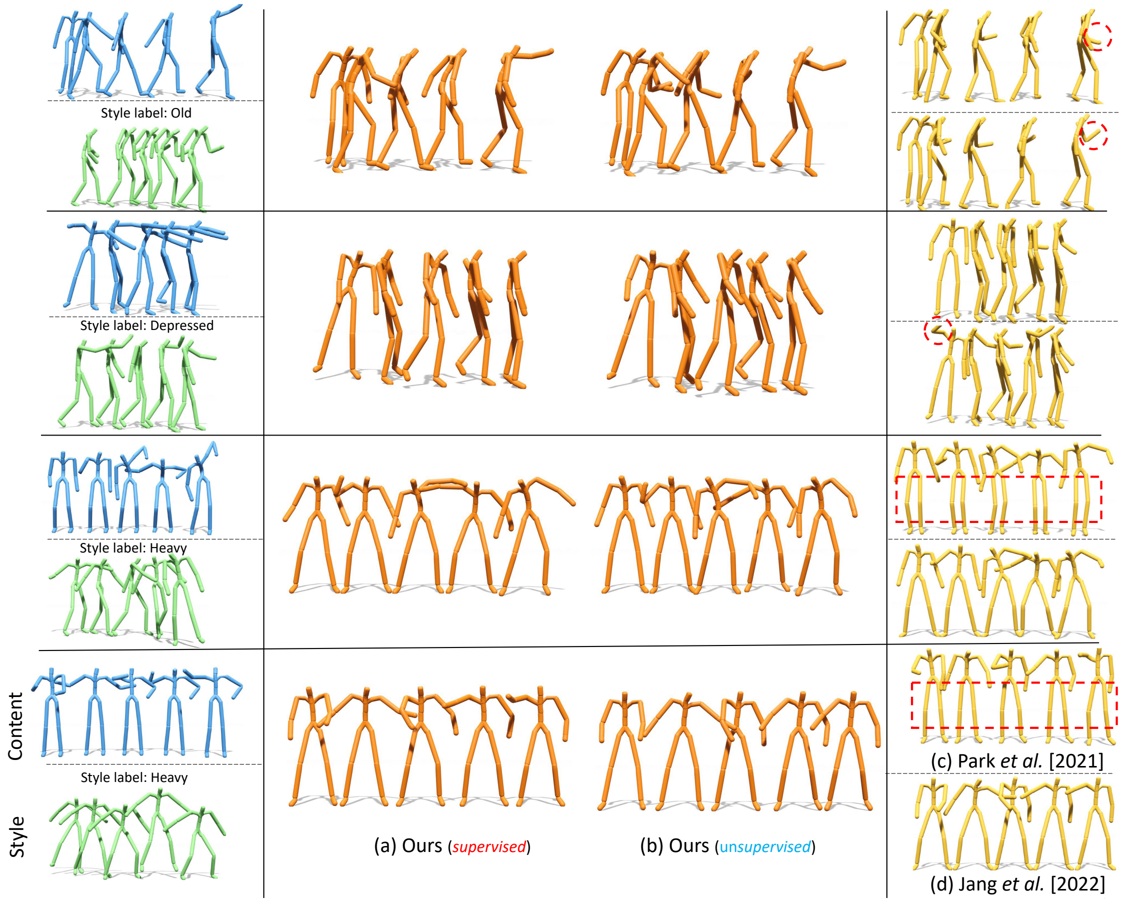


Figure 5.4: Qualitative comparisons of motion-based stylization. Given the style motion (green) and content motion (blue), we apply stylization using our methods (orange), Park et al. [108] (supervised), and Jang et al. [109] (unsupervised). The content motions in top two cases come from Aberman et al. [6], while the bottom two from CMU Mocap [225] test sets. Example artifacts are highlighted using red signs.

Efficiency. Table 5.5 presents the comparisons of average time cost for a single forward pass with 160-frame motion inputs, evaluated on a single Tesla P100 16G GPU. Previous methods apply style injection at each generator layers until the motion output, and usually involves computationally intensive operations such as multi-scale skeleton-based GCN and forward-loop kinematics. Benefiting from our latent stylization and lightweight network design, our model appears to be much faster and shows the potential for real time applications.

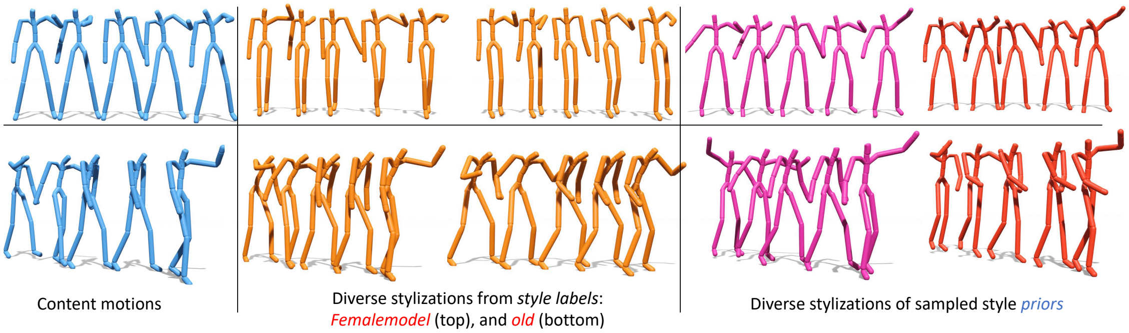

Content motions
Diverse stylizations from *style labels*:
Femalemodel (top), and *old* (bottom)
Diverse stylizations of sampled style *priors*

Figure 5.5: Two examples of diverse label-based stylization (middle) and prior-based stylization (right).

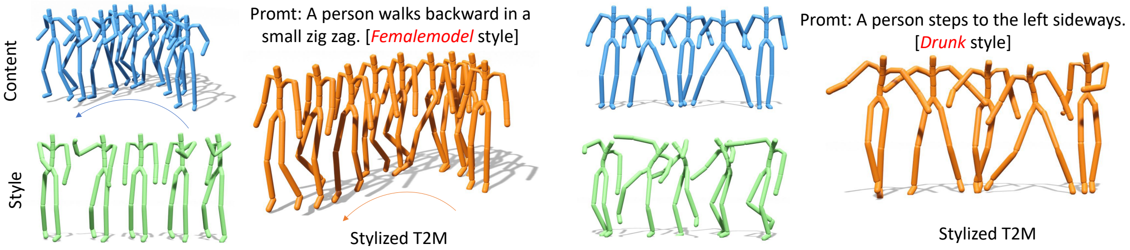

Content
Style
Promt: A person walks backward in a small zig zag. [*Femalemodel* style]
Stylized T2M
Promt: A person steps to the left sideways. [*Drunk* style]
Stylized T2M

Figure 5.6: Two stylized text2motion examples, by applying our method behind text2motion [3].

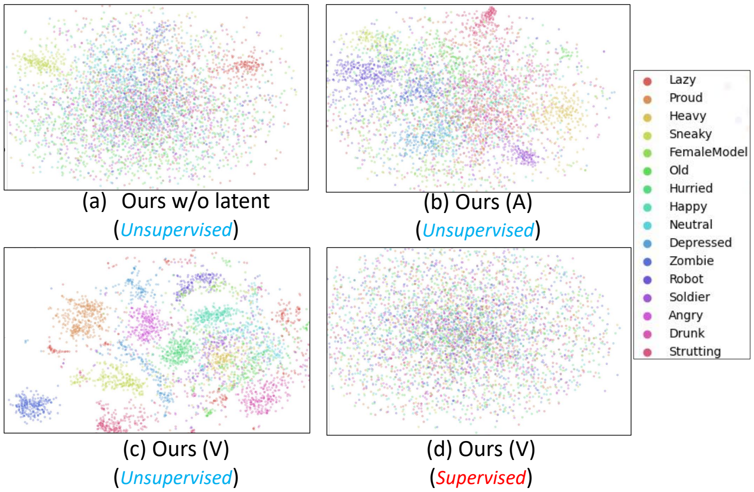
5.3.2 Qualitative Results

Figure 5.4 presents the visual comparison results on test sets of Aberman et al. [6] (top two) and CMU Mocap [225] (bottom two), in supervised (ours vs.Park et al. [108]) and unsupervised (ours vs. Jang et al. [109]) settings. For our model, we use *ours(V)* by default. In unsupervised setting, Jang et al. [109] has comparable performance on transferring style from style motion to content motion; but it sometimes changes the actions from content motion, as indicated in red circles. Supervised baselinePark et al. [108] follows similar trend. Moreover, the results of Park et al. [108] on CMU MocapAberman et al. [6] commonly fail to capture the style information from input style motion. This also agrees with the observation of limited generalization ability of GAN-based models in Tab. 5.2. Other artifacts such as unnatural poses ([6, 108]) and foot sliding(Jang et al. [109]) can be better viewed in the supplementary video. This can be partially attributed to the baselines directly applying global velocities of content motion for stylization results. In contrast, our approach show reliable performance on both maintaining content semantics and capturing style characteristics for robust stylization.

Diverse and Stochastic Stylization. Our approach allows for diverse label-based and stochastic prior-based stylization. As presented in Figure 5.5, for label-based stylization, taken one content motion and style label as input, our model is able to generate multiple stylized results with inner-class variations, *i.e.,* different manners of *old* man walking. On the other hand, our prior-based stylization can produce results with very distinct styles that are learned unsupervisedly from motion data. These styles are undefined, and possibly non-existing in training data.

Stylized Text2Motion. Text-to-motion synthesis has attracted significant interests in recent years [3, 5, 60, 62]; however, the results often exhibit limited style expression. Here in Fig. 5.6, we demonstrate the feasibility of generating stylistic human motions from the text prompt, by simply plugging our model behind an text2motion generator [3]. It is worth noting that the motions from [3] differs greatly from our learning data in terms of motion domain and frame rate (20 fps vs. ours 30 fps).

Style Code Visualization. Figure 5.7 displays the t-SNE 2D projection of our extracted style codes using four model variants, where each sample is color-coded according to its label. In the unsupervised setting, each style code is associated with global style features that are expected to be distinctive with respect to the style category. It can be observed that our *latent stylization* method produces clearer style clusters aligned with style labels compared to our non-latent method, with VAE-based latent model (*ours(V)*) performing the best. While in supervised setting, as discussed in Sec. 5.2.2, our approach learns label-invariant style features (Fig. 5.7 (d)); These style features may arise from individual and environmental factors.



(a) Ours w/o latent (*Unsupervised*)

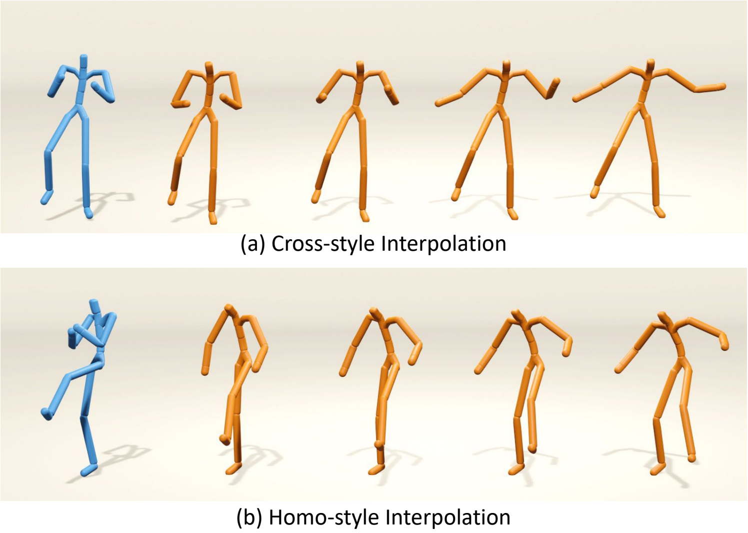
(b) Ours (A) (*Unsupervised*)

(c) Ours (V) (*Unsupervised*)

(d) Ours (V) (*Supervised*)

Figure 5.7: Style code visualization.

Interpolation. We present the results of interpolation in the respective style spaces learned unsupervisedly Fig. 5.8(a) and supervisedly Fig. 5.8(b). We are able to interpolate between styles from different labels in unsupervised setting. Specifically, two style codes are extracted from *sneaky* motion and *heavy* motion respectively. Then we mix these two style codes through linear interpolation, and apply them to stylize the given content motion. In supervised setting, the generator is conditioned on a specific style label. Here, we interpolate styles between two random style codes sampled from the prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Stylization results are produced conditioned a common style label, *heavy*. From Figure 5.8, we can observe the smooth transitions along the interpolation trajectory of two different style codes.


(a) Cross-style Interpolation

(b) Homo-style Interpolation

Figure 5.8: **Style Interpolation.** **(a)** Cross-style interpolation in unsupervisedly learned style space. Styles are interpolated between style codes of *sneaky* (left) and *heavy* (right) motions. **(b)** Homo-style interpolation in supervisedly learned style space. With style label *heavy* as condition input, styles are interpolated between two style codes that randomly sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. One key pose for each motion is displayed.

5.3.3 Component Analysis

Tab. 5.6 presents more quantitative results of our models on Aberman et al. [6] and Xia et al. [7] test sets. Specifically, we provide the ablation evaluations in both supervised (S) and unsupervised setting (U). For supervised setting, we conduct experiments on label-based stylization which also compares the diversity; and for unsupervised setting we adopt motion-based stylization. Note the base models are not necessarily our final models, here they are set only for reference.

S / U	λ_{cyc}	λ_{kl}	λ_{hsa}	Aberman et al. [6]			Xia et al. [7]		
				Style Acc↑	Geo Dis↓	Div↑	Style Acc↑	Content Acc↑	Content FID↓
S (base)	0.1	0.01	0.1	$0.937^{\pm 008}$	$0.415^{\pm 003}$	$0.153^{\pm 016}$	$0.913^{\pm 008}$	$\underline{0.669}^{\pm 013}$	$0.202^{\pm 006}$
			$\mathbf{0}$	$0.856^{\pm 011}$	$0.516^{\pm 003}$	$\mathbf{0.382}^{\pm 031}$	$0.829^{\pm 014}$	$0.525^{\pm 015}$	$0.230^{\pm 005}$
			0.5	$0.936^{\pm 008}$	$\underline{0.369}^{\pm 003}$	$0.091^{\pm 011}$	$0.924^{\pm 007}$	$\mathbf{0.706}^{\pm 010}$	$\mathbf{0.197}^{\pm 007}$
		0.001		$\mathbf{0.962}^{\pm 006}$	$0.429^{\pm 004}$	$0.125^{\pm 016}$	$\underline{0.933}^{\pm 009}$	$0.619^{\pm 014}$	$\underline{0.197}^{\pm 005}$
		0.1		$0.940^{\pm 008}$	$0.414^{\pm 004}$	$0.141^{\pm 015}$	$0.914^{\pm 009}$	$0.634^{\pm 013}$	$0.209^{\pm 005}$
	$\mathbf{0}$			$0.915^{\pm 007}$	$\mathbf{0.367}^{\pm 003}$	$0.129^{\pm 011}$	$0.836^{\pm 007}$	$0.605^{\pm 011}$	$0.213^{\pm 006}$
	0.01			$\underline{0.955}^{\pm 006}$	$0.419^{\pm 003}$	$0.107^{\pm 011}$	$\mathbf{0.957}^{\pm 007}$	$0.609^{\pm 011}$	$0.207^{\pm 006}$
	1			$0.880^{\pm 011}$	$0.423^{\pm 003}$	$\underline{0.302}^{\pm 026}$	$0.833^{\pm 011}$	$0.625^{\pm 013}$	$0.236^{\pm 006}$
U (base)	1	0.01	0.1	$\mathbf{0.804}^{\pm 011}$	$0.441^{\pm 003}$	-	$\mathbf{0.814}^{\pm 014}$	$0.588^{\pm 010}$	$0.217^{\pm 006}$
			$\mathbf{0}$	$0.742^{\pm 017}$	$0.511^{\pm 003}$	-	$0.725^{\pm 009}$	$0.522^{\pm 013}$	$0.239^{\pm 006}$
			0.01	$\underline{0.790}^{\pm 015}$	$0.489^{\pm 004}$	-	$0.761^{\pm 012}$	$0.567^{\pm 016}$	$0.224^{\pm 007}$
		0.1		$0.659^{\pm 018}$	$0.430^{\pm 004}$	-	$0.701^{\pm 014}$	$0.619^{\pm 013}$	$\mathbf{0.190}^{\pm 005}$
	$\mathbf{0}$			$0.778^{\pm 013}$	$0.434^{\pm 003}$	-	$0.753^{\pm 015}$	$\underline{0.477}^{\pm 012}$	$0.230^{\pm 006}$
	0.01			$0.669^{\pm 013}$	$\mathbf{0.388}^{\pm 003}$	-	$0.671^{\pm 015}$	$\mathbf{0.641}^{\pm 012}$	$\underline{0.206}^{\pm 006}$
	0.1			$0.739^{\pm 015}$	$\underline{0.420}^{\pm 004}$	-	$\underline{0.762}^{\pm 016}$	$\underline{0.619}^{\pm 014}$	$0.214^{\pm 007}$

Table 5.6: Effect of hyper-parameters of *ours (A)* on the Aberman et al. [6] and Xia et al. [7] test sets. ± indicates 95% confidence interval. **Bold** face indicates the best result, while underscore refers to the second best. (S) and (U) denote *supervised* and *unsupervised* setting. For (S), we present results of label-based stylization; and for (U), we present motion-based stylization.

Effect of λ_{hsa}. Homo-style alignment ensures the style space of the sub-clips from one motion sequence to be close to each other; it is an important self-supervised signal in our approach. Our experiment also shows a sharp drop of performance, on almost all criteria, while homo-style alignment is disabled ($\lambda_{hsa} = 0$). Increasing the weight of homo-style commonly helps style modeling (style accuracy) and content preservation (content accuracy, FID), which however also comes with lower diversity. A common observation is that the performance on style and content always contradicts with the diversity. It could be possibly attributed to the inherently limited diversity in our training datasetAberman et al. [6], which is collected by one person performing several styles.

Effect of λ_{kl}. λ_{kl} weighs how much the overall style space aligns with the prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Smaller λ_{kl} usually increases the capacity of the model exploiting styles, which on the other hand deteriorate the performance on content maintenance and diversity.

Effect of λ_{cyc}. Cycle reconstruction constraint plays an important role in unsupervised setting. In supervised setting, strong cycle reconstruction constraint is detrimental to style modeling. In contrast, while learning unsupervisedly, strengthening the cycle constraint enhances the performance

Model	Style Acc↑	Style FID↓	Geo Dis↓	Foot Skating↓
Ours (S)	$\mathbf{0.945}^{\pm 007}$	$\mathbf{0.020}^{\pm 002}$	$0.344^{\pm 002}$	$\mathbf{0.130}^{\pm 001}$
Ours *w/o* GMP (S)	$0.942^{\pm 006}$	$0.021^{\pm 002}$	$\mathbf{0.341}^{\pm 003}$	$0.141^{\pm 001}$
Ours (U)	$\mathbf{0.840}^{\pm 010}$	$\mathbf{0.036}^{\pm 003}$	$\mathbf{0.478}^{\pm 004}$	$\mathbf{0.102}^{\pm 001}$
Ours *w/o* GMP (U)	$0.817^{\pm 013}$	$0.038^{\pm 003}$	$0.478^{\pm 003}$	$0.116^{\pm 001}$

Table 5.7: Effect of global motion prediction (GMP, Sec. 5.2.2) on the [6] test set. ± indicates 95% confidence interval. **Bold** face indicates the best result. (S) and (U) denote *supervised* and *unsupervised* setting. Results of motion-based stylization is presented.

on style transferring, and at the same time compromises the preservation of content.

Effect of Global Motion Prediction. To quantify the impact of global motion prediction (shorthanded GMP, Sec. 5.2.2), we compare our full model with the variant without the GMP module (ours w/o GMP), where global motions are derived directly from source content input as in previous works. Furthermore, we introduce a *foot skating metric*, measuring the average velocity of foot joints on the XZ-plane during foot contact. Table 5.7 presents the motion-based results on [6] test set. On both supervised and unsupervised settings, our proposed GMP effectively mitigates foot skating issue. Notably, GMP exhibits a more pronounced effect in the unsupervised setting, leading to an approximate 2% improvement in style accuracy.

5.3.4 Limitations and Failure Cases

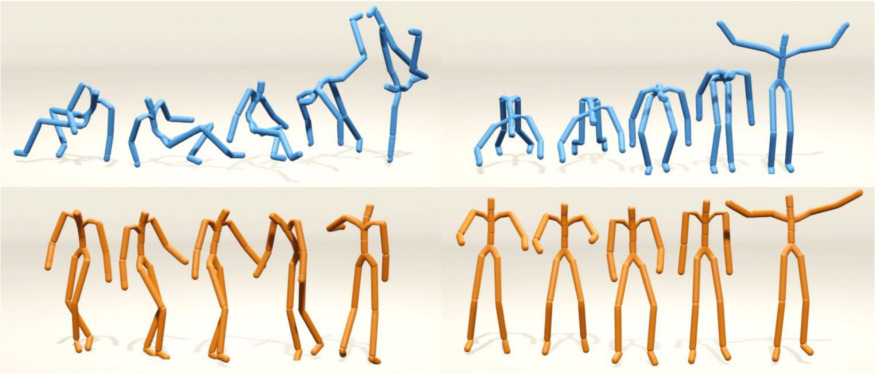


Figure 5.9: **Failure cases.** Top row shows content motion; bottom row shows our corresponding results. Stylization results of breaking dance motion (left) and push-up motion (right) using *happy* style label are displayed.

Firstly, our model may encounter difficulties when the input motion substantially deviates from our training data. Figure 5.9 presents two failed stylization results on rare content actions, *i.e.,* breaking dance and push-up. Given that our model has only seen standing motions during training, it commonly fails to reserve the lower-body movements in these two cases. Interestingly, our model can still retain the general motions of upper-body.

Secondly, the underlying reason for different performance of ours(V) and ours(A) on for example, diversity, style and content accuracy, remains unclear.

Lastly, certain styles are inherently linked to specific content characteristics, particularly within the datasets of [6, 7]. For instance, styles like old, depressed and lazy typically relate to slow motions, while happy, hurried, angry motions tend to be fast. As our stylization process aims to preserve content information, including speed, there could be contradictions with these style attributes. For instance, stylizing an slow motion with a hurried style might not yield an outcome resembling a hurried motion. We acknowledge this aspect for potential exploration in future studies.

5.4 Conclusion

This chapter looks into the problem of 3D human motion stylization, with particular emphasis on generative stylization in the neural latent space. Our approach learns a probabilistic style space from motion latent codes; this space allows style sampling for stylization conditioned on reference style motion, target style label, or free-form novel re-stylization. Experiments on three mocap datasets also demonstrate other merits of our model such as better generalization ability, flexibility in style controls, stylization diversity and efficiency in forward pass.

Chapter 6

Reciprocal Generation of Motions and Texts

The aim of this chapter is to investigate the bi-directional ties between 3D human full-body motions and their language descriptions, as illustrated in Fig. 6.1. To tackle the existing challenges, we propose the use of motion tokens, a discrete and compact motion representation. This provides one level playing ground when considering both motions and text signals, as the motion and text tokens, respectively. Moreover, our motion2text module is integrated into the inverse alignment process of our text2motion training pipeline, where a significant deviation of synthesized text from the input text would be penalized by a large training loss. Finally, the mappings in-between the two modalities of motions and texts are facilitated by adapting the neural model for machine translation (NMT) to our context. This autoregressive modeling of the distribution over discrete motion tokens further enables the non-deterministic production of pose sequences, of variable lengths, from an input text. Our approach is flexible, and could be used for both text2motion and motion2text tasks. This chapter has been published as [5]. The related data, pre-trained model, and code are publicly available: https://ericguo5513.github.io/TM2T/.

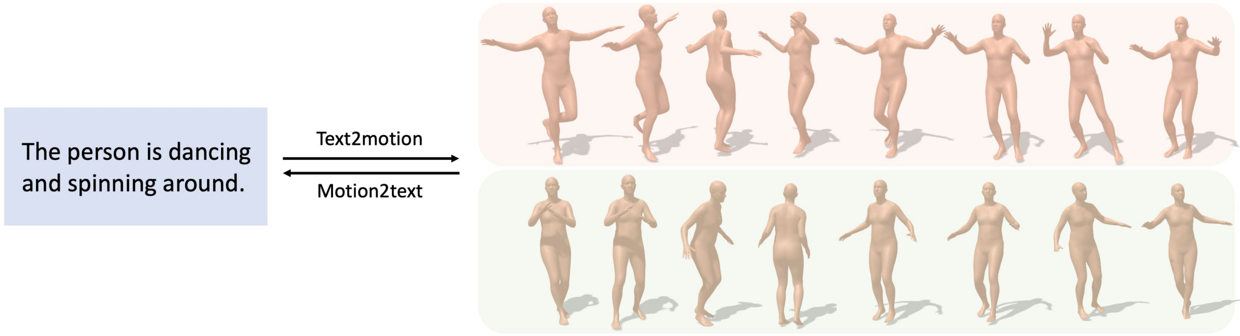


Figure 6.1: An illustration of our bidirectional TM2T approach that captures the interplay between text (left) and 3D motion (right) through the text2motion and motion2text modules.

6.1 Introduction

The interplay of vision and language is important in our daily life and social functions. It has motivated considerable research progresses in related topics such as image or video captioning [24, 25], and language grounded generation of images or videos [22, 23, 145]. On the other hand, when coming to human motion analysis, the connections between visual and textural aspects of human motions are much less studied. Existing efforts primarily focus on unidirectional mapping of either motion captioning (motion2text) [48, 127] or language grounded motion generation (text2motion) [12, 59, 37], with only two [39, 38] exploring the integration of visual 3D motions and their textural descriptions. However, both studies tend to produce static pose sequences when motion lengths are longer than 3-4 seconds. Both requires as input the initial pose & target motion length. They are also deterministic methods. That is, each of them always generates the same motions from a given text script. The first phenomenon of lifeless motions could be largely attributed to the direct use of raw 3D poses as their motion representation, which is unnecessarily redundant and yet fails to capture the local contexts of the underlying motion dynamics. The second issue is rooted in their deterministic motion generation processes, that are in contrary to our daily experiences, where multiple distinct motion styles often exist for a character to perform under a same textural script. The conditioning on initial state and target length further imposes strict constraint toward being practically feasible.

The aim of this chapter is to investigate the bi-directional ties between 3D human full-body motions and their language descriptions, as illustrated in Fig. 6.1. Given the asymmetric nature of the two underlying tasks, where text2motion is typically a much harder problem than the reciprocal task, motion2text, our primary focus is text2motion, with a secondary emphasis on motion2text. It is worth noting that in our approach, the module (also called motion2text for simplicity) developed for motion2text task, is also utilized as an integral part of our text2motion training process, referred to as inverse alignment in Fig. 6.2(c). Empirical evidences suggest the benefit of this strategy in improving our performance for the text2motion problem. To address the lifeless motion issue, we introduce motion token, a compact and semantically rich representation for 3D motions. This is achieved by adapting the deep vector quantization [65] in our context to learn a spatial-temporal codebook from the 3D pose sequences in the training set, with each entry in the codebook describing a particular kind of motion segments. 3D motions are then reconstructed by decoding the compositions of a list of codebook entries. This way, a 3D human motion is represented as a list of motion tokens (i.e. discrete indices to the codebook entries), each encoding its local spatial-temporal context. This discrete representation also facilitates the follow-up neural machine translators (NMTs) [226, 227] to construct mappings between the stream of motion tokens from the motion side, and the stream of text tokens from the language side. Furthermore, our proposed approach is able to explicitly model the underlying distribution of 3D motions conditioned on texts, instead of regressing the mean motions as in previous works [12, 59, 37, 38, 39], thus allows non-deterministic text2motion generation.

Our main contributions can be summarized as follows: (i) a motion token representation that compactly encodes 3D human motions. Together with the other key ingredients, including NMT mappings in-between the motion-token and text-token sequences, the motion2text-based inverse alignment, as well as the distribution sampling for non-deterministic predictions, our approach is capable of generating 3D motions (i.e. pose sequences) that are distinct in their lengths and styles, visually pleasing, and importantly, semantically faithful to the same input script. Our approach is also flexible, in that it can be use for both text2motion and motion2text tasks. (ii) Extensive empirical evaluations over two motion-language benchmark datasets demonstrate the superior performance of our approach over a variety of state-of-the-art methods when examined on each of the two tasks.

6.2 Our Approach

In what follows, we first detail how discrete motion tokens are obtained from raw 3D motions via vector quantization in Sec. 6.2.1. Based on this new motion representation, autoregressive NMT networks are used for modeling the bi-modal mappings of motion2text (Sec. 6.2.2) and text2motion (Sec. 6.2.3), with inverse alignment elaborated in Sec. 6.2.3.

6.2.1 Motion Tokens

We pre-train a latent quantization model on 3D human motions as presented in Fig. 6.2 (a). Given the pose sequence $\mathbf{m} \in \mathbb{R}^{T \times D_p}$, where T denotes the number of poses and D_p pose dimension, a series of 1D convolutions are applied along the time (i.e. 1st) dimension that yields latent vectors $\hat{\mathbf{b}} \in \mathbb{R}^{t \times d}(t < T)$ with d being number of convolution kernels. This process could be written as $\hat{\mathbf{b}} = \mathrm{E}(\mathbf{m})$.

Then, $\hat{\mathbf{b}}$ is transformed to a collection of codebook entries $\mathbf{b_q} \in \mathbb{R}^{t \times d}$ through discrete quantization. Specifically, the learnable codebook $\mathcal{B} = \{\mathbf{b}\}_{k=1}^{K} \subset \mathbb{R}^d$ consists of K latent embedding vectors with each a d-dimensional vector. The process of quantization $\mathrm{Q}(\cdot)$ is operated by replacing each row vector $\hat{\mathbf{b}}_i \in \mathbb{R}^d$ in $\hat{\mathbf{b}}$ with its nearest codebook entry $\mathbf{b}_k$ in $\mathcal{B}$, defined as

$$\mathbf{b_q} = \mathrm{Q}(\hat{\mathbf{b}}) := \left(\mathrm{argmin}_{\mathbf{b}_k \in \mathcal{B}} \|\hat{\mathbf{b}}_i - \mathbf{b}_k\|\right) \in \mathbb{R}^{t \times d}. \tag{6.1}$$

A following de-convolutional decoder D projects $\mathbf{b_q}$ back to the 3D motion space as a pose sequence, $\hat{\mathbf{m}}$. Now, the entire process can be formulated as

$$\hat{\mathbf{m}} = \mathrm{D}(\mathbf{b_q}) = \mathrm{D}(\mathrm{Q}(\mathrm{E}(\mathbf{m}))). \tag{6.2}$$

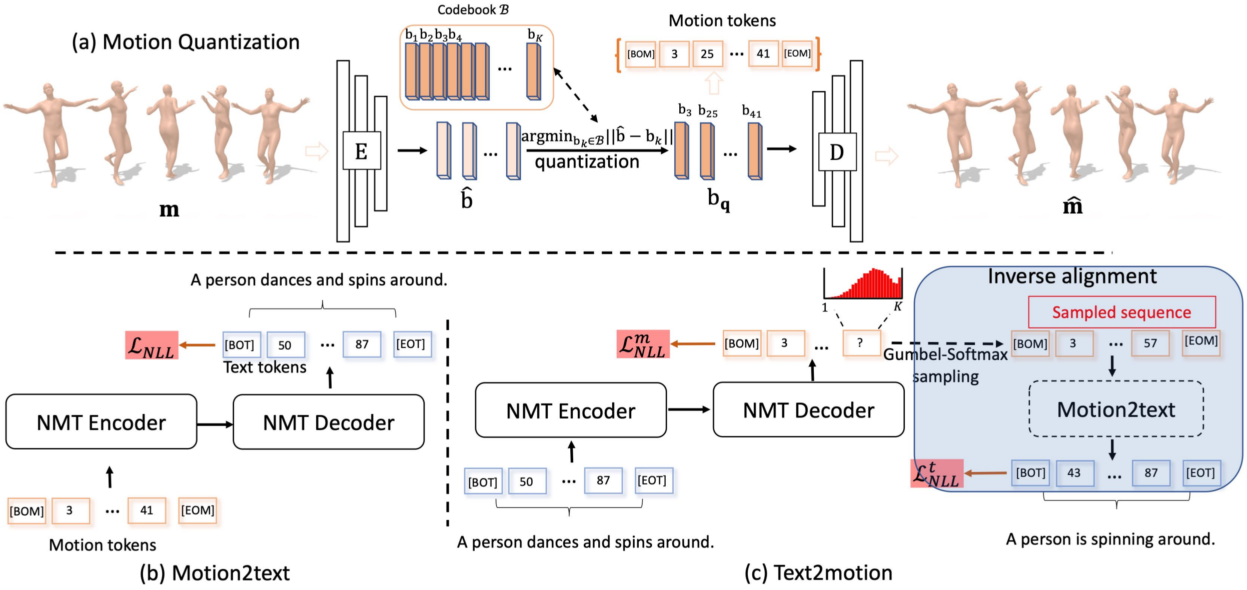


Figure 6.2: **Approach overview.** (a) A 1D CNN based latent quantization model is firstly learned to reconstruct training motions. After training, a motion can be subsequently converted to a tuple of discrete motion tokens (i.e., codebook-indices). [BOM] and [EOM] are indicators of start and end added in a motion token sequence. (b-c) Mappings between motion and text tokens are modeled by autoregressive NMT networks and optimized by maximizing the log-likelihood of the targets ($\mathcal{L}_{NLL}$ and $\mathcal{L}^m_{NLL}$). (c) While training text2motion, motion tokens sampled from the resulting discrete distributions are inversely mapped to the text space via the learned motion2text model. Loss $\mathcal{L}^t_{NLL}$ penalizes the inverse alignment error. Finally, the 3D pose sequence is obtained by decoding motion tokens via the decoder D in (a).

This is trained via a reconstruction loss combined with embedding commitment loss terms that encourage latent alignment and stabilize training process:

$$\mathcal{L}_{vq} = \|\hat{\mathbf{m}} - \mathbf{m}\|_1 + \|\text{sg}[\text{E}(\mathbf{m})] - \mathbf{b_q}\|_2^2 + \beta\|\text{E}(\mathbf{m}) - \text{sg}[\mathbf{b_q}]\|_2^2, \tag{6.3}$$

where sg[·] denotes the stop-gradient operation, and β a weighting factor. Straight-through gradient estimator [65] is employed to allow gradient backpropagation through the non-differentiable quantization operation in Eq.(6.1) that simply copies the gradients from the decoder D to the encoder E.

During inference, a pose sequence $\mathbf{m} \in \mathbb{R}^{T \times D_p}$ can be represented as a sequence of discrete codebook-indices $s \in \{1, ..., |\mathcal{B}|\}^t$ (namely *motion tokens*) of quantized embedding vectors $\mathbf{b_q}$,

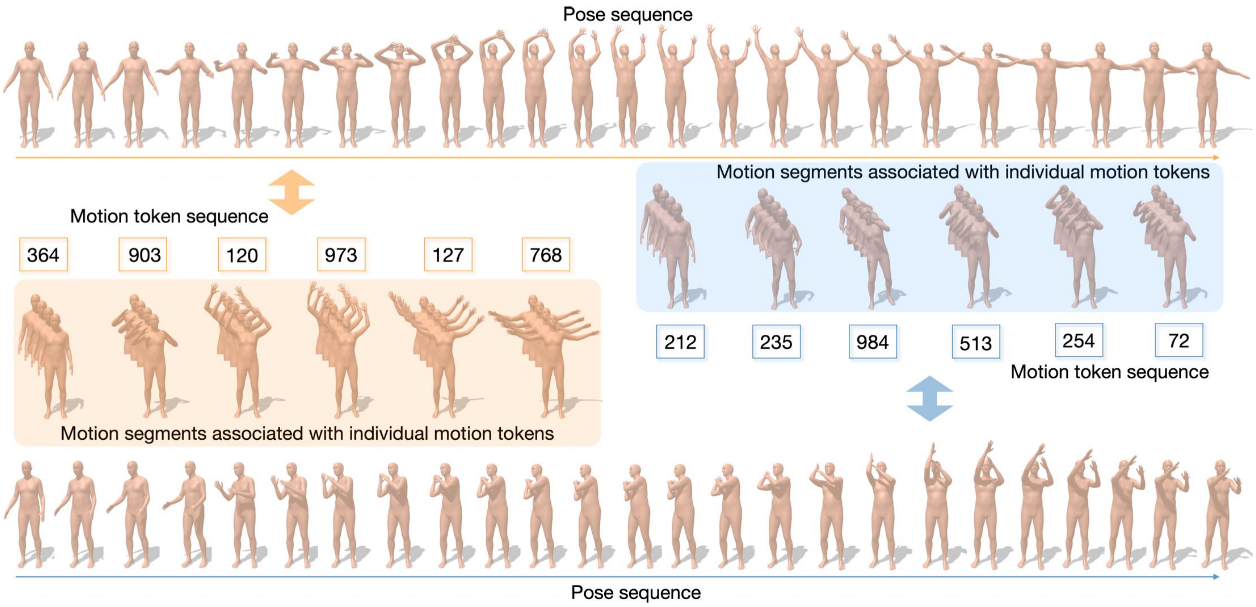

Pose sequence
Motion token sequence
364
903
120
973
127
768
Motion segments associated with individual motion tokens
Motion segments associated with individual motion tokens
212
235
984
513
254
72
Motion token sequence
Pose sequence

Figure 6.3: Exemplar results of motion tokens (middle) and their corresponding pose sequences (top and bottom). Here two 24-frame pose sequence examples are presented; each is reconstructed from a motion token sequences of size 6. Each motion token is associated with a specific local spatial-temporal context, visualized in 4-frame motions.

where $s_i = k$ such that $(\mathbf{b_q})_i = \mathbf{b}_k$. By mapping motion tokens back to their corresponding codebook entries $\mathbf{b_q} = (\mathbf{b}_{s_i})$, human poses are then readily recovered using decoder $\hat{\mathbf{m}} = \mathrm{D}(\mathbf{b_q})$. [BOM] and [EOM] are respectively added to the start and end of a motion token sequence as boundary indicators.

Motion Token Contexts.

With vector quantization, each motion token is associated with a particular type of motion contexts, thus a 3D motion can be regarded as a meaningful composition of motion tokens. We decode each entry in the learned codebook $\mathcal{B}$ using decoder D and get 4-frame motion segments ($t = \frac{T}{4}$ in our setting) that reflect the contexts associated with individual motion tokens. Fig. 6.3 presents two raw pose sequences and their motion token representations, as well as the associated motion segments. We can observe that, with global dependencies maintained in motion token sequences, each motion token successfully captures the spatial-temporal characteristics in local contexts.

6.2.2 Learning Motion2text

Given tokenized motion representation, we are able to efficiently build mapping from human motions to texts using NMT models such as Transformer [226]. Assume the target is a sequence of text tokens $x \in \{1, ..., |\mathcal{V}|\}^N$, where $\mathcal{V}$ is the word vocabulary and N number of words in the description. As described in Fig. 6.2 (b), source motion tokens are fed into Transformer encoder and then the decoder predicts the probability distribution of possible discrete text tokens at each step $p_\theta(x|s) = \prod_i p_\theta(x_i|x_{<i}, s)$. Thus the training goal is to maximize the log-likelihood of the target sequence,

$$\mathcal{L}_{NLL} = -\sum_{i=0}^{N-1} \log p_\theta(x_i|x_{<i}, s). \tag{6.4}$$

6.2.3 Learning Text2motion

Similarly, generating motions from language description can be modeled as autoregressive next-token predictions conditioned on textual inputs. Here we investigate two NMT models as our backbone: attentive GRU and Transformer, and examine our idea of *inverse alignment* on GRU-based model. Since Transformer is typically trained with full teacher force, optimizing the Transformer-based text2motion with inverse alignment is extremely complicated. In other words, every time when generating the density function of next motion token, we need to input the whole history to the Transformer decoder and feed forward. As a result, to sample a complete motion token sequence, the computational (or optimization) graph will be extremely high. Therefore, we specifically introduce the procedure of using GRU based model as an example.

As is shown in Fig. 6.2 (c), firstly, a bi-directional GRU (i.e., NMT Encoder) models the temporal dependencies in language $x \in \{1, ..., |\mathcal{V}|\}^N$, and produces sentence feature vector $\mathbf{s} \in \mathbb{R}^{d_l}$ as well as word feature vectors $\mathbf{w} \in \mathbb{R}^{N \times d_l}$, with d_l denoting the dimensionality of hidden vectors. The NMT decoder, modeled as attention-based GRU, processes $\mathbf{s}$ and $\mathbf{w}$ and predicts the probability distribution over discrete motion tokens $\{1, ..., |\mathcal{B}|\}$ autoregressively. In particular, GRU decoder is initialized by sentence vector $\mathbf{s}$, and then takes the attention vector $\mathbf{w}_{att}$ together with motion token as input at each time step. The attention vector $\mathbf{w}_{att}^t$ at time t is obtained via

$$\mathbf{Q} = \mathbf{h}_{t-1}\mathbf{W}^Q, \mathbf{K} = \mathbf{w}\mathbf{W}^K, \mathbf{V} = \mathbf{w}\mathbf{W}^V, \tag{6.5}$$

$$\mathbf{w}_{att}^t = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_{att}}}\right)\mathbf{V}, \tag{6.6}$$

where $\mathbf{h}_{t-1} \in \mathbb{R}^{d_h}$ is previous hidden state in decoder, $\mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d_l \times d_{att}}$ and $\mathbf{W}^Q \in \mathbb{R}^{d_h \times d_{att}}$ are trainable weights with d_h and d_{att} denoting the dimension of hidden unit and attention vector respectively. During generation, motion tokens are sampled from predicted distribution $p_\phi(s_i|s_{<i}, x)$ recursively until the end token (i.e., [EOM]) comes with maximum probability.

Inverse Alignment.

Here we re-utilize the motion2text model in Sec. 6.2.2 to further align the semantics between texts and generated motions. In detail, motion token sequence $\hat{s}$ is sampled from the approximated distribution $p_\phi(s|x)$, which is taken as input to the learned motion2text model and mapped to language tokens x with probability $p_\theta(x|\hat{s})$. Note motion2text model is no longer updated here. However, sampling from discrete distribution is non-differentiable that does not allow the gradients back-propagating to the text2motion encoder and decoder. We instead resort to *Gumbel-Softmax* reparameterization trick [228] to approximate the discrete sampling process. As the temperature τ of Gumbel-Softmax approaches 0, the resulting Gumbel-Softmax distribution becomes identical to the discrete distribution $p_\phi(s_i|s_{<i}, x)$ and the sampled vectors become one-hot.

In summary, the final training objective turns to be

$$\mathcal{L} = -\left(\sum_{i=0}^{K-1} \log p_\phi(s_i|s_{<i}, x) + \sum_{i=0}^{N-1} \log p_\theta(x_i|x_{<i}, \hat{s})\right). \tag{6.7}$$

3D pose sequences can finally be obtained by decoding sampled motion tokens $\hat{s}$ using quantization decoder D as described in Sec. 6.2.1. With discrete motion tokens and autoregressive modeling, variable motion lengths are implicitly modeled by text2motion, that the NMT model particularly learns to predict the end token i.e. [EOM] with maximum probability as signal of termination. Moreover, our proposed approach is easy to train, and does not suffer from the known shortcomings in GAN and VAE such as ”mode collapse”.

6.3 Experiments

Extensive experiments are conducted to evaluate our learned motion2text (Sec. 6.3.5) and text2motion mapping models(Sec. 6.3.6).

6.3.1 Datasets

Two 3D human motion-language datasets are considered for evaluation:

- *HumanML3D* [3] is a large 3D human motion dataset that covers a broad range of human actions such as locomotion, sports, and dancing. It consists of 14,616 motions and 44,970 text descriptions. Each motion clip comes with at least 3 descriptions. Motions are re-scaled to 20 frames per second (FPS), resulting in duration ranges from 2 to 10 seconds.
- *KIT Motion-Language* [43] contains 3,911 3D human motion clips and 6,278 text descriptions. For each motion, the corresponding number of text descriptions ranges from one to four. Following [12, 59], these pose sequences are all sub-sampled to 12.5 FPS.

Both datasets are split into training, testing and validation sets with ratio of 0.8:0.15:0.05, which are further augmented by mirroring motions and replacing corresponding words in their text descriptions (e.g., ’left’→’right’).

6.3.2 Metrics

Besides traditional measurements, we also manage to evaluate the correspondences between motion and language using deep multimodal features. In particular, we train a simple framework that engages a motion feature extractor and a text feature extractor under contrastive assumption, that learn to produce geometrically closed feature vectors for matched text-motion pairs, and vice versa.

R-Precision and Multimodal Distance are proposed to gauge how well a text and a motion are semantically aligned. Take the evaluation of motion2text mapping for an example. For each generated description, we take its corresponding motion as well as 31 randomly selected mismatched motions from the test set as a motion pool. With text and motion feature extractors available, Euclidean distances between the description feature and each motion feature in the pool are calculated and ranked. The ground truth entry falling into the top-k (k=1,2,3) candidates is regarded as a successful retrieval. Then we count the average accuracy at top-k places, known as *top-k R-precision*. Meanwhile, *multimodal distance* is computed as the average Euclidean distance between text feature of each generated description and motion feature of its corresponding motion in the test set. Computing R-precision and multimodal distance for text2motion mapping is analogically carried out except generated motions and ground truth description are accordingly used.

Overall, an extensive set of metrics including Bleu [229], Rouge [230], Cider [126], BertScore [231], R Precision and multimodal distance are adopted to quantitatively measure the performance of our motion2text mapping. For evaluation of non-deterministic text2motion mapping, we primarily follow [2] which uses Frechet Inception Distance (FID), diversity and multimodality, and our complementary metrics, R precision and multimodal distance.

6.3.3 Implementation Details

Our framework is implemented by PyTorch. Our codebook $\mathcal{B}$ contains 1024 1024-dimentional embedding vectors. Encoder and decoder in motion quantization are two 1D convolutional/upsampling layers with resblocks. Weighting factor β is set to 1. Transformers for motion2text and text2motion have 4 and 3 attention layers respectively, both with 8 attention heads with 512 hidden size. The GRU based text2motion model have encoder with hidden size of 512 while the decoder is modeled as 1-layer GRU with hidden size of 1024. This GRU model is trained with teacher force ratio of 0.4. Bi-directional GRUs with hidden size 1024 are used for motion & text feature extractors. Adam is used for all experiments with learning rate of 0.0002. We use the codebase NLPEval [1] to calculate linguistic metrics (e.g., Bleu, Rouge). In text2motion, we use pre-trained 300-dimensional word

[1] https://github.com/Maluuba/nlg-eval

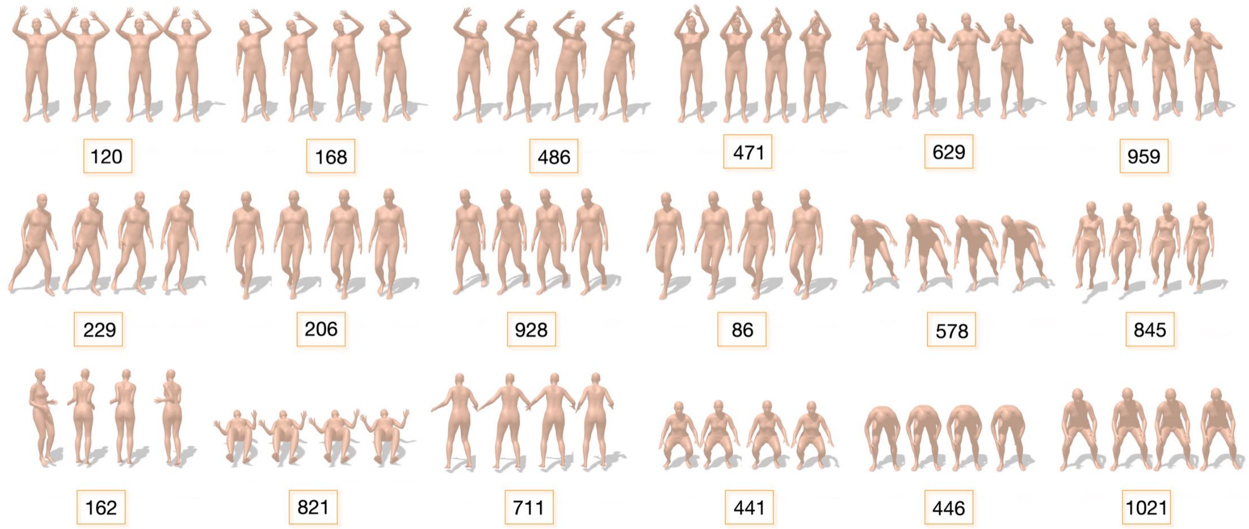


Figure 6.4: Exemplar motion tokens and their associated local spatial-temporal contexts, visualized in 4-frame motion segments.

embedding vectors from GloVe [223].

Pose Representation. For pose representation, we extract root angular velocity, root linear velocities, root height, local joint positions, velocities, 6D rotations [222] and foot contacts from raw motions as in [221]. This results in 263 and 251 dimensional pose vectors for HumanML3D (22-joint skeleton) and KIT-ML (21-joint skeleton) dataset respectively. After all, Z-score normalization is applied to both datasets.

During training motion quantization model, to mitigate foot sliding phenomenon, the decoder D is asked to additionally predict foot contact information which is not provided to the encoder E. We also scale the magnitude of root angular velocity, root linear velocities, root height and foot contacts by a value of 5 to amplify their importance. To improve the robustness of our approach, during learning motion2text and text2motion models, we randomly cutting off 0 to 4 frames at the head or tail of pose sequences, which increases the data variance while not scarifying the quality.

6.3.4 Motion Token Contexts

To visualize the local context associated with each motion token, we decode individual tokens using the quantization decoder D, that produces short 4-frame motion segments for each token. In Fig. 6.4, we present a gallery of learned motion tokens, as well as the motion segments reflecting their contexts. Note given a tuple of motion tokens, the quantization decoder D learns to naturally mingle their local context with seamless transitions, rather than simply concatenating their motion segments.

6.3.5 Evaluation of Motion-to-text Translation

We adopt RAEs [38] and SeqGAN [48] as our baseline methods; RAEs [38] learns a shared embedding space for language and human motions via two recurrent autoencoders, while SeqGAN [48] combines recurrent sequence-to-sequence model with a discriminator that judges whether a sentence is real or not. We further equip the vanilla RNN model in Seq2Seq [39] with late attention as another strong baseline (termed as Seq2Seq(Att)). A variant of our method not using motion tokens (ours w/o MT) is also engaged to analyze the role of motion token. Note that grammatical tense and plural of words are neglected in our setting in order to ease the learning process. Descriptions are produced using beam search strategy with size of 2 throughout all experiments.

Quantitative Analysis.

Table 6.1 presents the quantitative evaluation results of motion to language mapping on HumanMl3D and KIT-ML test sets. The R precision and multimodal distance of **real** descriptions are provided for reference.

The high R precision of real descriptions also evidences the effectiveness of learned motion & text feature extractors and R precision metric. Overall, our method clearly outperforms all baseline methods over a large margin on all datasets and metrics. RAEs [147] suffers from limited capability on modeling long-term dependencies between 3D motion and language, thus resulting in

Datasets	Methods	R Precision↑			MM Dist↓	Bleu@1↑	Bleu@4↑	Rouge↑	Cider↑	BertScore↑
		Top 1	Top 2	Top 3						
Human ML3D	**Real Desc**	0.523	0.725	0.828	2.901	-	-	-	-	-
	RAEs [38]	0.100	0.188	0.261	6.337	33.3	10.2	37.5	22.1	10.7
	Seq2Seq(Att)	0.436	0.611	0.706	3.447	51.8	17.9	46.4	58.4	29.1
	SeqGAN [48]	0.332	0.457	0.532	4.895	47.8	13.5	39.2	50.2	23.4
	Ours w/o MT	0.483	0.678	0.783	3.124	59.5	21.2	47.8	68.3	34.9
	Ours	**0.516**	**0.720**	**0.823**	**2.935**	**61.7**	**22.3**	**49.2**	**72.5**	**37.8**
KIT-ML	**Real Desc**	0.399	0.618	0.793	2.772	-	-	-	-	-
	RAEs [38]	0.034	0.063	0.106	9.364	30.6	0.10	25.7	8.00	0.40
	Seq2Seq(Att)	0.293	0.450	0.555	4.455	34.3	9.30	36.3	37.3	5.30
	SeqGAN [48]	0.109	0.345	0.425	6.283	3.12	5.20	32.4	29.5	2.20
	Ours w/o MT	0.284	0.466	0.595	3.979	42.8	14.7	39.9	60.1	18.9
	Ours	**0.359**	**0.561**	**0.668**	**3.298**	**46.7**	**18.4**	**44.2**	**79.5**	**23.0**

Table 6.1: Quantitative evaluation results for motion-to-text translation on HumanML3D and KIT-ML test sets. For each metric, the best score is highlighted in **bold**, with the second best highlighted using underscore.

low R precision and linguistic evaluation scores. This is mitigated by introducing attention mechanism in Seq2Seq(Att) or adversarial learning in SeqGAN, which effectively lifts the top-1 R precision up by more than 20% on HumanML3D and 10% on KIT-ML test sets. By utilizing motion token in our framework (ours), we can observe a obvious jump on both linguistic quality (i.e., Bleu, BertScore) and motion-retrieval precision (i.e., R precision) of generated language descriptions, which is surprisingly approaching the scores of real descriptions.

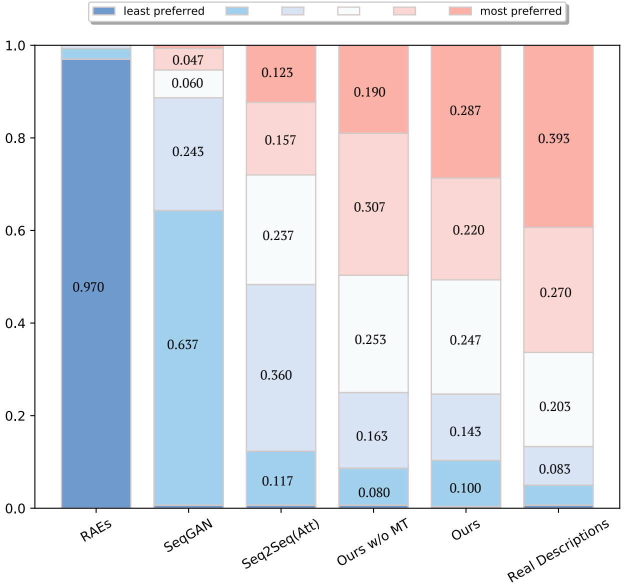
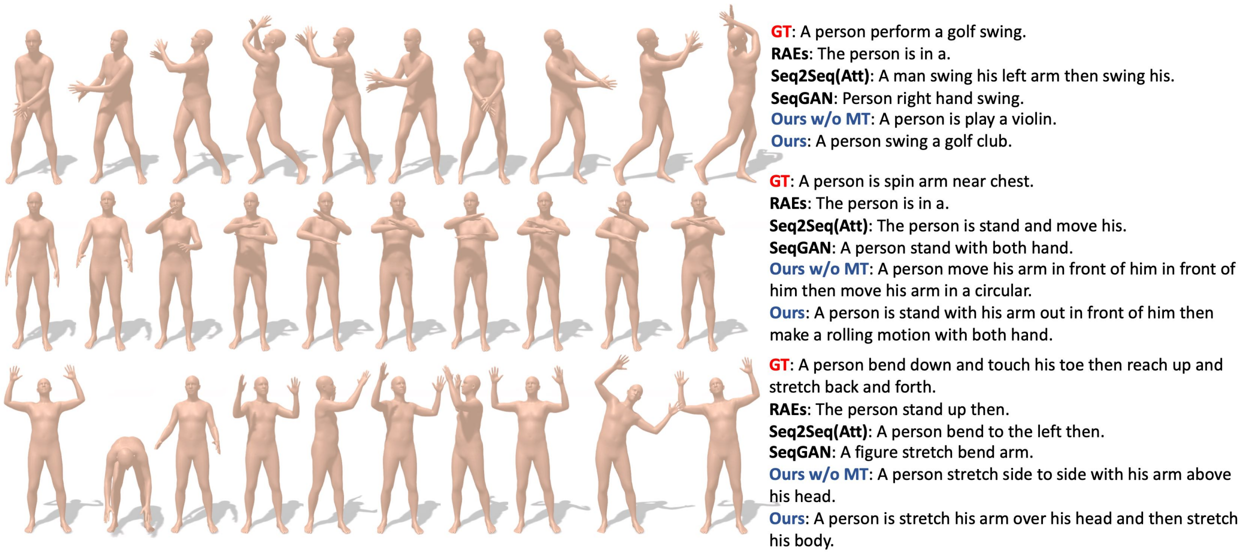


Figure 6.5: Statistics of human preference amongst the generated descriptions for given human motions. For each method, a color bar (from blue to red) indicated the the percentage of its preference level (from least to most preferred).

Qualitative Comparisons.

Fig. 6.6 qualitatively compares the generated descriptions from different methods grounded



GT: A person perform a golf swing.
RAEs: The person is in a.
Seq2Seq(Att): A man swing his left arm then swing his.
SeqGAN: Person right hand swing.
Ours w/o MT: A person is play a violin.
Ours: A person swing a golf club.

GT: A person is spin arm near chest.
RAEs: The person is in a.
Seq2Seq(Att): The person is stand and move his.
SeqGAN: A person stand with both hand.
Ours w/o MT: A person move his arm in front of him in front of him then move his arm in a circular.
Ours: A person is stand with his arm out in front of him then make a rolling motion with both hand.

GT: A person bend down and touch his toe then reach up and stretch back and forth.
RAEs: The person stand up then.
Seq2Seq(Att): A person bend to the left then.
SeqGAN: A figure stretch bend arm.
Ours w/o MT: A person stretch side to side with his arm above his head.
Ours: A person is stretch his arm over his head and then stretch his body.

Figure 6.6: Examples of motion-to-text translation results from different approaches. Grammatical tense and plural of words are not considered for simplifying learning process.

on the same 3D human motions. RAEs [39] consistently produces descriptions with simple patterns like ”is in a” , resulting in meaningless linguistic combinations; descriptions from Seq2Seq(Att) and SeqGAN are relatively more complex which however are usually incomplete and lack of details. Our approach without motion tokens starts to generate long and complex descriptions. Nonetheless, these descriptions sometimes fail to capture the characteristics of the input 3D motions (e.g, ”play a violin”). In contrast, our approach is able to provide fluent and descriptive sentences that accurately depict various aspects of 3D motions, such as body part (”both hand”), action category (”swing”, ”stretch”), spatial relations (”over head”).

User Study.

Beside the aforementioned objective evaluations, a crowd-sourced subjective assessment is also conducted on Amazon Mechanical Turk (AMT) involving hundreds of AMT users with *master* recognition. Particularly, descriptions are generated from 100 randomly selected 3D human motions using different methods. For each human motion, the corresponding generated and real descriptions are randomly reordered and shown to 3 AMT users, who are asked to rank their preference over these descriptions based on the accuracy and fluency.

Datasets	Methods	R Precision↑			FID↓	MM Dist↓	Diversity→	MModality↑
		Top 1	Top 2	Top 3				
Human ML3D	**Real motions**	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$0.002^{\pm.000}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	-
	Seq2Seq[37]	$0.180^{\pm.002}$	$0.300^{\pm.002}$	$0.396^{\pm.002}$	$11.75^{\pm.035}$	$5.529^{\pm.007}$	$6.223^{\pm.061}$	-
	Language2Pose[12]	$0.246^{\pm.002}$	$0.387^{\pm.002}$	$0.486^{\pm.002}$	$11.02^{\pm.046}$	$5.296^{\pm.008}$	$7.676^{\pm.058}$	-
	Text2Gesture[219]	$0.165^{\pm.001}$	$0.267^{\pm.002}$	$0.345^{\pm.002}$	$5.012^{\pm.030}$	$6.030^{\pm.008}$	$6.409^{\pm.071}$	-
	Hier[59]	$0.301^{\pm.002}$	$0.425^{\pm.002}$	$0.552^{\pm.004}$	$6.532^{\pm.024}$	$5.012^{\pm.018}$	$8.332^{\pm.042}$	-
	MoCoGAN[21]	$0.037^{\pm.000}$	$0.072^{\pm.001}$	$0.106^{\pm.001}$	$94.41^{\pm.021}$	$9.643^{\pm.006}$	$0.462^{\pm.008}$	$0.019^{\pm.000}$
	Dance2Music[53]	$0.033^{\pm.000}$	$0.065^{\pm.001}$	$0.097^{\pm.001}$	$66.98^{\pm.016}$	$8.116^{\pm.006}$	$0.725^{\pm.011}$	$0.043^{\pm.001}$
	Ours baseline(T)	$\underline{0.351}^{\pm.003}$	$0.521^{\pm.003}$	$0.627^{\pm.003}$	$\underline{1.669}^{\pm.025}$	$4.046^{\pm.018}$	$\mathbf{9.632}^{\pm.072}$	$\mathbf{4.352}^{\pm.149}$
	Ours baseline	$\underline{0.351}^{\pm.002}$	$\underline{0.526}^{\pm.002}$	$\underline{0.635}^{\pm.002}$	$1.739^{\pm.022}$	$\underline{3.965}^{\pm.010}$	$\underline{8.651}^{\pm.083}$	$\underline{3.139}^{\pm.083}$
	Ours	$\mathbf{0.424}^{\pm.003}$	$\mathbf{0.618}^{\pm.003}$	$\mathbf{0.729}^{\pm.002}$	$\mathbf{1.501}^{\pm.017}$	$\mathbf{3.467}^{\pm.011}$	$8.589^{\pm.076}$	$2.424^{\pm.093}$
KIT-ML	**Real motions**	$0.424^{\pm.005}$	$0.649^{\pm.006}$	$0.779^{\pm.006}$	$0.031^{\pm.004}$	$2.788^{\pm.012}$	$11.08^{\pm.097}$	-
	Seq2Seq[37]	$0.103^{\pm.003}$	$0.178^{\pm.005}$	$0.241^{\pm.006}$	$24.86^{\pm.348}$	$7.960^{\pm.031}$	$6.744^{\pm.106}$	-
	Language2Pose[12]	$0.221^{\pm.005}$	$0.373^{\pm.004}$	$0.483^{\pm.005}$	$6.545^{\pm.072}$	$5.147^{\pm.030}$	$9.073^{\pm.100}$	-
	Text2Gesture[219]	$0.156^{\pm.004}$	$0.255^{\pm.004}$	$0.338^{\pm.005}$	$12.12^{\pm.183}$	$6.964^{\pm.029}$	$9.334^{\pm.079}$	-
	Hier[59]	$0.255^{\pm.006}$	$\underline{0.432}^{\pm.007}$	$0.531^{\pm.007}$	$5.203^{\pm.107}$	$4.986^{\pm.027}$	$9.563^{\pm.072}$	-
	MoCoGAN[21]	$0.022^{\pm.002}$	$0.042^{\pm.003}$	$0.063^{\pm.003}$	$82.69^{\pm.242}$	$10.47^{\pm.012}$	$3.091^{\pm.043}$	$0.250^{\pm.009}$
	Dance2Music[53]	$0.031^{\pm.002}$	$0.058^{\pm.002}$	$0.086^{\pm.003}$	$115.4^{\pm.240}$	$10.40^{\pm.016}$	$0.241^{\pm.004}$	$0.062^{\pm.002}$
	Ours baseline(T)	$\underline{0.260}^{\pm.005}$	$0.426^{\pm.007}$	$\underline{0.538}^{\pm.008}$	$\underline{4.628}^{\pm.126}$	$4.835^{\pm.076}$	$\underline{12.16}^{\pm.120}$	$\underline{4.436}^{\pm.106}$
	Ours baseline	$0.251^{\pm.007}$	$0.418^{\pm.008}$	$0.535^{\pm.007}$	$4.814^{\pm.145}$	$\underline{4.682}^{\pm.048}$	$\mathbf{10.13}^{\pm.117}$	$\mathbf{4.486}^{\pm.117}$
	Ours	$\mathbf{0.280}^{\pm.005}$	$\mathbf{0.463}^{\pm.006}$	$\mathbf{0.587}^{\pm.005}$	$\mathbf{3.599}^{\pm.153}$	$\mathbf{4.591}^{\pm.026}$	$9.473^{\pm.117}$	$3.292^{\pm.081}$

Table 6.2: Quantitative evaluation results for text-to-motion mapping on HumanML3D and KIT-ML test sets. All baselines requires fixed motion lengths, and initial poses are further in demand for deterministic methods (first 4 baselines), which are all unnecessary in our approach. ± indicates 95% confidence interval, and → means the closer to the real motion the better. For each metric, the best score is highlighted in **bold**, while the second best is hightlighted using $\underline{\text{underscore}}$.

As shown in Figure.3, our method earns the most appreciation from users over all baselines. In detail, RAEs [38] is the least preferred method, from which 97% descriptions are ranked at the last place; Seq2Seq(Att) and SeqGAN [48] gain comparably more positive feedback from users; while our method without motion tokens comes to the second to the best. This objective study solidly substantiates the capability of our approach toward generating natural as well as motion-aligned language descriptions.

6.3.6 Evaluation of Text-to-motion Generation

Mapping language to 3D human motions in a non-deterministic fashion is relatively new. Here we compare our method to four state-of-the-art methods: Seq2Seq [37], Language2Pose [12], Text2Gesture [219] and Hier [59]. As with all existing methods, they are unfortunately deterministic methods. Therefore, two stochastic methods in other related fields are adopted here for more

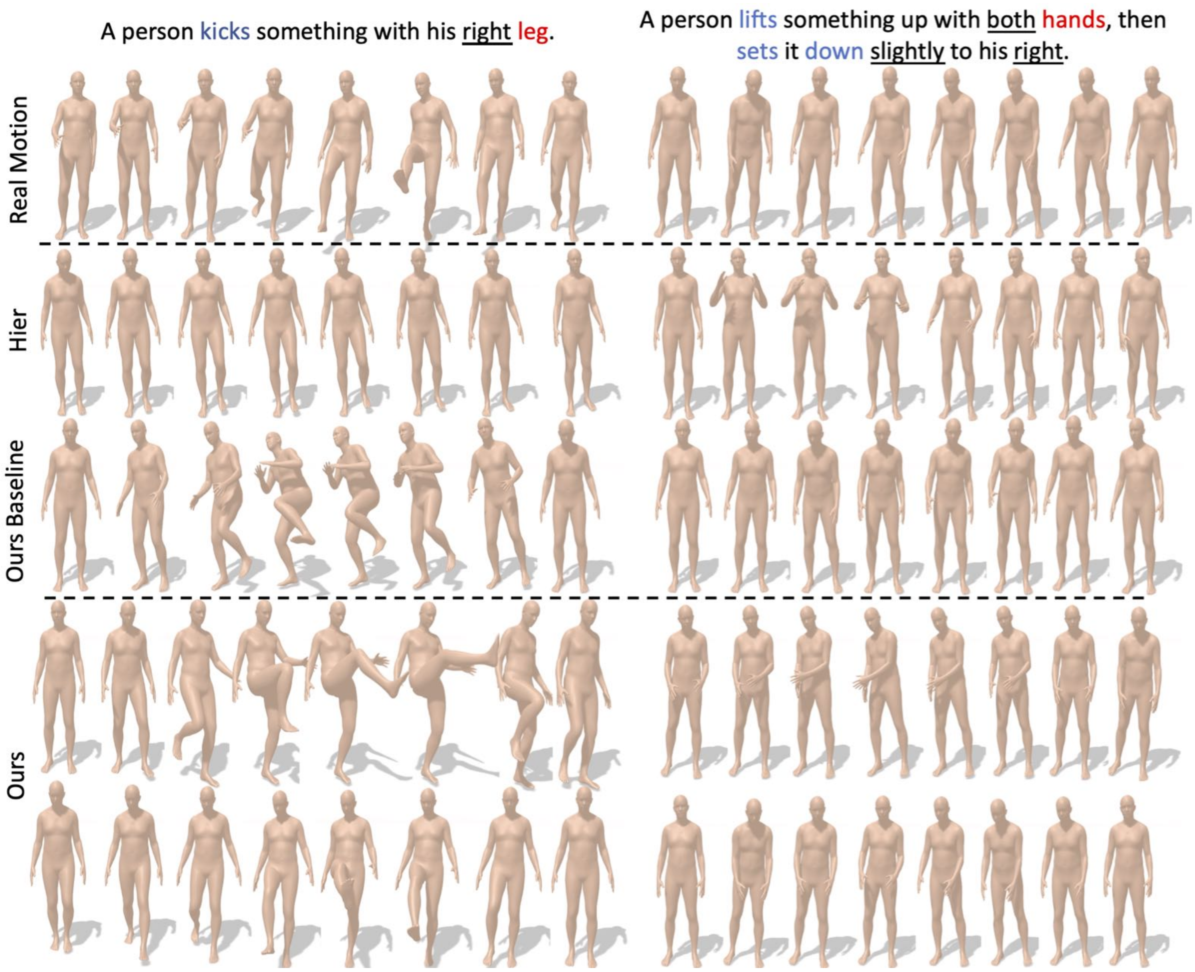

A person kicks something with his right leg.
A person lifts something up with both hands, then sets it down slightly to his right.
Real Motion
Hier
Ours Baseline
Ours

Figure 6.7: Visual comparisons of generated motions from the same language descriptions. For each description, we show its corresponding real motion, one motion from Hier [59] (since it's deterministic) and ours method without inverse alignment, as well as two motions from our method.

fair and in-depth evaluations: MoCoGAN [21] and Dance2Music [53]. MoCoGAN is widely used for conditioned video sequence synthesis, and Dance2Music learns to map sequential audio signals to 2D human dance motions. Proper changes are made to these methods for language-grounded 3D human motion generation. Ours baseline and ours baseline(T) ablates inverse alignment module during training Text2motion and map texts to motions using GRU and Transformer respectively. We repeat each experiment 20 times and report the mean value with 95% statistical confidence interval.

Quantitative Analysis.

Table 6.2 shows the quantitative evaluation results of language grounded 3D human motion generation. We can observe that the motions from non-determinstic baselines, MoCoGAN [21] and Dance2Music [53], suffers from severely low quality and diversity, as reflected by their low R precision and mutimodality score. Deterministic baselines such as Seq2Seq [37] and Text2Gesture [219] autoregressively regress human poses from textual input via vanilla sequence-to-sequence RNN and transformer respectively. However, such straightforward approaches find difficulty in maintaining textual semantics during generating human dynamics, which results in low motion-based text retrieval precision and high multimodal distance. Language2Pose [12] and Hier [59] propose to learn a co-embedding space between language and human motions, while Hier [59] go one step forward by incorporating the hierarchical topology of human skeleton. These have effectively boosted the performance on both datasets. Nevertheless, there still remain a significant gap between the synthetic results and real motions. Our framework of incorporating motion token and NMT model (*ours, ours baseline/baseline(T)*) in general achieve better performance, while the inverse alignment strategy greatly benefits this framework (*ours*) with the top-1 and top-3 precision increased by nearly 7% and 10% on HumanML3D.

Visual Comparisons.

In Fig. 6.7, we visually compares the generated motions from our method (ours), our method not using inverse alignment (ours baseline), and the best performing state of the art, Hier [59]. The corresponding real motions are also provided for reference. Hier [59] could somewhat capture partial concept (e.g., ”kick”) in descriptions, while the produced motions are unfaithfully in low-mobility. Our method without inverse alignment is capable of generating natural and plausible human motions. It sometimes however still fail to present fine details (e.g., ”right leg”) from texts. On the contrary, our approach consistently produce visually appealing motions which precisely convey the language concepts in descriptions.

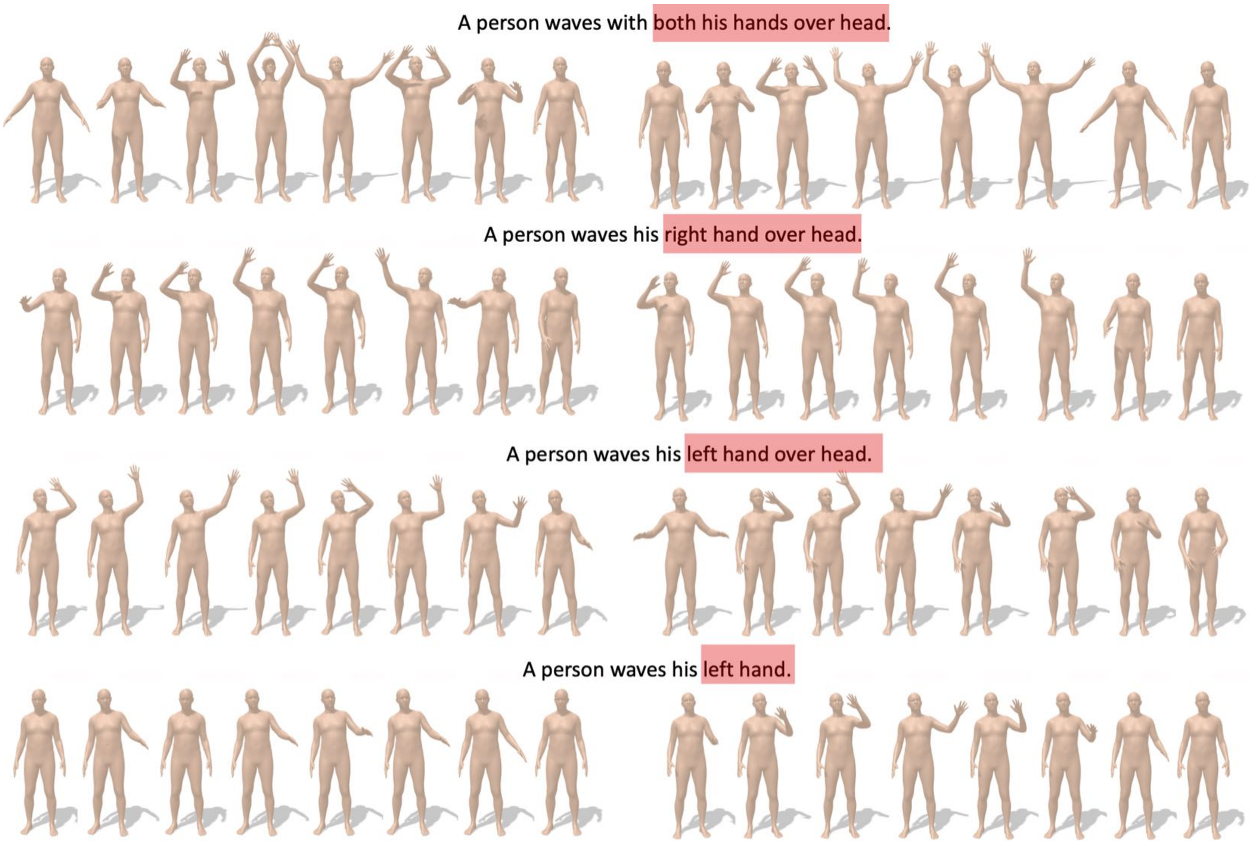

A person waves with both his hands over head.
A person waves his right hand over head.
A person waves his left hand over head.
A person waves his left hand.

Figure 6.8: Examples of text-to-motion mapping by modifying specific parts of text descriptions (highlighted in red box). For each description, we show two resultant motions.

Text Modifications

We also generate 3D motions from language by modifying fixed components of the input text descriptions (Fig. 6.8). Our text2motion is able to capture the subtle semantic differences (e.g., ”both/left/right hand”, ”over head”) in text descriptions.

Inference Time Analysis

Time consumption of generating 300 motions from different methods on one Nvidia2080Ti: Seq2Seq (14s), Language2Pose (10s), MoCoGAN (1s), Dance2Music (1s), Text2Gesture (250s), Hier(39s), Ours (9s). Benefiting from reduced time length, our approach is able to provide the same amount of motions with even less time cost than most baselines.

6.3.7 Limitations and Discussions

Although our proposed TM2T achieves superior performance on both tasks, some limitations and potential remedies can be taken into accounts in future studies. First, the approximation in motion quantization is unfortunately not lossless, which sometimes lead to blurriness and artifacts in local body (e.g., foot sliding). Second, dealing with long and complex descriptions for text2motion is somewhat beyond our capability. This could be possibly solved by using more advanced NMT models. Third, our motion2text model is trained independently with text2motion. Learning these two mapping functions jointly and reciprocally could be another interesting topic.

6.4 Conclusion

This chapter presents TM2T, a general framework that works on the bi-modal mutual mappings between 3D human motions and texts, where motion2text is further reciprocally integrated as a part of text2motion learning through inverse alignment. A new motion representation, motion token, is proposed that compress 3D motions into short sequence of discrete variables. With motion token, neural machine translation networks efficiently build mappings in-between two modalities, that is able to produces accurate descriptions as well as sharp and diverse 3D human motions. Our proposed framework is shown to produce state-of-the-art results on two motion-language dataset in both tasks.

Chapter 7

Conclusion and Discussion

Studying humans has profound impacts on both our daily lives and society, which is also a fundamental subject in computer vision. In this thesis, our research focuses on developing a series of solutions and resources to model and understand human actions intellectually using the recent expressive deep learning apparatus. This involves a set of topics including diverse and natural motion synthesis from action categories (action2motion) or textual descriptions (text2motion), image-based character animation (motion2video), generative motion stylization, and reciprocal generation of motions and texts (motion2text-2-motion). From a technical perspective, we investigate a range of deep generative models, including variational autoencoders and neural machine translation models. Although these models so far have achieved the-state-of-the-art performance, there are several limitations to our works and potential directions for future research.

Expressive Generative Models: We have explored the use of Variation Autoencoders, GANs, and GPT for human action generation. However, the field continues to advance with new models such as diffusion models [140] and masked generative Tranformers [232, 233]. These approaches may further improve the quality of generated motions and videos.

Dataset Scale, Diversity and Contexts: Currently, the motion datasets [43, 3, 1] are still in small scale compared to video and image datasets. Since motions are mostly captured indoor, these datasets often possess a limited number of objects, diversity, and content. Expanding these aspects of motion datasets, such as capturing outdoor motions, human-object interactions, and multi-person interactions is crucial for real-world applications. These are all important yet underexplored topics. Leveraging user-generated content on social media platforms like YouTube and TikTok can be a valuable resource for collecting such data.

Physics-Based Motion Generation: In the real world, human actions follow the physical law of the world, which has not been achieved yet in our works. For example, when jumping up and down, the speed should be constrained by gravity, or the foot would never slide or penetrate the ground floor. This is not easily integrated into a kinematic-based approach. One remedy is to operate the generation in a physics simulation engine, employing reinforcement learning to model the instant feedback from the simulated world.

Real-world Input Data: In Chapter 6, we make efforts to understand 3D human motions using natural language, where high-quality captured 3D motions are experimented. This however is different in the real world, where videos are more common and the estimated human poses are of much lower quality. To adapt to these wild applications, we plan to involve real-world 2D videos and motions on internet platforms for learning our models.

Real-Time Applications: Developing real-time capable models for applications like virtual reality, gaming, and live performance is an exciting direction. Efficient algorithms and hardware acceleration can be explored to meet the real-time processing requirements.

Human-AI Collaboration: Designing user-friendly interfaces and interactive systems that facilitate collaboration between humans and AI in motion synthesis and understanding tasks can make these technologies more accessible and useful.

Quality of Appearance: our pipeline of motion animation in Chap. 3, though being automatic, still involves complicated steps. Alternatively, recent advances [9, 234] directly reconstruct animatable 3D human avatars from single images, saving the intermediate steps of rigging. Moreover, the final appearance quality of our pipeline highly relies on the performance of human avatar reconstruction. If the camera views or human postures in the images are rare, the shape and texture reconstruction could easily fail.

References

[1] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020.

[2] Chuan Guo, Xinxin Zuo, Sen Wang, Xinshuang Liu, Shihao Zou, Minglun Gong, and Li Cheng. Action2video: Generating videos of human 3d actions. *International Journal of Computer Vision*, 130(2):285–315, 2022.

[3] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022.

[4] Chuan Guo, Yuxuan Mu, Xinxin Zuo, Peng Dai, Youliang Yan, Juwei Lu, and Li Cheng. Generative human motion stylization in latent space. In *The Twelfth International Conference on Learning Representations*, 2024.

[5] 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*, pages 580–597. Springer, 2022.

[6] Kfir Aberman, Yijia Weng, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Unpaired motion style transfer from video to animation. *ACM Transactions on Graphics (TOG)*, 39(4):64–1, 2020.

[7] Shihong Xia, Congyi Wang, Jinxiang Chai, and Jessica Hodgins. Realtime style transfer for unlabeled heterogeneous human motion. *ACM Transactions on Graphics (TOG)*, 34(4):1–10, 2015.

[8] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *IEEE International Conference on Computer Vision*, pages 5904–5913, 2019.

[9] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3093–3102, 2020.

[10] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo wake-up: 3d character animation from a single photo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5908–5917, 2019.

[11] Yunji Kim, Seonghyeon Nam, In Cho, and Seon Joo Kim. Unsupervised keypoint learning for guiding class-conditional video prediction. In *Advances in Neural Information Processing Systems*, pages 3809–3819, 2019.

[12] Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. In *International Conference on 3D Vision (3DV)*, pages 719–728. IEEE, 2019.

[13] Sebastian Starke, Yiwei Zhao, Fabio Zinno, and Taku Komura. Neural animation layering for synthesizing martial arts movements. *ACM Transactions on Graphics (TOG)*, 40(4):1–16, 2021.

[14] Russell H Taylor, Arianna Menciassi, Gabor Fichtinger, Paolo Fiorini, and Paolo Dario. Medical robotics and computer-integrated surgery. *Springer handbook of robotics*, pages 1657–1684, 2016.

[15] Mahdiar Nekoui and Li Cheng. Enhancing human motion assessment by self-supervised representation learning. BMVC, 2021.

[16] Haotian Zhang, Cristobal Sciutto, Maneesh Agrawala, and Kayvon Fatahalian. Vid2player: Controllable video sprites that behave and appear like professional tennis players. *ACM Transactions on Graphics (TOG)*, 40(3):1–16, 2021.

[17] Haotian Zhang, Ye Yuan, Viktor Makoviychuk, Yunrong Guo, Sanja Fidler, Xue Bin Peng, and Kayvon Fatahalian. Learning physically simulated tennis skills from broadcast videos. *ACM Transactions On Graphics (TOG)*, 42(4):1–14, 2023.

[18] Przemyslaw A Lasota and Julie A Shah. Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration. *Human factors*, 57(1):21–33, 2015.

[19] Kaifeng Zhao, Shaofei Wang, Yan Zhang, Thabo Beeler, and Siyu Tang. Compositional human-scene interaction synthesis with semantic control. In *European Conference on Computer Vision*, pages 311–327. Springer, 2022.

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[21] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[22] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.

[23] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.

[24] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[25] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[26] Vladimir Iashin and Esa Rahtu. Multi-modal dense video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 958–959, 2020.

[27] Yehao Li, Yingwei Pan, Ting Yao, and Tao Mei. Comprehending and ordering semantics for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17990–17999, 2022.

[28] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4570–4580, 2019.

[29] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2256–2265, 2021.

[30] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[32] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[33] Haoye Cai, Chunyan Bai, Yu-Wing Tai, and Chi-Keung Tang. Deep video generation, prediction and completion of human action sequences. In *Proceedings of the European conference on computer vision (ECCV)*, pages 366–382, 2018.

[34] Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–216, 2018.

[35] Xinchen Yan, Akash Rastogi, Ruben Villegas, Kalyan Sunkavalli, Eli Shechtman, Sunil Hadap, Ersin Yumer, and Honglak Lee. Mt-vae: Learning motion transformations to generate multimodal human dynamics. In *Proceedings of the European conference on computer vision (ECCV)*, pages 265–281, 2018.

[36] Ping Yu, Yang Zhao, Chunyuan Li, Junsong Yuan, and Changyou Chen. Structure-aware human-action generation. In *European Conference on Computer Vision*, pages 18–34. Springer, 2020.

[37] 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. *Learning*, 2018(1), 2018.

[38] Tatsuro Yamada, Hiroyuki Matsunaga, and Tetsuya Ogata. Paired recurrent autoencoders for bidirectional translation between robot actions and linguistic descriptions. *IEEE Robotics and Automation Letters*, 3(4):3441–3448, 2018.

[39] Matthias Plappert, Christian Mandery, and Tamim Asfour. Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks. *Robotics and Autonomous Systems*, 109:13–26, 2018.

[40] Lu Xia, Chia-Chih Chen, and Jake K Aggarwal. View invariant human action recognition using histograms of 3d joints. In *CVPR Workshops*, pages 20–27, 2012.

[41] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *CVPR Workshop on Human Communicative Behavior Analysis*, pages 9–14, 2010.

[42] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[43] Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big data*, 4(4):236–252, 2016.

[44] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *IEEE/CVF International Conference on Computer Vision*, pages 5933–5942, 2019.

[45] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. In *Advances in Neural Information Processing Systems*, 2019.

[46] César Roberto de Souza, Adrien Gaidon, Yohann Cabon, Naila Murray, and Antonio Manuel López. Generating human action videos by coupling 3d game engines and probabilistic graphical models. *International Journal of Computer Vision*, 128(5):1505–1536, 2020.

[47] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[48] Yusuke Goutsu and Tetsunari Inamura. Linguistic descriptions of human motion with generative adversarial seq2seq learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4281–4287. IEEE, 2021.

[49] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[50] Kenta Takeuchi, Dai Hasegawa, Shinichi Shirakawa, Naoshi Kaneko, Hiroshi Sakuta, and Kazuhiko Sumi. Speech-to-gesture generation: A challenge in deep learning approach with bi-directional lstm. In *Proceedings of the 5th International Conference on Human Agent Interaction*, pages 365–369, 2017.

[51] Eli Shlizerman, Lucio Dery, Hayden Schoen, and Ira Kemelmacher-Shlizerman. Audio to body dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7574–7583, 2018.

[52] Taoran Tang, Jia Jia, and Hanyang Mao. Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis. In *Proceedings of the 26th ACM international conference on Multimedia (ACM MM)*, pages 1598–1606, 2018.

[53] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In *Advances in Neural Information Processing Systems*, pages 3581–3591, 2019.

[54] Ruozi Huang, Huang Hu, Wei Wu, Kei Sawada, and Mi Zhang. Dance revolution: Long-term dance generation with music via curriculum learning. In *International Conference on Learning Representations*, 2021.

[55] Rui Zhao, Hui Su, and Qiang Ji. Bayesian adversarial human motion synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6225–6234, 2020.

[56] Rui Zhao and Qiang Ji. An adversarial hierarchical hidden markov model for human pose modeling and generation. In *AAAI Conference on Artificial Intelligence*, pages 2636–2643, 2018.

[57] Sijie Yan, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin. Convolutional sequence generation for skeleton-based action synthesis. In *IEEE/CVF International Conference on Computer Vision*, pages 4394–4402, 2019.

[58] Jingwei Xu, Huazhe Xu, Bingbing Ni, Xiaokang Yang, Xiaolong Wang, and Trevor Darrell. Hierarchical style-based networks for motion synthesis. In *European Conference on Computer Vision*, pages 178–194, 2020.

[59] Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. Synthesis of compositional animations from textual descriptions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1396–1406, 2021.

[60] Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 480–497. Springer, 2022.

[61] Nikos Athanasiou, Mathis Petrovich, Michael J Black, and Gül Varol. Teach: Temporal action composition for 3d humans. In *2022 International Conference on 3D Vision (3DV)*, pages 414–423. IEEE, 2022.

[62] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 358–374. Springer, 2022.

[63] Junfan Lin, Jianlong Chang, Lingbo Liu, Guanbin Li, Liang Lin, Qi Tian, and Changwen Chen. Being comes from not-being: Open-vocabulary text-to-motion generation with wordless training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23222–23231, 2023.

[64] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable

visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[65] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[66] Thomas Lucas, Fabien Baradel, Philippe Weinzaepfel, and Grégory Rogez. Posegpt: Quantization-based 3d human motion generation and forecasting. In *European Conference on Computer Vision*, pages 417–435. Springer, 2022.

[67] 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. *arXiv preprint arXiv:2301.06052*, 2023.

[68] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *arXiv preprint arXiv:2306.14795*, 2023.

[69] Yaqi Zhang, Di Huang, Bin Liu, Shixiang Tang, Yan Lu, Lu Chen, Lei Bai, Qi Chu, Nenghai Yu, and Wanli Ouyang. Motiongpt: Finetuned llms are general-purpose motion generators. *arXiv preprint arXiv:2306.10900*, 2023.

[70] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11050–11059, 2022.

[71] Kehong Gong, Dongze Lian, Heng Chang, Chuan Guo, Xinxin Zuo, Zihang Jiang, and Xinchao Wang. Tm2d: Bimodality driven 3d dance generation via music-text integration. *arXiv preprint arXiv:2304.02419*, 2023.

[72] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

[73] Jihoon Kim, Jiseob Kim, and Sungjoon Choi. Flame: Free-form language-based motion synthesis & editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8255–8263, 2023.

[74] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022.

[75] Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 448–458, 2023.

[76] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18000–18010, 2023.

[77] Hanyang Kong, Kehong Gong, Dongze Lian, Michael Bi Mi, and Xinchao Wang. Priority-centric human motion generation in discrete latent space. *arXiv preprint arXiv:2308.14480*, 2023.

[78] Yunhong Lou, Linchao Zhu, Yaxiong Wang, Xiaohan Wang, and Yi Yang. Diversemotion: Towards diverse human motion generation via discrete diffusion. *arXiv preprint arXiv:2309.01372*, 2023.

[79] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[80] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[81] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

[82] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Advances in Neural Information Processing Systems*, pages 1144–1156, 2018.

[83] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2377–2386, 2019.

[84] Jessica Lee, Deva Ramanan, and Rohit Girdhar. MetaPix: Few-Shot Video Retargeting. In *International Conference on Learning Representations*, 2020.

[85] Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. Neural kinematic networks for unsupervised motion retargetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8639–8648, 2018.

[86] Kfir Aberman, Peizh Uo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG)*, 39(4):62–1, 2020.

[87] Zhenglong Zhou, Bo Shu, Shaojie Zhuo, Xiaoming Deng, Ping Tan, and Stephen Lin. Image-based clothes animation for virtual fitting. In *SIGGRAPH Asia*, pages 1–4. 2012.

[88] Alexander Hornung, Ellen Dekkers, and Leif Kobbelt. Character animation from 2d pictures and 3d motion data. *ACM Transactions on Graphics*, 26(1):1–es, 2007.

[89] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[90] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016.

[91] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *arXiv preprint arXiv:1603.03417*, 2016.

[92] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[93] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1897–1906, 2017.

[94] Hang Zhang and Kristin Dana. Multi-style generative network for real-time transfer. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[95] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

[96] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[97] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[98] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[99] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems*, 33:7198–7211, 2020.

[100] Kenji Amaya, Armin Bruderlin, and Tom Calvert. Emotion from motion. In *Graphics interface*, volume 96, pages 222–229. Toronto, Canada, 1996.

[101] Munetoshi Unuma, Ken Anjyo, and Ryozo Takeuchi. Fourier principles for emotion-based human figure animation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 91–96, 1995.

[102] Eugene Hsu, Kari Pulli, and Jovan Popović. Style translation for human motion. In *ACM SIGGRAPH 2005 Papers*, pages 1082–1089. Association for Computing Machinery, 2005.

[103] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016.

[104] Han Du, Erik Herrmann, Janis Sprenger, Klaus Fischer, and Philipp Slusallek. Stylistic locomotion modeling and synthesis using variational generative models. In *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games*, pages 1–10, 2019.

[105] M. Ersin Yumer and Niloy J. Mitra. Spectral style transfer for human motion between independent actions. *ACM Transactions on Graphics*, 35(4):1–8, July 2016.

[106] Daniel Holden, Ikhsanul Habibie, Ikuo Kusajima, and Taku Komura. Fast neural style transfer for motion data. *IEEE computer graphics and applications*, 37(4):42–49, 2017.

[107] Harrison Jesse Smith, Chen Cao, Michael Neff, and Yingying Wang. Efficient Neural Networks for Real-time Motion Style Transfer. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 2(2):1–17, July 2019.

[108] Soomin Park, Deok-Kyeong Jang, and Sung-Hee Lee. Diverse motion stylization for multiple style domains via spatial-temporal graph-based generative model. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 4(3):1–17, 2021.

[109] Deok-Kyeong Jang, Soomin Park, and Sung-Hee Lee. Motion puzzle: Arbitrary motion style transfer by body part. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022.

[110] Tianxin Tao, Xiaohang Zhan, Zhongquan Chen, and Michiel van de Panne. Style-ERD: Responsive and Coherent Online Motion Style Transfer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6583–6593, New Orleans, LA, USA, June 2022. IEEE.

[111] Yu-Hui Wen, Zhipeng Yang, Hongbo Fu, Lin Gao, Yanan Sun, and Yong-Jin Liu. Autoregressive stylized motion synthesis with generative flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13612–13621, 2021.

[112] Sigal Raab, Inbal Leibovitch, Guy Tevet, Moab Arar, Amit H Bermano, and Daniel Cohen-Or. Single motion diffusion. *arXiv preprint arXiv:2302.05905*, 2023.

[113] Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *International Journal of Computer Vision*, 50(2):171–184, 2002.

[114] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2891–2903, 2013.

[115] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

[116] Yongri Piao, Wei Ji, Jingjing Li, Miao Zhang, and Huchuan Lu. Depth-induced multi-scale recurrent attention network for saliency detection. In *ICCV*, pages 7254–7263, 2019.

[117] Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara L Berg, and Mohit Bansal. Mart: Memory-augmented recurrent transformer for coherent video paragraph captioning. *arXiv preprint arXiv:2005.05402*, 2020.

[118] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[119] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8347–8356, 2019.

[120] Miao Zhang, Jingjing Li, Wei Ji, Yongri Piao, and Huchuan Lu. Memory-oriented decoder for light field salient object detection. In *NeurIPS*, pages 896–906, 2019.

[121] Jae Sung Park, Marcus Rohrbach, Trevor Darrell, and Anna Rohrbach. Adversarial inference for multi-sentence video description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6598–6608, 2019.

[122] Ramakanth Pasunuru and Mohit Bansal. Reinforced video captioning with entailment rewards. *arXiv preprint arXiv:1708.02300*, 2017.

[123] Lijun Li and Boqing Gong. End-to-end video captioning with multitask reinforcement learning. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 339–348. IEEE, 2019.

[124] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1029–1038, 2016.

[125] Bairui Wang, Lin Ma, Wei Zhang, Wenhao Jiang, Jingwen Wang, and Wei Liu. Controllable video captioning with pos sequence guidance based on gated fusion network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2641–2650, 2019.

[126] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[127] Wataru Takano and Yoshihiko Nakamura. Statistical mutual conversion between whole body motion primitives and linguistic sentences for human motions. *The International Journal of Robotics Research*, 34(10):1314–1328, 2015.

[128] Fei Han, Brian Reily, William Hoff, and Hao Zhang. Space-time representation of people based on 3d skeletal data: A review. *Computer Vision and Image Understanding*, 158:85–105, 2017.

[129] Mohamed E Hussein, Marwan Torki, Mohammad A Gowayyed, and Motaz El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.

[130] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1297. IEEE, 2012.

[131] Alexandros Andre Chaaraoui, José Ramón Padilla-López, Pau Climent-Pérez, and Francisco Flórez-Revuelta. Evolutionary joint selection to improve human action recognition with rgb-d devices. *Expert systems with applications*, 41(3):786–794, 2014.

[132] Yaser Yacoob and Michael J Black. Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding*, 73(2):232–247, 1999.

[133] Meinard Müller. *Information retrieval for music and motion*, volume 2. Springer, 2007.

[134] Dariu M Gavrila, Larry S Davis, et al. Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *International workshop on automatic face-and gesture-recognition*, pages 272–277. Citeseer, 1995.

[135] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 588–595, 2014.

[136] Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 6099–6108, 2017.

[137] Chi Xu, Lakshmi Narasimhan Govindarajan, Yu Zhang, and Li Cheng. Lie-x: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups. *International Journal of Computer Vision*, 123(3):454–478, 2017.

[138] Zhenguang Liu, Shuang Wu, Shuyuan Jin, Qi Liu, Shijian Lu, Roger Zimmermann, and Li Cheng. Towards natural and accurate future motion prediction of humans and animals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10004–10012, 2019.

[139] Dario Pavllo, Christoph Feichtenhofer, Michael Auli, and David Grangier. Modeling human motion with quaternion-based neural networks. *International Journal of Computer Vision*, 128(4):855–872, 2020.

[140] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[141] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35:31841–31854, 2022.

[142] Kundan Kumar, Rithesh Kumar, Thibault De Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre De Brebisson, Yoshua Bengio, and Aaron C Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. *Advances in neural information processing systems*, 32, 2019.

[143] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.

[144] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE, 2020.

[145] Yitong Li, Martin Min, Dinghan Shen, David Carlson, and Lawrence Carin. Video generation from text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[146] Andres Munoz, Mohammadreza Zolfaghari, Max Argus, and Thomas Brox. Temporal shift gan for large scale video generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3179–3188, 2021.

[147] Zhenyi Wang, Ping Yu, Yang Zhao, Ruiyi Zhang, Yufan Zhou, Junsong Yuan, and Changyou Chen. Learning diverse stochastic human-action generators by learning smooth latent transitions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12281–12288, 2020.

[148] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[149] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[150] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015.

[151] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[152] Narayanaswamy Siddharth, Brooks Paige, Jan-Willem Van de Meent, Alban Desmaison, Noah Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. Learning disentangled rep-

resentations with semi-supervised deep generative models. In *Advances in Neural Information Processing Systems*, pages 5925–5935, 2017.

[153] Yen-Chi Cheng, Hsin-Ying Lee, Min Sun, and Ming-Hsuan Yang. Controllable image synthesis via segvae. In *European Conference on Computer Vision*, pages 159–174, 2020.

[154] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8247–8255, 2019.

[155] Zheng Ding, Yifan Xu, Weijian Xu, Gaurav Parmar, Yang Yang, Max Welling, and Zhuowen Tu. Guided variational autoencoder for disentanglement learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7920–7929, 2020.

[156] Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf. S3vae: Self-supervised sequential vae for representation disentanglement and data generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6538–6547, 2020.

[157] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016.

[158] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning*, pages 1558–1566, 2016.

[159] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, pages 2980–2988, 2015.

[160] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Conference on Computational Natural Language Learning*, 2016.

[161] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In *International Conference on Machine Learning*, pages 3881–3890, 2017.

[162] Tanya Marwah, Gaurav Mittal, and Vineeth N Balasubramanian. Attentive semantic video generation using captions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1426–1434, 2017.

[163] Jiawei He, Andreas Lehrmann, Joseph Marino, Greg Mori, and Leonid Sigal. Probabilistic video generation using holistic attribute control. In *European Conference on Computer Vision*, pages 452–467, 2018.

[164] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International Conference on Machine Learning (ICML)*, pages 1174–1183, 2018.

[165] Tsun-Hsuan Wang, Yen-Chi Cheng, Chieh Hubert Lin, Hwann-Tzong Chen, and Min Sun. Point-to-point video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10491–10500, 2019.

[166] Ikhsanul Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and Taku Komura. A recurrent variational autoencoder for human motion synthesis. In *British Machine Vision Conference*, 2017.

[167] Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Lars Petersson, and Stephen Gould. A stochastic conditioning scheme for diverse human motion prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5223–5232, 2020.

[168] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[169] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[170] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[171] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[172] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[173] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[174] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.

[175] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[176] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481*, 2022.

[177] Eyal Molad, Eliahu Horwitz, Dani Valevski, Alex Rav Acha, Yossi Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. Dreamix: Video diffusion models are general video editors. *arXiv preprint arXiv:2302.01329*, 2023.

[178] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7346–7356, 2023.

[179] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

[180] Rongjie Huang, Max WY Lam, Jun Wang, Dan Su, Dong Yu, Yi Ren, and Zhou Zhao. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. *arXiv preprint arXiv:2204.09934*, 2022.

[181] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[182] Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. Mofusion: A framework for denoising-diffusion-based motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9760–9770, 2023.

[183] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, Jingyi Yu, and Gang Yu. Executing your commands via motion diffusion in latent space. *arXiv preprint arXiv:2212.04048*, 2022.

[184] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[185] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[186] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[187] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021.

[188] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

[189] CMU. Cmu graphics lab motion capture database. 2003.

[190] Jun Liu, Amir Shahroudy, Mauricio Lisboa Perez, Gang Wang, Ling-Yu Duan, and Alex Kot Chichung. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[191] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

[192] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F Troje. Movi: A large multipurpose motion and video dataset. *arXiv preprint arXiv:2003.01888*, 2020.

[193] Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. Babel: Bodies, action and behavior with english labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 722–731, 2021.

[194] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5442–5451, 2019.

[195] Yue Wu, Rongrong Gao, Jaesik Park, and Qifeng Chen. Future video synthesis with object motion prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5539–5548, 2020.

[196] Hang Gao, Huazhe Xu, Qi-Zhi Cai, Ruth Wang, Fisher Yu, and Trevor Darrell. Disentangling propagation and generation for video prediction. In *IEEE/CVF International Conference on Computer Vision*, pages 9006–9015, 2019.

[197] Carl Vondrick and Antonio Torralba. Generating the future with adversarial transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1020–1028, 2017.

[198] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[199] Verica Lazova, Eldar Insafutdinov, and Gerard Pons-Moll. 360-degree textures of people in clothing from a single image. In *International Conference on 3D Vision*, pages 643–653, 2019.

[200] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *IEEE/CVF International Conference on Computer Vision*, pages 2304–2314, 2019.

[201] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 84–93, 2020.

[202] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[203] Vida Adeli, Ehsan Adeli, Ian Reid, Juan Carlos Niebles, and S Hamid Rezatofighi. Socially and contextually aware human motion and pose forecasting. *IEEE Robotics and Automation Letters*, 2020.

[204] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. Structured prediction helps 3d human motion modelling. In *IEEE/CVF International Conference on Computer Vision*, pages 7144–7153, 2019.

[205] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015.

[206] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International conference on machine learning*, pages 41–48, 2009.

[207] Z Cao, T Simon, SE Wei, YA Sheikh, et al. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186, 2021.

[208] S Geman and D McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, pages 5–21, 1987.

[209] 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*, pages 561–578, 2016.

[210] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018.

[211] Xinxin Zuo, Sen Wang, Jiangbin Zheng, Weiwei Yu, Minglun Gong, Ruigang Yang, and Li Cheng. Sparsefusion: Dynamic human avatar modeling from sparse rgbd images. *IEEE Transactions on Multimedia*, 23:1617–1629, 2020.

[212] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007.

[213] Chao Zhang, Sergi Pujades, Michael J. Black, and Gerard Pons-Moll. Detailed, accurate, human shape estimation from clothed 3d scan sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5484–5493, 2017.

[214] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8387–8397, 2018.

[215] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2020.

[216] Hyemin Ahn, Timothy Ha, Yunho Choi, Hwiyeon Yoo, and Songhwai Oh. Text2action: Generative adversarial synthesis from language to action. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 5915–5920. IEEE, 2018.

[217] Stephanie Stoll, Necati Cihan Camgoz, Simon Hadfield, and Richard Bowden. Text2sign: Towards sign language production using neural machine translation and generative adversarial networks. *International Journal of Computer Vision*, 128:891–908, 2020.

[218] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[219] Uttaran Bhattacharya, Nicholas Rewkowski, Abhishek Banerjee, Pooja Guhan, Aniket Bera, and Dinesh Manocha. Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents. In *IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 1–10. IEEE, 2021.

[220] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756. PMLR, 2016.

[221] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.

[222] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[223] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[224] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[225] CMU. Carnegie-mellon mocap database. Retrieved from http://mocap.cs.cmu.edu.

[226] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[227] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[228] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[229] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[230] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[231] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[232] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

[233] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-

to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.

[234] Tong He, Yuanlu Xu, Shunsuke Saito, Stefano Soatto, and Tony Tung. Arch++: Animation-ready clothed human reconstruction revisited. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11046–11056, 2021.