TEXT-DRIVEN MOTION SYNTHESIS AND INTERACTION GENERATION USING MASKED DECONSTRUCTED DIFFUSION AND MULTI-TASK SCENE-AWARE MODELS

TEXT-DRIVEN MOTION SYNTHESIS AND INTERACTION GENERATION USING MASKED DECONSTRUCTED DIFFUSION AND MULTI-TASK SCENE-AWARE MODELS

By JIA CHEN, B.Eng.

A Thesis Submitted to the School of Graduate Studies in Partial Fulfillment of the Requirements for the Degree Master of Science

McMaster University
MASTER OF SCIENCE (2025)
Hamilton, Ontario, Canada (Computing and Software)

TITLE:	Text-driven Motion Synthesis and Interaction Generation using Masked Deconstructed Diffusion and Multi-task Scene-aware Models
AUTHOR:	Jia Chen B.Eng. (Computer Science and Technology), Hangzhou Dianzi University, Hangzhou, China
SUPERVISOR:	Dr. Yingying Wang
NUMBER OF PAGES:	xvii, 95

Lay Abstract

This thesis introduces a new generative AI approach that addresses three long-standing hurdles in human motion generation: accuracy, speed, and reliable alignment with user-written text. From a simple sentence, the system quickly produces natural, high-quality 3D movements that can be retargeted to digital characters for animation, virtual reality (VR), and games. The experiment demonstrates its practical value in VR, where the generated motions enhance immersion and responsiveness. Building on this, the thesis explores a second, scene-aware model that works with large language models to understand both the instruction and the surrounding scene. It can break down long requests into smaller steps and generate motions that interact with objects, for example walking to a chair and then sitting down. Together, these contributions point to more intuitive, text-driven tools for creating lifelike character animation.

Abstract

This thesis introduces Masked Deconstructed Diffusion (MDD) for text-to-motion generation and a complementary multi-task scene-aware pipeline (MTSA-T2M). MDD couples a multi-stage Kinematic Chain Quantization (KCQ) module with a Masked Deconstructed Diffusion Transformer (MDDT): KCQ compresses human motion into a compact, expressive codebook by jointly capturing local joint dynamics and global kinematic structure, while MDDT performs parallel masked index refinement conditioned on text, enabling faithful many-to-many text–motion mapping with fast inference. Empirically, the approach improves semantic alignment and motion quality against contemporary baselines while reducing runtime, and the discrete interface simplifies user control and editing. Building on this core, MTSA-T2M targets long, composite instructions in 3D scenes: it decomposes a prompt, plans sub-goals against a scene map, and invokes aligned diffusion modules to synthesize coherent motion segments that respect navigation and interaction cues. The resulting motions transfer effectively to VR avatars and scenes through rapid retargeting, supporting interactive applications and VR prototyping. Together, MDD and MTSA-T2M advance text-driven human motion synthesis by jointly addressing accuracy, diversity, efficiency, and scene awareness.

Acknowledgements

I would like to extend my heartfelt appreciation to my supervisor, Dr. Yingying Wang, whose guidance, encouragement, and thoughtful advice have been invaluable throughout my Master's studies.

My thanks also go to my colleagues and friends for their generous support, stimulating conversations, and motivation that helped me grow during this work.

I am grateful as well for the financial assistance from McMaster University, which provided the foundation for carrying out this research.

Above all, I owe the deepest gratitude to my family. Their unwavering love, patience, and belief in me have been a steady source of strength whenever I encountered obstacles.

Lastly, I would like to express my appreciation for inspiring 3D animation works such as *Girls Band Cry, BanG Dream*, which have continually fueled my passion for research on human motion generation.

Table of Contents

List of Figures

List of Tables

Notation, Definitions, and Abbreviations

Notation

$\mathbf{c}$	Text condition.
$\mathbf{c}_i$	Sub-prompt obtained from decomposition.
$\mathbf{m} \in \mathbb{R}^{N \times D}$	Human motion with N frames and D-dimensional pose features per frame.
$\mathbf{m}_i$	Scene-aware motion segment for sub-task i.
N, D	Number of frames; feature dimensionality per frame.
T, J	Number of frames; number of body joints in the skeletal format.
$\mathbf{r}_h$	Root height.
$\mathbf{r}_{rv}$	Root angular velocity.
$\mathbf{r}_{lv}$	Root linear velocity.

$\mathbf{j}_r$	Joint rotations.
$\mathbf{j}_{lp}$	Joint local positions.
$\mathbf{j}_v$	Joint velocities.
$\mathbf{f}$	Foot-contact indicators.
$\mathcal{E}, \mathcal{D}$	Encoder and decoder networks for motion reconstruction.
$\mathbf{z} \in \mathbb{R}^{n \times d}$	Latent sequence (compressed time n and latent dimension d).
$\widehat{\mathbf{z}}$	Quantized latent sequence (after vector quantization).
$\mathcal{Q}(\cdot)$	Quantizer mapping $\mathbf{z}$ to nearest code entries.
$\mathcal{C} \in \mathbb{R}^{K \times d}$	Codebook with K discrete code vectors (entries).
$\mathbf{b} \in \mathbb{R}^{n}$	Index sequence referencing entries in $\mathcal{C}$.
$\overline{\mathbf{b}}$	Masked version of $\mathbf{b}$ used as input to MDDT.
$\widehat{\mathbf{b}}$	Predicted index sequence output by MDDT.
$\oplus$	Element-wise summation used for hierarchical feature fusion.
t	Diffusion step (training or inference timestep).
L	Number of inference refinement steps.
$r(\beta)$	Mask ratio schedule with $\beta \sim \mathcal{U}(0, 1)$.
γ_t, σ_t	Diffusion schedule scalars with $\gamma_t^2 + \sigma_t^2 = 1$.
η_t	Noise-increase schedule parameter in conventional DDPM settings.

h	Text/step embedding dimensionality for the transformer input.
$\mathbb{E}[\cdot]$	Expectation over data and noise draws.
$\mathcal{L}_{\text{KCQ}}$	KCQ training loss (reconstruction plus VQ commitment/precision).
$\mathcal{L}^{t}_{\text{MDDT}}$	MDDT training loss at step t (weighted cross-entropy over indices).
$\mathbf{R}$	Scene road map in $\{0,1,2\}^{H\times W}$ (0 free, 1 obstacle, 2 target).
$\mathbf{H}^{o}$	Object o's top-surface height map in $\mathbb{R}^{h_o\times w_o}$.
$\mathbf{g}_i$	Planner guidance for sub-task i: trajectory sketch, keyframes, contact cues.
Ω_i	Index set of constrained frames and joints for alignment.
$\boldsymbol{\xi}$	Stochastic noise/seed for sampling.

Definitions

Motion A sequence of 3D human poses $\mathbf{m} \in \mathbb{R}^{N\times D}$ over N frames, where each frame stores D-dimensional pose attributes (root signals, joint rotations and positions, velocities, and foot-contact indicators).

Scene A 3D environment with layout and objects, represented by a road map $\mathbf{R}$ for traversability and per-object height maps $\mathbf{H}^{o}$ for contact reasoning.

Abbreviations

MDD	Masked Deconstructed Diffusion (overall text-to-motion framework).
KCQ	Kinematic Chain Quantization (representation and reconstruction module).
MDDT	Masked Deconstructed Diffusion Transformer (index-prediction module).
MTSA-T2M	Multi-task Scene-aware Text-to-Motion (scene-aware pipeline built on MDM).
VQ-VAE	Vector Quantized Variational Autoencoder.
DDPM	Denoising Diffusion Probabilistic Model.
MDM	Human Motion Diffusion Model.
RVQ	Residual Vector Quantization.
CLIP	Contrastive Language–Image Pretraining.
FID	Fréchet Inception Distance.
R-Precision	Retrieval Precision.
MM-Dist	Multimodal Distance.
MModality	Multimodality.
AIT	Average Inference Time.

VR	Virtual Reality
AR	Augmented Reality
LLM	Large Language Model

Declaration of Academic Achievement

This thesis contains my research outcome from 2024 to 2025.

Chapter 1

Introduction

The generation of 3D human motions that align with user intentions has emerged as a vibrant area of research, driven by its broad utility in domains such as virtual reality (VR), augmented reality (AR), video games, and various forms of digital media. Among the different input modalities, natural language stands out as one of the most intuitive and accessible interfaces for guiding motion synthesis. As a result, text-to-motion approaches have received growing attention, offering users the ability to create complex motion sequences with simple verbal descriptions. Beyond producing motions in isolation, a further challenge arises when these motions are deployed within concrete scenes. In such contexts, users often expect characters to perform actions that meaningfully interact with their surroundings, for instance, "sitting on a chair" or "walking towards the door."

1.1 Challenges

Generating realistic and expressive 3D human motions from textual descriptions and scenes presents several critical challenges. 1) Text-based conditions are often abstract and inherently ambiguous, leaving space for multiple plausible motion outcomes. The many-to-many relationship between language and motion makes it difficult to establish a deterministic one-to-one mapping from sentences to motion sequences. 2) Raw motion data exist in high-dimensional spaces and are both redundant and noisy, making it essential to derive compact latent representations. Yet, employing a single autoencoder or variational autoencoder is limiting, as they rely on either deterministic mappings or a single Gaussian distribution, often resulting in poor precision and a lack of diversity. 3) Synthesizing long motion sequences is computationally demanding. Frame-by-frame autoregressive generation is prohibitively slow, and extended sequences frequently suffer from temporal drift, which can lead to unnatural or degenerate neutral poses. 4) Interactions within scenes are often composed of multiple stages, requiring the model to understand contextual relationships among spatial and temporal scene elements, including objects. Existing methods are generally confined to single-step motions and rely on overly complex multimodal architectures to unify different input modalities.

1.2 Methods

In this paper, I propose a Masked Deconstructed Diffusion (MDD) framework for generating plausible and diverse 3D human motions conditioned on text input, with an efficient runtime suitable for VR applications. Considering the hierarchical kinematic

structure of human models, MDD employs multi-stage Kinematic Chain Quantization (KCQ) encoders to learn a compact and expressive codebook from human motions. Each encoder focuses on a specific body part, such as the left arm or right leg. Local kinematic features are then hierarchically fused to form the latent vectors of the full-body motion. These continuous latent vectors are further discretized through quantization and organized into a codebook. Codebook entries, referenced by their indices, can be inverse-projected back to the human motion space through decoders trained by minimizing motion reconstruction loss. Given a text input, the MDD framework generates the corresponding motions using a Masked Deconstructed Diffusion Transformer (MDDT), which predicts the sequence of codebook indices through diffusion. The diffusion process begins with no knowledge of the output, so all indices in the sequence are masked with zeros as unknown. At each step, the transformer predicts all masked indices simultaneously, and only predictions with high confidence are retained. Low-confidence predictions are discarded and re-masked for subsequent iterations until the entire sequence is reliably predicted. I adopt a deconstructed approach to accelerate the diffusion process. Finally, the output 3D human motion sequence is reconstructed by inverse-projecting the indexed codebook entries through the decoder.

In addition, I introduce the Multi-task Scene-aware Text-to-Motion (MTSA-T2M) framework, which explores scene-aware motion generation without explicitly building a unified multimodal model of scenes and motions. The approach begins by using a large language model (LLM) to decompose a long prompt into multiple shorter prompts. For each segment, the LLM performs path planning based on the provided scene road map and height map, thereby generating appropriate motion guidance.

These guidance signals are then fed into a pre-defined Aligned Motion Diffusion Model to produce short segments of scene-aware motions. Finally, the generated motion segments are smoothly connected through motion interpolation, completing the full multi-task scene-aware motion generation process.

1.3 Contributions

This thesis introduces two novel frameworks, MDD and MTSA-T2M, to address key challenges in 3D human motion generation from text and scenes. The proposed approaches tackle issues of accuracy, diversity, efficiency, and scene-awareness in a systematic manner.

The first part of the thesis focuses on the Masked Deconstructed Diffusion (MDD) framework. MDD bases motion generation on diffusion, which has shown great success in visual generation tasks [33]. Compared to deterministic one-to-one mappings and naive cross-modality alignment [81, 70], diffusion models naturally capture ambiguity through stochastic sampling, better handling the many-to-many relationships between texts and motions. Furthermore, instead of applying diffusion directly to raw motion data [29], MDD leverages multi-staged KCQ to learn compact and expressive latent motion representations. The derived codebook is more efficient than multi-codebook designs [27, 60, 97] and provides richer diversity than single VQ-VAE representations [26, 13, 25]. In addition, the deconstructed diffusion operates on codebook indices, making it lightweight and scalable. Unlike auto-regressive methods [29], my masking mechanism incorporates global context at each step, with low-confidence predictions re-masked until convergence, resulting in faster generation and adaptability to varying sequence lengths [11, 27, 52].

The second part of the thesis presents the Multi-task Scene-aware Text-to-Motion (MTSA-T2M) framework. MTSA-T2M automatically decomposes long prompts into sub-tasks using large language models (LLMs), which perform path planning with scene road maps and height maps to produce appropriate motion guidance. These guidance signals are then integrated into an Aligned Motion Diffusion Model to generate coherent short motion segments. This design achieves scene-aware motion generation while maintaining a clear separation between motion synthesis and scene reasoning.

I summarize the major contributions of this thesis as follows:

- I propose MDD, a diffusion-based framework for generating high-fidelity and expressive human motions from text descriptions;
- I present multi-staged KCQ for learning motion representations and deriving a compact yet diverse codebook;
- I design a novel transformer that incorporates masking and deconstructed diffusion to achieve efficient index prediction for motion generation;
- I conduct comprehensive evaluations showing that my results are on par with or surpass the state-of-the-art across multiple metrics, and demonstrate practical applications in VR settings;
- I introduce the MTSA-T2M framework, which explores the possibility of generating multi-task scene-aware motions without requiring unified multimodal models.

1.4 Organization

The remaining dissertation is organized as follows:

- Chapter 2: Background knowledge of diffusion-related techniques for human motion generation.
- Chapter 3: A literature review of related methods for human motion generation.
- Chapter 4: Data representation and problem formulation.
- Chapter 5: Details of the MDD framework, including Kinematic Chain Quantization (KCQ), the Masked Deconstructed Diffusion Transformer (MDDT), and the multi-step inference strategy.
- Chapter 6: Experimental evaluations of the approaches developed in Chapters 5.
- Chapter 7: Details of the Multi-task Scene-aware Text-to-Motion (MTSA-T2M) framework and its experimental visualization results.
- Chapter 8: Thesis conclusion and discussion of future directions.

Chapter 2

Background on Diffusion for Text-to-Motion Generation

In Chapter 2, I review techniques commonly used for diffusion models that are particularly relevant to the text-to-motion generation task. The discussion centers on three aspects: (1) **Vector Quantized Variational Autoencoders (VQ-VAE)**, which are employed to simplify the latent space (Section 2.1); (2) the foundation of diffusion modeling, namely **Denoising Diffusion Probabilistic Models (DDPM)** (Section 2.1); and (3) a widely used text-to-motion framework, **MDM**, which also serves as a key component in my proposed Multi-task Scene-aware Text-to-Motion approach (Section 2.3).

2.1 Vector Quantized Variational Autoencoders

A Vector Quantized Variational Autoencoder (VQ-VAE) is a generative model that extends the classical VAE framework by introducing discrete latent variables through

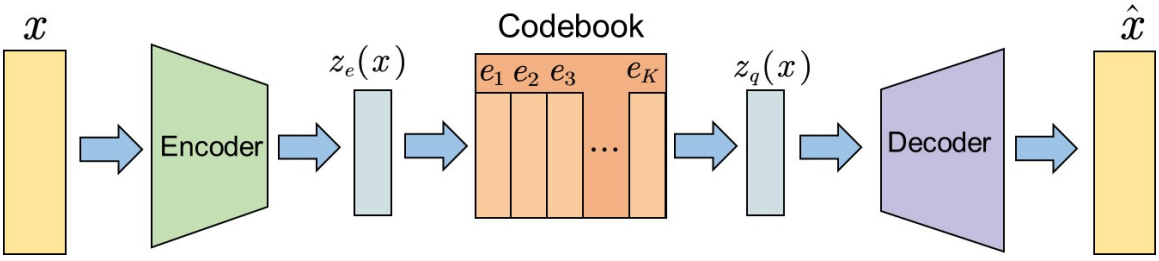

Figure 2.1: Framework of VQ-VAE [82].

vector quantization. Unlike standard VAEs that rely on continuous Gaussian-distributed latents, VQ-VAE encodes the input into a latent space where each point is replaced by the closest entry from a learned embedding dictionary. This design prevents the common "posterior collapse" issue observed in traditional VAEs, where powerful decoders tend to ignore latent variables [82].

As illustrated in Figure 2.1, the model is composed of three key components: an encoder network $\mathcal{E}$ that maps the input x into latent representations $z_e(x)$, a codebook $\mathcal{C} = \{e_1, e_2, ..., e_K\}$ with K embedding vectors, and a decoder $\mathcal{D}$ that reconstructs the input from the quantized embeddings. The discrete latent variable $z_q(x)$ is obtained by nearest-neighbor lookup:

$$z_q(x) = e_k, \quad k = \arg\min_j \|z_e(x) - e_j\|_2^2, \tag{2.1.1}$$

where e_k is the embedding vector closest to $z_e(x)$. This quantization step acts as a bottleneck that forces the encoder output to commit to a discrete code, while the decoder learns to reconstruct x from $z_q(x)$.

The overall training objective balances three terms: the reconstruction loss, which updates encoder and decoder parameters; the codebook loss, which moves embedding vectors closer to the encoder outputs; and a commitment loss, which prevents encoder

outputs from fluctuating excessively. The combined loss function can be written as:

$$\mathcal{L} = \log p(x|z_q(x)) + \|\text{sg}[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - \text{sg}[e]\|_2^2, \tag{2.1.2}$$

where sg[·] denotes the stop-gradient operator and β is a weight controlling the commitment cost.

By compressing continuous motion or image data into a sequence of discrete tokens, VQ-VAE makes it possible to model long-term structures using powerful autoregressive or diffusion priors in the discrete latent space. This discrete representation not only reduces the dimensionality of the data but also captures high-level semantics, enabling more effective generation in tasks such as image synthesis, speech modeling, and text-to-motion learning. For motion diffusion in particular, VQ-VAE plays a crucial role by converting high-dimensional motion sequences into compact, semantically meaningful codebook indices. This transformation simplifies the diffusion process, reduces computational cost, and allows the model to focus on learning the distribution of motion primitives rather than raw joint trajectories, ultimately improving both efficiency and controllability in text-to-motion generation.

2.2 Denoising Diffusion Probabilistic Models

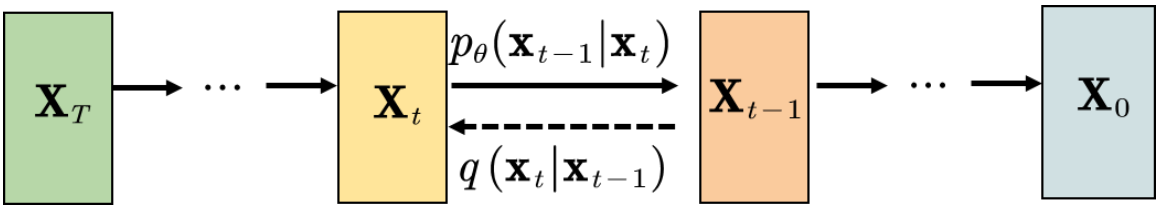

Figure 2.2: The forward diffusion and reverse denoising of DDPM [62]

Denoising Diffusion Probabilistic Models (DDPM) are a class of latent variable generative models that progressively add Gaussian noise to data and then learn to reverse this process to generate new samples. In the forward (diffusion) process, a clean data point $\mathbf{x}_0$ is gradually corrupted into a sequence $\{\mathbf{x}_t\}_{t=1}^T$ according to a variance schedule $\{\beta_t\}_{t=1}^T$, as illustrated in Figure 2.2:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\big(\sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\, \beta_t\mathbf{I}\big), \tag{2.2.1}$$

and equivalently, a noisy sample at step t can be drawn directly from $\mathbf{x}_0$:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\big), \quad \bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s). \tag{2.2.2}$$

The reverse process, shown together with the forward process in Figure 2.2, is parameterized by a neural network $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, often instantiated as a U-Net or Transformer, which aims to denoise step by step. Training can be simplified into a denoising score matching objective:

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2\Big], \tag{2.2.3}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\epsilon}_\theta$ is trained to predict the added noise from the corrupted sample.

DDPM has demonstrated state-of-the-art results in image generation by producing diverse and high-fidelity samples. For motion generation tasks, diffusion models provide two main benefits: (1) Stable training, avoiding mode collapse commonly observed in GANs, and (2) Flexible conditioning, where text, audio, or scene information can be injected at each denoising step.

However, standard DDPM frameworks also suffer from drawbacks, such as slow sampling speed due to the large number of denoising iterations and difficulties in balancing efficiency with motion quality. These limitations motivate the improvements proposed in this thesis, which are introduced and discussed in the following chapters.

2.3 MDM: Human Motion Diffusion Model

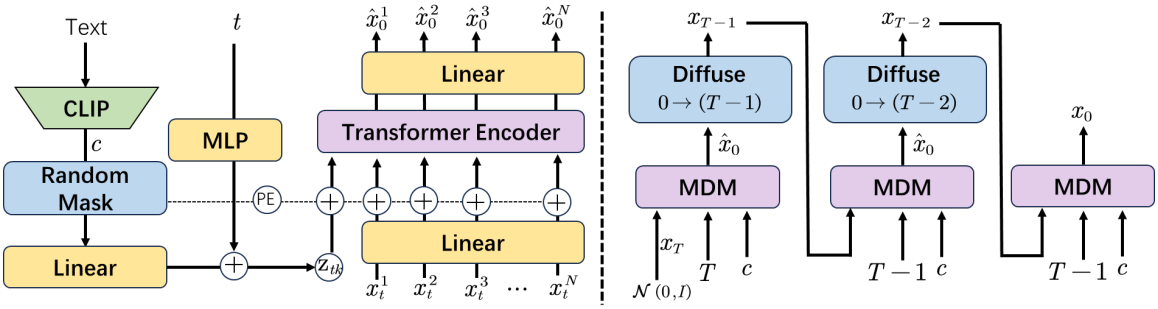

Figure 2.3: Framework of MDM [29]

The Human Motion Diffusion Model (MDM) is a diffusion-based generative framework specifically designed for the motion domain [29]. Unlike earlier autoencoder or VAE-based methods that often restrict motion representations to simplified latent distributions, MDM leverages diffusion to better capture the inherent many-to-many mapping between textual conditions and motion sequences.

A motion sequence is represented as $\mathbf{x}_0 = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}$, where each frame $\mathbf{p}_i$ contains joint-level information such as positions or rotations. As illustrated in Figure 2.3, the forward process gradually perturbs the clean sequence into noisy states $\mathbf{x}_t$, while the reverse process employs a transformer-based network to iteratively denoise the sequence. Unlike standard DDPM, MDM directly predicts the clean motion $\mathbf{x}_0$ at

each denoising step instead of the noise $\boldsymbol{\epsilon}$, which enables the incorporation of motion-specific geometric losses such as foot-contact constraints [74] or velocity regularization [69] to improve realism.

Conditioning is applied in a classifier-free manner. For text-to-motion generation, the text prompt $\mathbf{c}$ is first embedded using a CLIP encoder and then combined with the timestep embedding t. These conditioning signals are injected into the transformer denoiser, which progressively reconstructs motion sequences $\mathbf{x}_0$ that align semantically with the input description. As shown in Figure 2.3, this iterative process allows the model to sample both conditional and unconditional motions within the same framework.

In addition to text-to-motion, MDM naturally extends to related tasks such as action-to-motion generation, motion completion, and motion editing (e.g., in-betweening or partial body control). This versatility makes MDM a strong baseline for controllable motion synthesis. Nevertheless, its reliance on a large number of denoising steps results in high inference cost, which limits efficiency. In this thesis, I build upon MDM by adapting it as a module within the proposed **Multi-task Scene-aware Text-to-Motion** framework, where it is used to generate scene-consistent motion segments from textual prompts.

Chapter 3

Literature Review

In this chapter, I lay out the background needed for the methods that follow. Section 3.1 reviews core AI paradigms for learning 3D human motion, from early fully connected and recurrent models to attention-based architectures and physics-aware control. Section 3.2 summarizes conditional generation, where motions are guided by trajectories, styles, references, or other input signals across VR and interactive settings. Section 3.3 focuses on text-to-motion, outlining latent alignment, tokenized representations, and diffusion-based approaches that translate language into movement. Section 3.4 turns to scene-aware synthesis, highlighting how motion generation is coupled with layout and object geometry to support navigation and physically plausible interactions.

3.1 AI for 3D Human Motion Learning

With the rapid advancement of artificial intelligence, deep learning methods have become the mainstream paradigm for learning and synthesizing 3D human motions.

Early work by Taylor et al. [79] introduced the Factored Conditional Restricted Boltzmann Machine (FCRBM), which modeled motion dynamics through stacked network layers. Early frameworks often relied on fully-connected architectures for autoregressive motion generation. For example, Holden et al. [36] developed Phase-Functioned Neural Networks (PFNN) to drive real-time locomotion in VR scenes across various terrains. Deep autoencoders [34, 85] further enabled the extraction of latent features in a compact manifold, facilitating retrieval and reconstruction of diverse human motions for VR and gaming applications. By stacking fully-connected layers on top of autoencoders, Holden et al. [35] generated locomotion sequences with accurate foot placement and path-following ability.

Recurrent Neural Networks (RNNs) [61, 86, 84], as well as their variants such as GRUs [23] and LSTMs [53, 78], have been widely adopted to model temporal motion dynamics. More recently, attention mechanisms and Transformer-based models [46, 47, 8] have gained prominence for their ability to capture long-range dependencies, making them particularly effective for generating complex free-form motions such as gestures or dance.

In the context of physics-based simulations, deep reinforcement learning (DRL) has been extensively used to generate physically realistic human skills. Representative works include DeepMimic [66] and Adversarial Motion Priors (AMP) [68], which synthesize a wide variety of motions via imitation and adversarial training. Beyond locomotion and navigation, DRL has been applied to balancing [56], basketball dribbling [57], and acrobatics [66]. In addition, adversarial generative models such as GANs have also been employed [50] to improve diversity and realism in synthesized motions.

3.2 Conditional Human Motion Generation

Unlike unconstrained motion synthesis, conditional motion generation focuses on producing motions that satisfy task-specific or user-defined requirements, which is of particular importance in VR/AR and interactive entertainment. Spatial conditioning is one common approach, enabling the system to enforce trajectory-following [35], terrain-aware navigation [36], or reference motion imitation [66].

Another direction is style-conditioned motion synthesis, where users specify both a target content and a reference style. Style transfer can then be realized using autoregressive mixtures [89], frequency-domain amplitude modulation [95], fully-connected neural models [76], GRAM matrices [35], or Adaptive Instance Normalization (AdaIN) [1]. With recent progress in human pose estimation, some approaches [67, 1] are even able to extract motion conditions from 2D videos to guide the generation of corresponding 3D outputs.

Multimodal conditioning has also been explored, particularly in audio-driven motion generation. Early works modeled gesture synthesis from speech using HMMs [44] and CRFs [45]. Leveraging large-scale speech–gesture datasets [20, 23, 54, 55], later studies proposed supervised learning with CNNs [30], VQ-VAEs [5], GRUs [23], LSTMs [5], and Transformers [8]. To better capture the many-to-many mappings between speech and gestures, GANs [3, 23, 31, 48, 58, 92, 42, 21] and diffusion models [6, 4] have been applied to enhance variation and realism. Similar frameworks are adopted for music-driven dance synthesis, where CNNs [43, 104, 91], VAEs [65], RNNs [78, 2], LSTMs [37, 17, 7], AdaIN [7], and Transformers [15, 16, 51, 75, 46, 47] have been successfully employed to choreograph realistic dance motions from music inputs. Unlike speech-driven gestures, music-driven motion typically requires less semantic

alignment.

3.3 Text-to-Motion Generation

Text-to-motion is a specific subtask of conditional motion generation where natural language descriptions serve as conditioning signals. To capture semantic information, CLIP embeddings [72] are widely adopted. One representative line of work aligns text and motion in a shared latent space. For example, TEMOS [70] employs transformer-based VAEs to project paired motion–text samples into a joint space, allowing motion generation via stochastic latent sampling. MotionCLIP [81] extends this idea by aligning motion with the CLIP space through transformer-based autoencoding and rendered motion images, enabling downstream tasks such as motion interpolation and editing. However, the discrepancy between text and motion distributions often leads to artifacts and reduced diversity [13].

Other approaches focus on token-based representations. TM2T [26] formulates bidirectional mappings between text and motion tokens using neural machine translation, enabling both text-to-motion and motion-to-text generation. To encourage diversity, Guo et al. [24] decomposed the task into text-to-length prediction and length-conditioned text-to-motion synthesis with GRU-based VAEs. MoMask [27] further advanced quantized representations by introducing hierarchical residual vector quantization (RVQ), where motion tokens are predicted through masked and residual transformers.

With the success of diffusion models in computer vision, several works have applied diffusion to text-to-motion generation [29, 13, 40, 99, 73, 18]. For instance, the Human Motion Diffusion Model (MDM) [29] adopts a transformer encoder for denoising-based

motion generation. MotionDiffuse [99] improves controllability through cross-modal linear transformers and body-part/time-varying controls. Guided Motion Diffusion (GMD) [40] emphasizes spatial constraints through dense propagation mechanisms. MoFusion [18] integrates textual and musical conditioning into a 1D UNet-based diffusion model, ensuring quality with kinematic and temporal consistency losses. Meanwhile, Motion Latent Diffusion (MLD) [13] proposes to operate diffusion in a compact latent space learned by a transformer-based VAE, which enhances both efficiency and motion quality.

3.4 Scene-Aware Motion Generation

Scene-aware motion generation moves beyond character-only synthesis. The goal is to respect scene geometry and semantics and affordances. One line of work uses affordance as an intermediate cue. Afford-Motion first predicts language-guided affordance maps and then generates motions that satisfy these cues [87]. GenZI targets zero-shot generation. It distills priors from vision–language models and produces text- and location-aware interactions without dedicated 3D HSI training data [49].

Another direction splits the task into object grounding and motion synthesis. Text–Scene–Motion grounds the referenced object and then produces object-centric motions in complex indoor layouts [10]. TeSMo [93] further advances this line by enabling text-controlled, scene-aware human-object interaction motions in diverse 3D environments [93]. TRUMANS scales data size and sequence length. It provides a large scene-aware corpus and an autoregressive diffusion backbone for long-horizon interactions with strong cross-scene generalization [39]. UniHSI aims for unified control. It uses a language-driven chain-of-contacts planner and a single controller to

realize long-horizon and fine-grained interactions across diverse layouts [90].

Foundational efforts shape today's landscape. COINS [103] establishes compositional control over action–object specifications and supports retargeting to unseen combinations. SceneDiffuser frames generation and optimization and planning directly in 3D scenes through diffusion [38]. Work from graphics and 3D vision offers strong priors. The Neural State Machine models physically plausible character–scene interactions with learned controllers [77]. PLACE [102] learns proximity and contact distributions that guide feasible placements in 3D environments . Recent systems stress physical plausibility and full-pipeline consistency. InterScene synthesizes physically plausible motions in 3D scenes [64]. HUMANISE explores language-conditioned interaction in 3D scenes [88]. Together these studies advance robust and controllable scene-aware motion generation that scales across objects and scenes and tasks.

Chapter 4

Problem Formulation

In this chapter, I first introduce the data resources used throughout this thesis, which include annotated human motion datasets and scene dataset. These provide the foundation for training and evaluating motion generation models. I then formulate the research problems considered in my work, starting from text-to-motion generation, extending to address the challenge of producing multi-task sequences guided by complex user instructions and realistic scene constraints.

4.1 Data Representation

4.1.1 Annoated Motion Datasets

I use text–motion corpora that pair natural language descriptions $\mathbf{c}$ with 3D human motions $\mathbf{m}$. Following the HumanML3D [25] convention, each motion is represented in a compact per–frame format $\mathbf{m} \in \mathbb{R}^{N \times D}$ with N frames and a D-dimensional attribute vector $\left[\mathbf{r}_h, \mathbf{r}_{rv}, \mathbf{r}_{lv}, \mathbf{j}_r, \mathbf{j}_{lp}, \mathbf{j}_v, \mathbf{f}\right]$ per frame, where $\mathbf{r}_h$ is the root height, $\mathbf{r}_{rv}$ the

root rotation velocity, $\mathbf{r}_{lv}$ the root linear velocity, $\mathbf{j}_r$ the joint rotations, $\mathbf{j}_{lp}$ the joint local positions, $\mathbf{j}_v$ the joint velocities, and $\mathbf{f}$ the binary foot–contact indicators. This unified representation facilitates training motion models directly in the attribute space while maintaining temporal and structural consistency across frames. HumanML3D

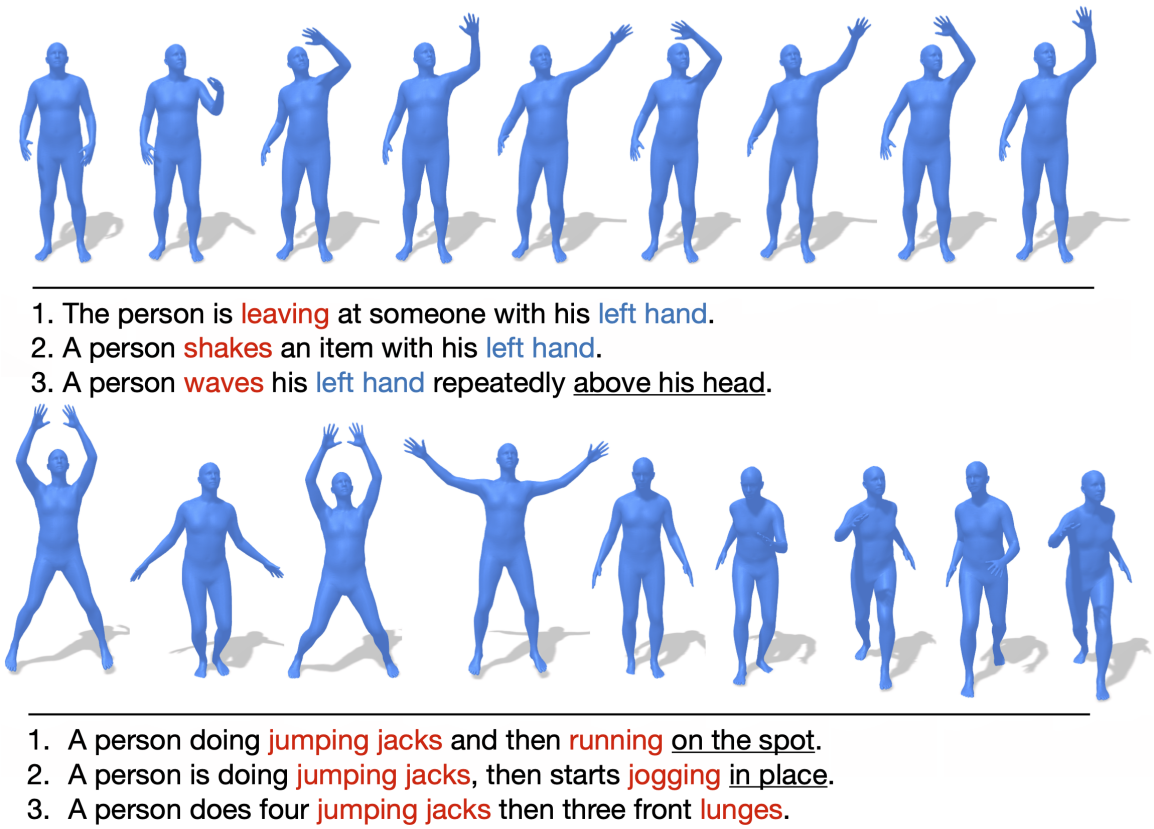

Figure 4.1: Samples of HumanML3D dataset [25]

is sizeable and diverse: it contains 14,616 motion clips paired with 44,970 descriptions spanning 5,371 unique words. The motion set totals 28.59 hours, with clips averaging 7.1 seconds (range 2–10 s). Texts are concise, with mean and median lengths of 12 and 10 tokens. As shown in Figure 4.1 HumanML3D pairs each motion with multiple free–form sentences covering action type, body part, direction, and style. Typical captions include short single–action prompts such as "*shakes an item with his left*

hand" or "*waves his left hand repeatedly above his head*," as well as multi–clause instructions like "*jumping jacks and then running on the spot*," "*jumping jacks then starts jogging in place*," and "*does four jumping jacks then three front lunges*." These examples reflect the many–to–many relation between text and motion and motivate the multi–task handling in later chapters. In addition, the HumanML3D authors release KIT–ML [71] in the same processed formats, enabling direct reuse of the notations above for cross–dataset training and evaluation.

4.1.2 Scene Datasets

I adopt the scene-aware data released with TSTMotion [28], which is built on the HUMANISE [88] corpus of human–scene interactions in furnished indoor scans. As shown in Figure 4.2. HUMANISE provides language descriptions aligned to human motions $\mathbf{m}$ performed within realistic 3D environments, together with per-scene object instances and geometry. This pairing of text $\mathbf{c}$, motion, and scene context enables evaluating scene-aware generation.

TSTMotion applies a Scene Compiler to HUMANISE scenes to convert raw scans and instance annotations into two rasterized products that are convenient for downstream use as shown in Figure 4.3. First, a scene-wide road map $\mathbf{R} \in \{0, 1, 2\}^{H \times W}$ encodes horizontal traversability: cells with value 0 are walkable, value 1 are obstacles (walls, furniture footprints, voids), and value 2 mark target regions referenced by the captions. The grid is defined in the ground plane with a fixed cell size, so that $\mathbf{R}$ uniformly represents room layout across scenes. Second, instead of a single global height raster, the compiler produces an upper-surface height map for every object instance. For each candidate object o, its 3D bounding box is projected onto

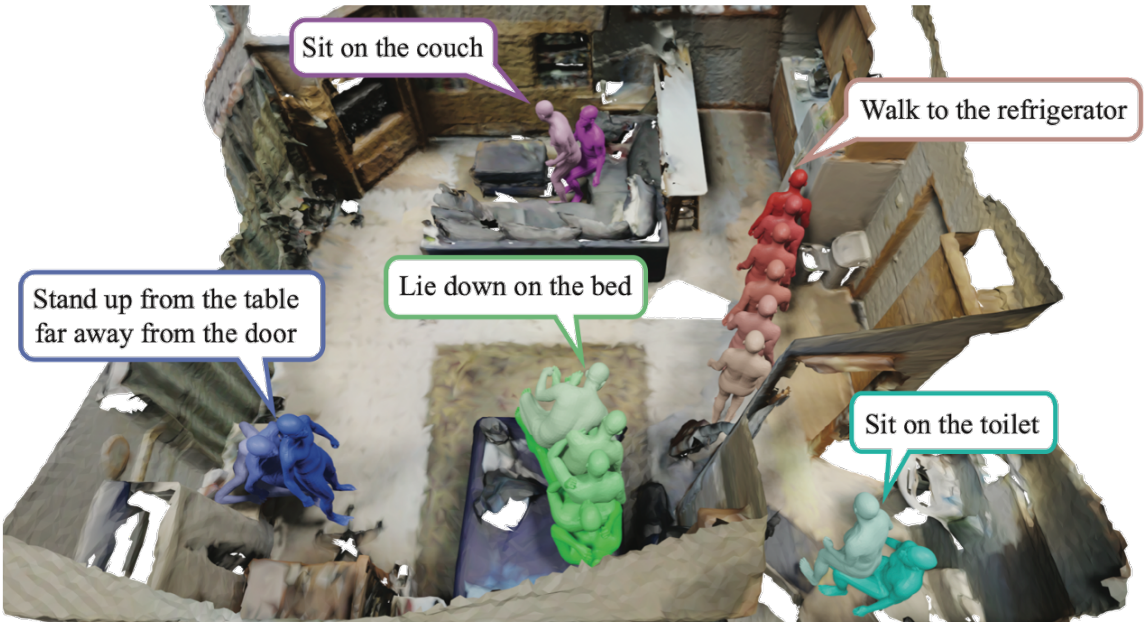

Figure 4.2: Samples of HUMANISE dataset. [88]

the scene coordinate system, where x and y represent horizontal floor positions and z denotes the vertical height. Based on this projection, an object-level height map $\mathbf{H}^o \in \mathbb{R}^{h_o \times w_o}$ is computed over the footprint of the object's bounding box, with each cell corresponding to a discrete (x, y) location and storing the elevation of the object's top surface along the z axis. The resolution (h_o, w_o) is determined by the physical footprint of the object, providing detailed surfaces for large items such as beds or tables, while maintaining compact grids for smaller objects.

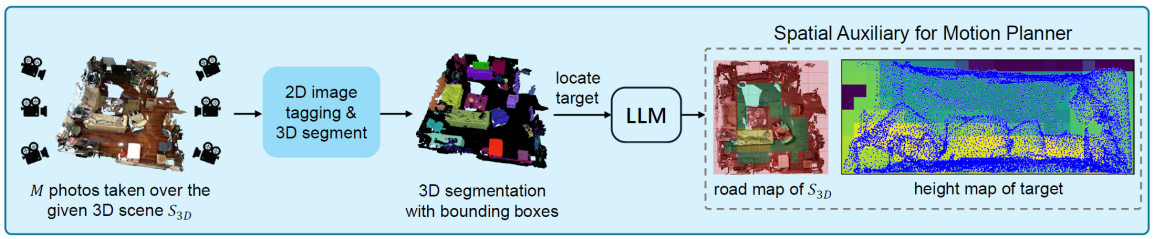

Figure 4.3: TSTMotion Scene Compiler [10]

The released scene-aware dataset therefore contains, for each scene: the rasterized road map $\mathbf{R}$, a collection of per-object surface maps $\{\mathbf{H}^o\}$ covering all annotated objects, and the original HUMANISE links between motions $\mathbf{m}$, texts $\mathbf{c}$, and the scenes in which they occur. This standardized representation supports experiments that require both global travel information from $\mathbf{R}$ and precise surface geometry from $\mathbf{H}^o$.

4.2 Problem Statement

4.2.1 Text-to-Motion

Let $\mathcal{D} = \{(\mathbf{c}_i, \mathbf{m}_i)\}_{i=1}^{M}$ denote the paired text–motion dataset, where each motion $\mathbf{m} \in \mathbb{R}^{N \times D}$. We aim to learn a stochastic generator

$$f_{\theta_1} : (\mathbf{c}, \mathbf{z}) \mapsto \widehat{\mathbf{m}}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

that models the conditional distribution $p_{\theta_1}(\mathbf{m} \mid \mathbf{c})$

The training goal is simply to make the model distribution close to the data distribution under the same text:

$$\min_{\theta_1} \ \mathbb{E}_{\mathbf{c}}\Big[\, \Delta\big(p_{\theta_1}(\mathbf{m} \mid \mathbf{c}) \parallel p_{\mathcal{D}}(\mathbf{m} \mid \mathbf{c})\big) \,\Big],$$

where $\Delta(\cdot\|\cdot)$ denotes a suitable discrepancy (e.g., likelihood-based or distributional distance). Equivalently, one may maximize the conditional log-likelihood:

$$\max_{\theta_1} \ \frac{1}{M} \sum_{i=1}^{M} \log p_{\theta_1}(\mathbf{m}_i \mid \mathbf{c}_i).$$

Beyond distributional closeness, the generator is expected to produce, for any given $\mathbf{c}$: (i) motions that are semantically consistent with the text, (ii) natural and physically reasonable sequences with smooth kinematics and sensible contacts, and (iii) diverse samples when queried multiple times with the same $\mathbf{c}$.

4.2.2 Multi-task Scene-aware Text-to-Motion

Given a scene and a long instruction, the goal is to generate a sequence of scene-aware motions that follow the instruction while respecting the scene layout. Let the scene be represented by a road map $\mathbf{R} \in \{0, 1, 2\}^{H \times W}$ and a set of object top-surface height maps $\{\mathbf{H}^o\}_{o=1}^{O}, \mathbf{H}^o \in \mathbb{R}^{h_o \times w_o}$. Let $\mathbf{c}$ be the user text, which may implicitly contain multiple sub-tasks $[\mathbf{c}_i]_{i=1}^{C}$ The objective is The objective is to design a generator

$$f_{\theta_2} : (\mathbf{c}_i, \mathbf{R}, \{\mathbf{H}^o\}, \boldsymbol{\xi}_i) \mapsto \widehat{\mathbf{m}}_i,$$

that generates a scene-aware motion segment $\widehat{\mathbf{m}}_i$ corresponding to each sub-task $\mathbf{c}_i$, where $\boldsymbol{\xi}_i$ is a stochastic seed enabling multiple plausible outcomes. The full motion sequence is obtained by temporally concatenating the set $[\widehat{\mathbf{m}}_i]_{i=1}^{C}$.

The optimization objective is to make the generated motions consistent with the dataset distribution under the same text and scene conditions. Beyond matching the data distribution, the generator is expected to offer diversity through the noise $\boldsymbol{\xi}_i$, and to remain natural and physically reasonable within the constraints implied by $\mathbf{R}$ and $\{\mathbf{H}^o\}$. These properties are promoted by stochastic sampling, soft scene-consistency penalties at sampling time.

t

Chapter 5

Masked Deconstructed Diffusion for Text-to-Motion

5.1 Overview

I formulate the problem of text-to-motion generation as follows: given a text condition $\mathbf{c}$, the goal is to synthesize a motion sequence $\mathbf{m} \in \mathbb{R}^{N \times D}$ consisting of N frames. Each frame is represented by D-dimensional pose attributes $[\mathbf{r}_h, \mathbf{r}_{rv}, \mathbf{r}_{lv}, \mathbf{j}_r, \mathbf{j}_{lp}, \mathbf{j}_v, \mathbf{f}]$, where $\mathbf{r}_h$ denotes root height, $\mathbf{r}_{rv}$ the root rotation velocity, $\mathbf{r}_{lv}$ the root linear velocity, $\mathbf{j}_r$ the joint rotations, $\mathbf{j}_{lp}$ the joint local positions, $\mathbf{j}_v$ the joint velocities, and $\mathbf{f}$ the binary foot-contact indicators.

As illustrated in Fig. 5.1, the proposed framework is composed of three major components: 1) **Kinematic Chain Quantization (KCQ)** encodes body-part-specific motion features into latent representations, which are subsequently discretized into vectors and stored as entries in a codebook (Section 5.2); 2) during training, the **Masked Deconstructed Diffusion Transformer (MDDT)** learns to align text

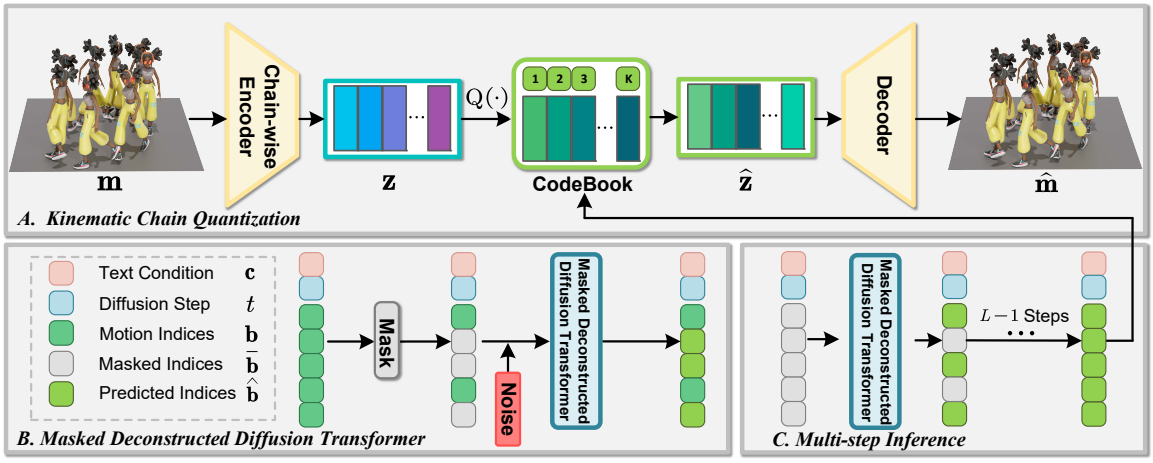

Figure 5.1: **MDD framework overview.** A. Kinematic Chain Quantization: The original motion sequence undergoes precise quantization through a chain-wise encoder designed based on human kinematic structure, producing an efficient latent vectors codebook with code entries referenced by their indices. B. Masked Deconstructed Diffusion Transformer: The masked motion indices, text condition, and diffusion step are noised in the latent space. The transformer is trained to predict the complete and clean index sequence, enabling a deconstructed diffusion process. C. At inference time, the text condition guides the transformer to generate indices, according to which, codes are retrieved from the codebook and fed to the decoder, generating high-quality motion output.

conditions with motion sequences by predicting an ordered set of code indices under masked diffusion (Section 5.3); 3) during inference, the trained MDDT applies a **multi-step prediction** strategy to progressively refine code indices until a full motion sequence is reconstructed through the decoder (Section 5.4).

5.2 Kinematic Chain Quantization

I begin by revisiting the vanilla Vector Quantized-Variational AutoEncoder (VQ-VAE), which has been widely adopted in prior work on motion generation [82, 96] as a means of transforming continuous motion signals into discrete representations. Given

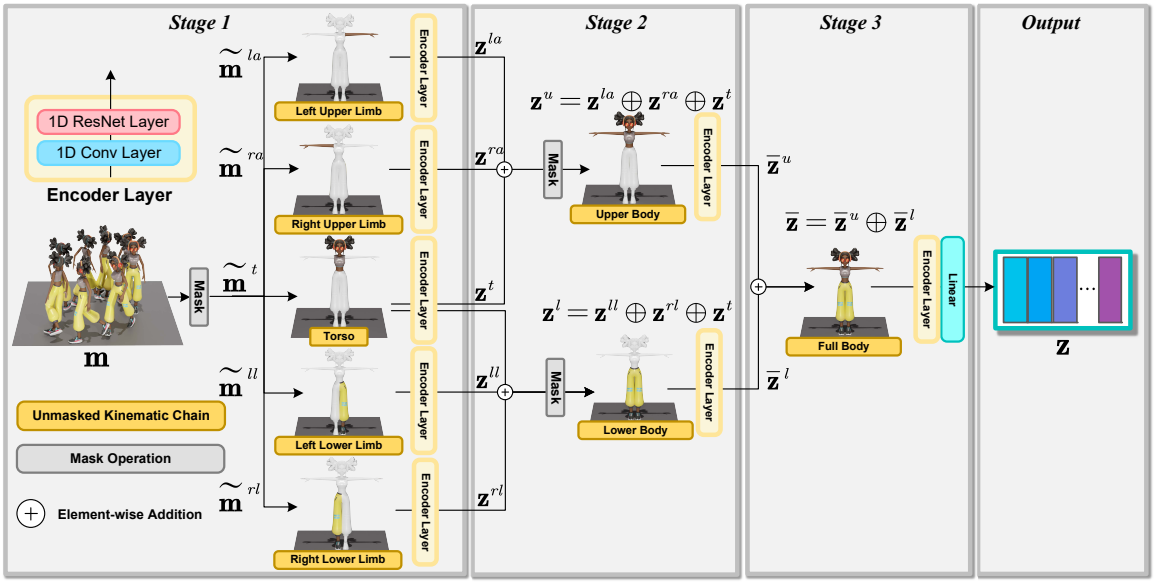

Figure 5.2: **Chain-wise Encoder.** Three stages model motion features from local to global levels based on different kinematic chains.

a motion sequence $\mathbf{m} \in \mathbb{R}^{N \times D}$, an encoder $\mathcal{E}$ composed of stacked convolutional layers projects it into a latent sequence $\mathbf{z} \in \mathbb{R}^{n \times d}$, where n and d indicate the temporal and feature dimensions in the compressed space. A quantizer $\mathrm{Q}(\cdot)$ then maps each latent vector to the closest codebook entry, yielding $\widehat{\mathbf{z}} = \mathrm{Q}(\mathbf{z}) \in \mathbb{R}^{n \times d}$. The codebook $\mathcal{C} \in \mathbb{R}^{K \times d}$ contains K entries, each representing a prototypical motion pattern. Following [82], the assignment index $\mathbf{b}_i$ for $\mathbf{z}_i$ is obtained by minimizing Euclidean distance:

$$\mathbf{b}_i = \arg\min_j \|\mathbf{z}_i - \mathcal{C}_j\|, \tag{5.2.1}$$

where $\mathcal{C}_j$ is the j-th entry in the codebook. The codebook is trained jointly with the encoder using a large set of motion examples, similar to the approach in [98].

Once quantized, the codes $\widehat{\mathbf{z}}$ are decoded by $\mathcal{D}$ to reconstruct the motion $\widehat{\mathbf{m}} = \mathcal{D}(\widehat{\mathbf{z}})$. Although this process enables motion data to be discretized and modeled under a Gaussian prior, standard VQ-VAE representations encode the entire body as

a single unit, which reduces efficiency and limits diversity across different body parts.

To address this limitation, I introduce **Kinematic Chain Quantization (KCQ)**. Unlike the conventional VQ-VAE, KCQ employs a chain-wise encoder $\mathcal{E}$ that separately extracts local latent features from multiple kinematic chains before combining them into a compact codebook (see Fig. 5.2). Each motion frame $\mathbf{m} \in \mathbb{R}^{N \times D}$ carries D-dimensional information including joint rotations, positions, and velocities. KCQ leverages a three-stage encoding strategy to capture both localized body-part features and holistic full-body representations.

At each stage, motion data corresponding to a specific kinematic chain $\widetilde{\mathbf{m}} \in \mathbb{R}^{N \times D}$ are isolated by zero-masking all other dimensions. The encoder layer, consisting of a 1D convolution followed by a ResNet-style 1D block [32], transforms these masked inputs into latent features $\widetilde{\mathbf{z}} \in \mathbb{R}^{\frac{N}{2} \times D}$:

$$\begin{aligned} \widetilde{\mathbf{m}}' &= \mathrm{ReLU}(\widetilde{\mathbf{m}} * W_0 + b_0), \\ \widetilde{\mathbf{z}} &= \mathrm{ReLU}\big(\mathrm{ReLU}(\widetilde{\mathbf{m}}' * W_1 + b_1) * W_2 + b_2 + \widetilde{\mathbf{m}}'\big), \end{aligned} \tag{5.2.2}$$

where W_0, b_0 are the parameters of the convolution, and W_1, b_1, W_2, b_2 belong to the ResNet block.

Stage 1: Five local kinematic chains are extracted: left arm ($\widetilde{\mathbf{m}}^{la}$), right arm ($\widetilde{\mathbf{m}}^{ra}$), torso ($\widetilde{\mathbf{m}}^{t}$), left leg ($\widetilde{\mathbf{m}}^{ll}$), and right leg ($\widetilde{\mathbf{m}}^{rl}$). Their latent codes $\mathbf{z}^{la}, \mathbf{z}^{ra}, \mathbf{z}^{t}, \mathbf{z}^{ll}, \mathbf{z}^{rl}$ are obtained by passing each through the encoder.

Stage 2: The features are fused into upper-body $\mathbf{z}^u = \mathbf{z}^{la} \oplus \mathbf{z}^{ra} \oplus \mathbf{z}^t$ and lower-body $\mathbf{z}^l = \mathbf{z}^{ll} \oplus \mathbf{z}^{rl} \oplus \mathbf{z}^t$, where $\oplus$ denotes element-wise addition. For joints contributing to both fusions (e.g., the hip), features are averaged to normalize their scale. The fused results $\mathbf{z}^u$ and $\mathbf{z}^l$ are then re-encoded into $\bar{\mathbf{z}}^u$ and $\bar{\mathbf{z}}^l$.

Stage 3: The upper- and lower-body features are further combined into $\bar{\mathbf{z}} = \bar{\mathbf{z}}^u \oplus \bar{\mathbf{z}}^l$. This representation is passed through the final encoder and a linear projection to produce the compact full-body latent vector $\mathbf{z} \in \mathbb{R}^{n \times d}$.

For reconstruction, the decoder $\mathcal{D}$ adopts the conventional VQ-VAE structure to maintain generalization ability. Training of KCQ follows [98] by minimizing both reconstruction loss $\mathcal{L}_{rec}$ and vector-quantization loss $\mathcal{L}_{vq}$:

$$\mathcal{L}_{KCQ} = \underbrace{\|\mathbf{m} - \widehat{\mathbf{m}}\|_1}_{\mathcal{L}_{rec}} + \underbrace{\alpha \|\widehat{\mathbf{z}} - \mathbf{z}\|_2^2}_{\mathcal{L}_{vq}}. \tag{5.2.3}$$

After training, the KCQ module can effectively extract multi-scale spatiotemporal features that capture different kinematic characteristics of human motion during encoding. For example, certain motions emphasize upper-body movements, while others focus more on lower-body dynamics. By modeling these variations, KCQ learns a more expressive and structured codebook, which is essential for high-quality motion reconstruction. In contrast, most existing methods that rely solely on a standard VQ-VAE are unable to achieve this level of representation learning, as demonstrated in the component analysis presented in Chapter 6 Section 6.5. Furthermore, in the subsequent conditional generation stage, one promising direction is to directly establish the mapping between motions and codebook indices, rather than between motions and latent variables, which can significantly improve inference speed.

5.3 Masked Deconstructed Diffusion Transformer

To capture the inherent many-to-many correspondence between text prompts and motion outputs, I propose the **Masked Deconstructed Diffusion Transformer**

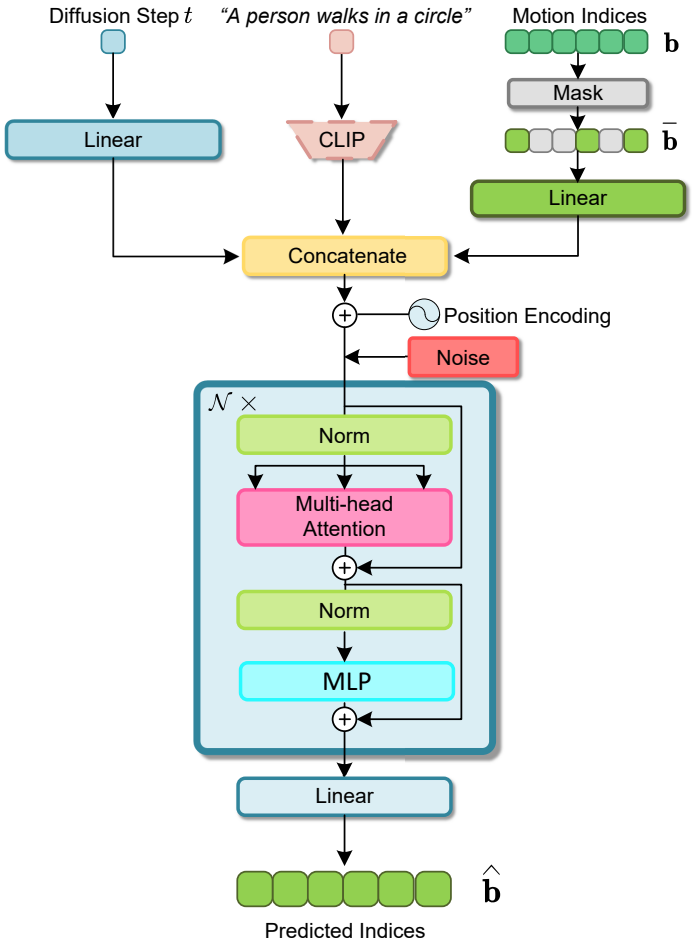

Figure 5.3: **Masked Deconstructed Diffusion Transformer** predicts clean indices based on the diffusion step, text condition, and masked indices.

(MDDT), shown in Fig. 5.3. Earlier studies mainly relied on autoregressive models [29, 25], which generate future frames sequentially from past information. Such approaches often suffer from inefficiency and compounding errors across long sequences. In contrast, I represent a motion sequence by the indices of its quantized codes $\widehat{\mathbf{z}}$ in the learned codebook $\mathcal{C}$, and adopt a masked prediction strategy [11, 27, 52] to enable parallel decoding of the entire sequence. Formally, MDDT takes as input the text condition $\mathbf{c}$, a masked index sequence $\overline{\mathbf{b}} \in \mathbb{R}^n$ for n frames, and the diffusion step t, and produces predicted indices $\widehat{\mathbf{b}} \in \mathbb{R}^n$. Text features are extracted from $\mathbf{c}$ using

CLIP [72], yielding an h-dimensional embedding.

During training, the true motion indices $\mathbf{b}$ associated with $\mathbf{c}$ are partially masked. The masking ratio is controlled by $r(\beta) = \cos\left(\frac{\pi\beta}{2}\right)$, where $\beta \sim \mathcal{U}(0, 1)$. The resulting masked indices $\overline{\mathbf{b}}$ and the diffusion step t are projected into h-dimensional embeddings and concatenated with the textual feature $\mathbf{c}$. After position encoding, this forms a matrix $\mathbf{x}_t \in \mathbb{R}^{(n+2)\times h}$ which serves as the input to the transformer [83].

A challenge of text-to-motion datasets such as HumanML3D [25] is that their textual annotations are manually labeled and may include ambiguous or noisy descriptions. To address this, previous works [41, 59] leverage Denoising Diffusion Probabilistic Models (DDPM) [33], where a clean feature $\mathbf{x}_0$ is gradually perturbed with Gaussian noise:

$$\widetilde{\mathbf{x}}_t = \gamma_t \mathbf{x}_0 + \sigma_t \epsilon, \tag{5.3.1}$$

with ϵ drawn from a Gaussian distribution, and γ_t, σ_t being schedule parameters such that $\gamma_t^2 + \sigma_t^2 = 1$, $\gamma_t^2 = \prod_{s=1}^{t}(1 - \eta_s)$ under a linear noise schedule [62]. In the reverse process, the clean signal is reconstructed by iteratively estimating and subtracting noise at each step. Although effective, this iterative denoising requires many passes, making training and inference costly [14].

Building on this idea, I develop a deconstructed diffusion scheme to accelerate the process. Instead of progressively increasing noise, I adopt a linearly decreasing schedule for γ_t^2, which reduces the number of steps while retaining useful clean features as shown in Figure 5.4. At step t, the corrupted feature $\widetilde{\mathbf{x}}_t$ is fed into an $\mathcal{N}$-layer transformer encoder that directly outputs the motion indices $\widehat{\mathbf{b}}$, bypassing explicit noise estimation.

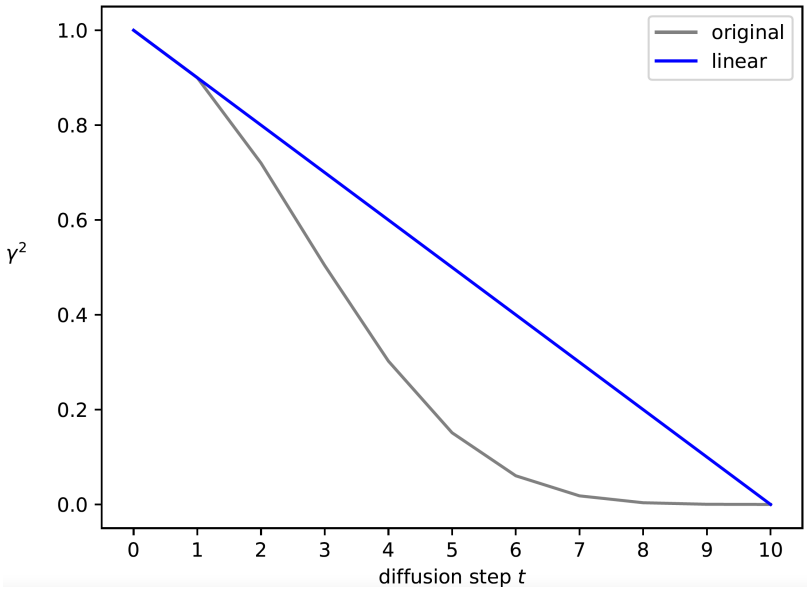

Figure 5.4: **Noise schedules.** The conventional setting defines $\gamma_t^2 = \prod_{s=1}^{t}(1-\eta_s)$ with linearly increasing η. By contrast, a linear decrease in γ_t^2 encourages the model to emphasize features with lower noise levels.

The learning objective at step t is defined as a cross-entropy loss between the masked indices $\overline{\mathbf{b}}$ and the predictions $\widehat{\mathbf{b}}$, weighted by γ_t^2 to emphasize denoising:

$$\mathcal{L}_{\text{MDDT}}^{t} = -\gamma_t^2 \cdot \mathbb{E}_{(\overline{\mathbf{b}},\widehat{\mathbf{b}},\mathbf{c},t)} \left[\log p(\widehat{\mathbf{b}} | \overline{\mathbf{b}}, \mathbf{c}, t) \right], \tag{5.3.2}$$

where $\mathbb{E}$ denotes the expectation over training samples.

In summary, I deconstruct both the forward and backward diffusion processes: the forward pass avoids excessive noise injection through a simplified schedule, and the backward pass replaces step-by-step noise removal with direct index prediction. This results in a more efficient and stable mapping between textual conditions and generated motion sequences.

5.4 Multi-step Inference

During inference, given a new text prompt, its CLIP embedding $\boldsymbol{c}$ is extracted and provided as input to the trained MDDT. The model then predicts the motion indices over L refinement steps. At the beginning ($l = 0$), the index sequence $\mathbf{b}^0$ is entirely masked. Inference proceeds iteratively, with step l ranging from 0 to L, while the corresponding diffusion step is set as $t = L - l$, decreasing from L down to 0. At each step l, the model receives the textual feature $\boldsymbol{c}$, the current index sequence $\mathbf{b}^l$, and the diffusion step t, and outputs an updated sequence $\mathbf{b}^{l+1}$. To progressively refine the predictions, $\lceil r\left(\frac{l}{L}\right) \cdot n \rceil$ indices with the lowest confidence are masked again before moving to the next iteration. After completing L steps, the final sequence $\mathbf{b}^{L+1}$ is obtained and used to retrieve quantized codes $\widehat{\mathbf{z}}$ from the codebook $\mathcal{C}$. Finally, the decoder $\mathcal{D}$ in KCQ transforms $\widehat{\mathbf{z}}$ back into the reconstructed motion sequence.

Chapter 6

Experiments and Results of Masked Deconcusted Diffusion

In this chapter, the proposed Masked Deconstructed Diffusion (MDD) framework is systematically evaluated through a series of experiments designed to assess its effectiveness in text-driven human motion generation. Section 6.1 introduces the evaluation metrics adopted in the study, followed by Section 6.2, which details the implementation settings, including the experimental environment and hyperparameter configuration. Section 6.3 presents the quantitative evaluation results, while Section 6.4 discusses the qualitative analyses, demonstrating that MDD achieves competitive accuracy and efficient inference speed while maintaining high realism. To further investigate the impact of individual components, Section 6.5 provides a detailed component analysis. Finally, Section 6.6 explores the practical applicability of the proposed framework in virtual reality (VR) environments, highlighting the flexibility and effectiveness of MDD-based motion generation for immersive VR applications.

6.1 Evaluation Metrics

Consistent with prior studies, I adopt several quantitative metrics to evaluate the proposed models:

- *R-Precision*: evaluates how well the generated motion corresponds to the input text by computing retrieval accuracy.
- *Multimodal Distance (MM-Dist)*: measures the semantic consistency between the synthesized motion and the given textual description.
- *Fréchet Inception Distance (FID)*: compares the statistical distribution of generated motions with that of real motions, serving as an indicator of overall motion realism and quality.
- *Multimodality (MModality)*: quantifies the diversity of results by examining the variation among motions produced from the same text prompt.

6.2 Implementation Details

All experiments were implemented in PyTorch 1.7.1 and conducted on an Ubuntu 20.04 workstation equipped with 32GB RAM, an Intel(R) Core(TM) i9-13900 CPU @ 2.00GHz, and an NVIDIA GeForce RTX 4080 GPU with 16GB memory. For the KCQ module, I adopted ResNet-1D blocks in both encoder and decoder, each stage using a sampling scale factor of 2. The codebook size K and embedding dimension d were both fixed to 512, and the quantization loss weight α was set to 0.08. The MDDT model consisted of six transformer layers, each with six attention heads and a hidden

dimension of 384. The number of diffusion steps t as well as inference iterations L was set to 10.

6.3 Quantitative Comparisons

I compared the proposed MDD against representative baselines, including VAE-based methods [25], VQ-VAE approaches [26, 98, 13, 27], and diffusion-based frameworks [13, 29, 101, 100]. Evaluations were performed using the aforementioned metrics together with inference runtime.

6.3.1 Metrics-based Comparison

To minimize randomness in evaluation, I followed the protocol in [25] and repeated each experiment 20 times. The reported values correspond to the mean performance with a 5% significance level. Table 6.1 summarizes the results on two datasets. In general, across 11 state-of-the-art baselines, the proposed approach consistently places within the top three on both *R-Precision* and *FID*, confirming its strong overall effectiveness. Although MoMask [27] achieves the best overall performance among existing methods, this advantage largely stems from its architectural complexity. The model employs multiple residual VQ-VAEs together with several autoregressive Transformers for motion index prediction, where the stacked modules compensate for the quantization inaccuracies that arise in each individual step. To enable a fair comparison under a comparable model scale, I also include MoMask (base), a simplified variant that uses only a single base codebook. Compared with this baseline, my framework further refines the encoder's feature extraction process and integrates a

Table 6.1: **Metrics-based evaluation on HumanML3D and KIT-ML test set.** ± indicates a 95% confidence interval. (*base*) indicates useing single codebook. **Bold** face indicates the best performance, single underline indicates the second-best, and double underline indicates the third-best.

Datasets	Methods	R Precision ↑			FID ↓	MultiModal Dist ↓	MultiModality↑
		Top 1	Top 2	Top 3			
HumanML3D	T2M [25]	$0.455^{\pm.003}$	$0.636^{\pm.003}$	$0.736^{\pm.002}$	$1.087^{\pm.021}$	$3.347^{\pm.008}$	$2.219^{\pm.074}$
	TM2T [26]	$0.424^{\pm.003}$	$0.618^{\pm.003}$	$0.729^{\pm.002}$	$1.501^{\pm.017}$	$3.467^{\pm.011}$	$2.424^{\pm.093}$
	T2M-GPT [98]	$0.492^{\pm.003}$	$0.679^{\pm.002}$	$0.775^{\pm.002}$	$0.141^{\pm.005}$	$3.121^{\pm.009}$	$1.831^{\pm.048}$
	MotionGPT [13]	$0.492^{\pm.003}$	$0.681^{\pm.003}$	$0.778^{\pm.002}$	$0.232^{\pm.008}$	$3.096^{\pm.008}$	$2.008^{\pm.084}$
	MLD [13]	$0.481^{\pm.003}$	$0.673^{\pm.003}$	$0.772^{\pm.002}$	$0.473^{\pm.013}$	$3.196^{\pm.010}$	$2.413^{\pm.079}$
	MDM [29]	-	-	$0.611^{\pm.007}$	$0.544^{\pm.044}$	$5.566^{\pm.027}$	$\mathbf{2.799}^{\pm.072}$
	MotionDiffuse [101]	$0.491^{\pm.002}$	$0.681^{\pm.001}$	$0.782^{\pm.001}$	$0.630^{\pm.001}$	$3.113^{\pm.001}$	$1.553^{\pm.042}$
	ReMoDiffuse [100]	$0.510^{\pm.005}$	$0.698^{\pm.006}$	$0.795^{\pm.004}$	$0.103^{\pm.004}$	$2.974^{\pm.016}$	$1.795^{\pm.043}$
	MoMask (base) [27]	$0.504^{\pm.004}$	$0.699^{\pm.006}$	$0.797^{\pm.004}$	$0.082^{\pm.008}$	$3.050^{\pm.013}$	$1.050^{\pm.061}$
	MoMask [27]	$\mathbf{0.521}^{\pm.002}$	$\mathbf{0.713}^{\pm.002}$	$\mathbf{0.807}^{\pm.002}$	$\mathbf{0.045}^{\pm.002}$	$\mathbf{2.958}^{\pm.008}$	$1.241^{\pm.040}$
	MDD	$0.506^{\pm.002}$	$0.702^{\pm.002}$	$0.799^{\pm.002}$	$0.076^{\pm.003}$	$3.090^{\pm.003}$	$1.236^{\pm.043}$
KIT-ML	T2M [25]	$0.361^{\pm.005}$	$0.559^{\pm.007}$	$0.681^{\pm.007}$	$3.022^{\pm.107}$	$3.488^{\pm.028}$	$2.052^{\pm.107}$
	TM2T [26]	$0.280^{\pm.005}$	$0.463^{\pm.006}$	$0.587^{\pm.005}$	$3.599^{\pm.153}$	$4.591^{\pm.026}$	$\mathbf{3.292}^{\pm.081}$
	T2M-GPT [98]	$0.416^{\pm.006}$	$0.627^{\pm.006}$	$0.745^{\pm.006}$	$0.514^{\pm.029}$	$3.007^{\pm.023}$	$1.570^{\pm.039}$
	MLD [13]	$0.390^{\pm.008}$	$0.609^{\pm.008}$	$0.734^{\pm.007}$	$0.404^{\pm.027}$	$3.204^{\pm.027}$	$2.192^{\pm.071}$
	MDM [29]	-	-	$0.396^{\pm.004}$	$0.497^{\pm.021}$	$9.191^{\pm.022}$	$1.907^{\pm.214}$
	MotionDiffuse [101]	$0.417^{\pm.004}$	$0.621^{\pm.004}$	$0.739^{\pm.004}$	$1.954^{\pm.062}$	$2.958^{\pm.005}$	$0.730^{\pm.013}$
	ReMoDiffuse [100]	$0.427^{\pm.014}$	$0.641^{\pm.004}$	$0.765^{\pm.055}$	$\mathbf{0.155}^{\pm.006}$	$2.814^{\pm.012}$	$1.239^{\pm.028}$
	MoMask (base) [27]	$0.415^{\pm.010}$	$0.634^{\pm.011}$	$0.760^{\pm.005}$	$0.372^{\pm.020}$	$2.931^{\pm.041}$	$1.097^{\pm.057}$
	MoMask [27]	$\mathbf{0.433}^{\pm.007}$	$\mathbf{0.656}^{\pm.005}$	$\mathbf{0.781}^{\pm.005}$	$0.204^{\pm.011}$	$\mathbf{2.779}^{\pm.022}$	$1.131^{\pm.043}$
	MDD	$0.424^{\pm.006}$	$0.637^{\pm.005}$	$0.765^{\pm.006}$	$0.344^{\pm.027}$	$2.933^{\pm.015}$	$1.234^{\pm.038}$

deconstructed diffusion mechanism into the Transformer for index generation. As a result, it achieves higher motion quality while maintaining comparable inference efficiency, which is particularly valuable when computational resources are limited. A detailed comparison of inference speed is presented in the following section, further underscoring the efficiency of the proposed framework.

6.3.2 Inference Speed Comparisons

I further benchmarked the inference efficiency of the proposed method against several state-of-the-art baselines, including MoMask [27], MLD [13], MotionDiffuse [101], MDM [29], and T2M-GPT [98], using an NVIDIA GeForce RTX 4080 GPU. For

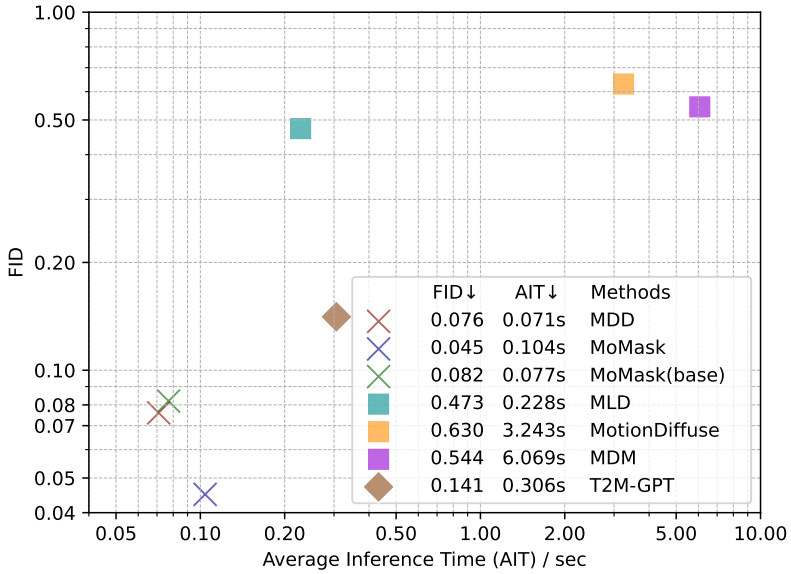

Figure 6.1: **Inference Speed Comparisons.** All experiments are conducted on an NVIDIA GeForce RTX 4080. Lower *FID* and average inference time indicate better performance.

each model, 100 motion sequences were generated and the *Average Inference Time (AIT)* was recorded as a measure of computational cost. As illustrated in Fig. 6.1, approaches that lie closer to the origin, corresponding to lower *FID* and *AIT*, are considered more favorable.

The proposed method attains a strong trade-off between runtime and quality, requiring only 0.071 seconds on average to generate a motion sequence. Compared to diffusion-based models such as MLD [13], MotionDiffuse [101], and MDM [29], my approach is faster by 2–3 orders of magnitude, owing to the fact that these methods depend on numerous denoising iterations, while the deconstructed diffusion process in my framework requires only a small number of steps. In terms of fidelity, the method also outperforms T2M-GPT [98], which is limited by its vanilla VQ-VAE design and autoregressive decoding. Finally, when compared to MoMask (base) [27],

the proposed approach provides both higher generation speed and superior motion quality.

6.4 Qualitative Comparisons

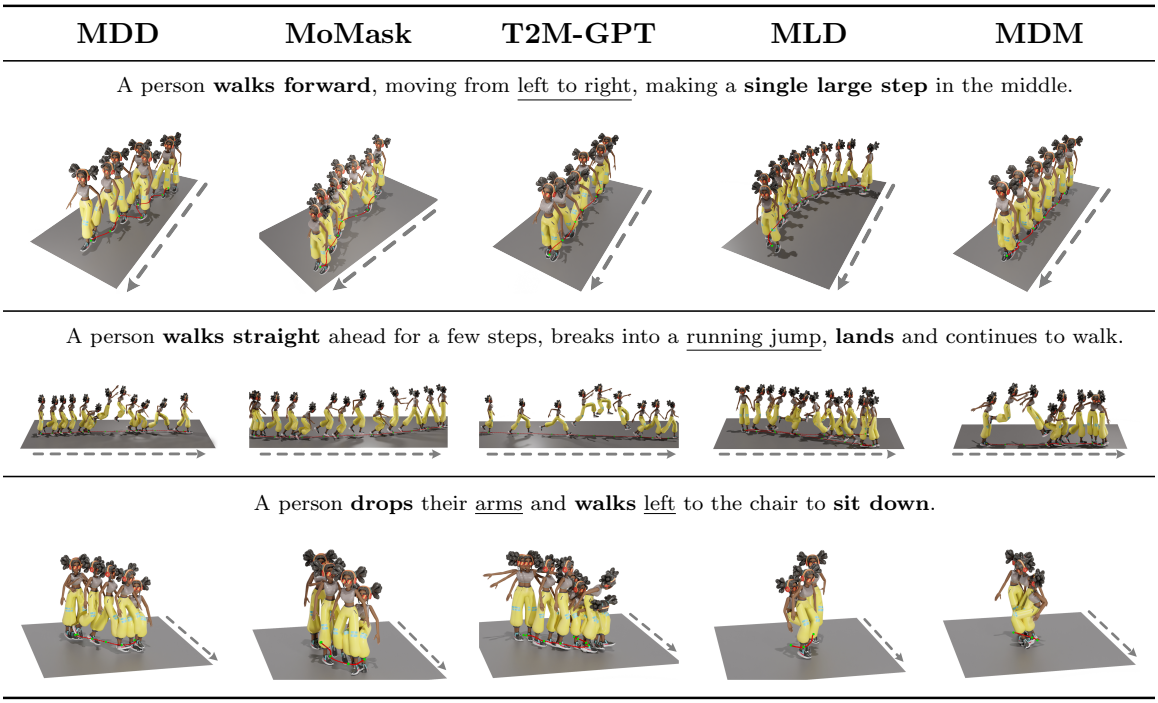

Figure 6.2: **Qualitative comparison** among multiple motion generation methods is conducted using three text descriptions from the HumanML3D dataset. Our method demonstrated superior performance in generating detailed motions and understanding long text. The axis represents the time axis, and the line indicates the motion trajectory. Please refer to the online videos [12] for dynamic visualizations.

As illustrated in Fig. 6.2, I present qualitative comparisons with MoMask [27], T2M-GPT [98], MLD [13], and MDM [29]. Overall, the proposed framework produces motions that follow textual descriptions more faithfully. In the first case, T2M-GPT [98] and MDM [29] fail to accurately realize the instruction *"moving from left to*

right". The second case focuses on longer textual inputs: MDM [29] and MLD [13] either omit required motions or introduce unnecessary turns, while MoMask [27] and T2M-GPT [98] also struggle to preserve the *"walk straight"* constraint. In the third scenario, which involves relatively simple movement, my method still demonstrates higher fidelity than competing approaches.

6.5 Component Analysis

I examined the contribution of individual components in the framework through several analyses, including a comparison of KCQ against the vanilla VQ-VAE for efficiency, as well as ablation experiments on the quantization loss weight α and the number of inference steps L.

6.5.1 KCQ Efficiency

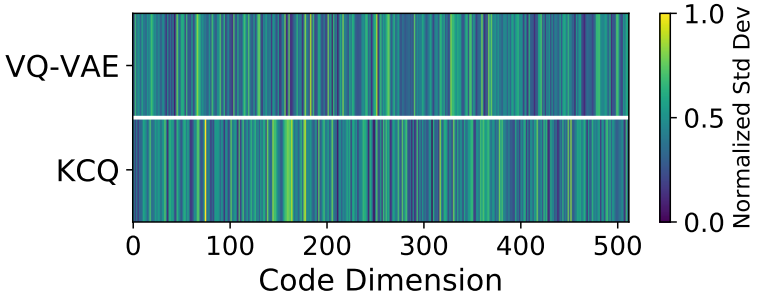

Figure 6.3: **Heatmap of the normalized standard deviation** of the codebooks from VQ-VAE and KCQ. The bright green areas represent more diverse codes.

As shown in the first section of TABLE 6.2, we evaluated the codebook efficiency of KCQ and VQ-VAE on the HumanML3D dataset with models of equivalent scale, focusing on both reconstruction and generation tasks. The evaluation metrics include

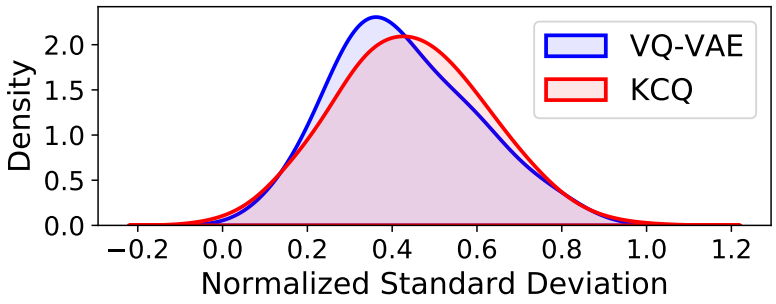

Figure 6.4: **Distribution of the normalized standard deviation** of the codebooks from VQ-VAE and KCQ, with a rightward shift in the distribution indicating that most standard deviations are larger.

FID, *Codebook Usage* (measured as the average proportion of codes used per evaluation batch) [94], and *Perplexity* [9]. For *FID*, our KCQ significantly outperforms over a single VQ-VAE, as a single VQ-VAE does not account for the structure of motion data. The *Codebook Usage* and *Perplexity* metrics indicate that KCQ's codebook is more efficient, offering greater diversity and capturing a wider range of motion features. Comparing the *normalized standard deviation* of the code dimensions in Fig. 6.3, it is evident that the codes learned by KCQ are more dispersed. Additionally, Fig. 6.4 shows that the *normalized standard deviation* in KCQ is more concentrated in a larger range, further confirming its greater diversity.

6.5.2 Quantization Loss Weight α

The second part of Table 6.2 reports results under varying quantization loss weights α. When α is too small, the reconstruction accuracy of KCQ degrades; conversely, an excessively large weight limits generalization and adversely affects the *FID* score of generated motions. These findings indicate that an appropriate trade-off between reconstruction fidelity and quantization precision is crucial during training.

Table 6.2: **Component Analysis of KCQ.** First part compares the reconstruction and generation performance with VQ-VAE. Second part focuses on ablation study of the quantization loss weight α.

Methods	Reconstruction			Generation	
	FID ↓	Codebook Usage ↑	Perplexity ↓	FID ↓	Codebook Usage ↑
Codebook Efficiency					
MDD (VQ-VAE)	$0.118^{\pm 0.001}$	77.823%	137.86	$0.124^{\pm 0.005}$	74.204%
MDD (KCQ)	$\mathbf{0.067}^{\pm 0.001}$	**79.295%**	**119.772**	$\mathbf{0.076}^{\pm 0.003}$	**76.148%**
Quantization Loss Weight α					
MDD(α, 0.02)	$0.070^{\pm 0.001}$	78.042%	134.247	$0.081^{\pm 0.004}$	74.913%
MDD(α, 0.04)	$0.102^{\pm 0.002}$	78.676%	138.934	$0.110^{\pm 0.003}$	75.525%
MDD(α, 0.06)	$0.081^{\pm 0.001}$	78.981%	136.630	$0.092^{\pm 0.004}$	75.632%
MDD(α, 0.08)	$\mathbf{0.067}^{\pm 0.001}$	**79.295%**	**119.772**	$\mathbf{0.076}^{\pm 0.003}$	**76.148%**
MDD(α, 0.1)	$0.068^{\pm 0.002}$	78.717%	133.774	$0.078^{\pm 0.005}$	75.564%
MDD(α, 0.12)	$0.076^{\pm 0.001}$	78.554%	129.276	$0.087^{\pm 0.002}$	75.422%

6.5.3 Inference Steps L

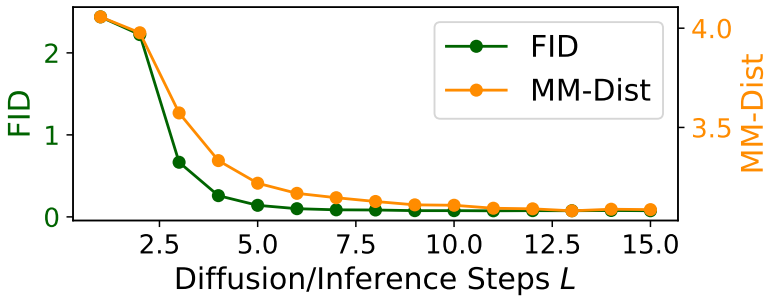

Figure 6.5: Illustration of the ablation study on diffusion/inference steps L.

Figure 6.5 presents the effect of varying the number of inference steps L on *FID* and *multimodality* distance. With too few steps, the accuracy of masked index prediction decreases and the denoising process becomes incomplete, leading to reduced motion quality. Beyond $L = 10$, however, no further improvements are observed. Therefore, $L = 10$ is adopted as the default setting in our experiments.

6.6 Applications in VR

The proposed text-to-motion framework can be applied to a variety of VR tasks, including virtual navigation, scene-level interactions, and the generation of complex motion behaviors. Given prior knowledge of the virtual characters and their environments, motions are first synthesized using the MDD framework with text prompts that define motion constraints. These motions are then retargeted to virtual characters in Blender [22] and subsequently exported into VR scenes. To improve realism and facilitate seamless integration into VR applications, I incorporated the following procedures:

1) *Global Alignment:* To ensure consistency between the generated motion and objects in the scene, keyframes and associated constraints are selected. Based on these, global repositioning and reorientation are applied to adjust the sequence to meet spatial requirements.

2) *Kinematic Refinement:* To give users more precise control over attributes such as distance, direction, and speed, the framework supports scaling, blending, and splicing of motion units derived from different prompts. This allows the enforcement of fine-grained spatial constraints in flexible ways.

3) *Dynamic Transition:* To build longer motion sequences involving multiple behaviors, smooth temporal transitions are added between motion units. Transition timing can be adaptively modified according to dynamic changes in the VR scene, resulting in more natural interactions.

To showcase the practicality and versatility of the framework across different VR use cases, several prototypes were developed in Unity3D [80].

Figure 6.6: **Application of MDD in VR.** High-quality motions generated from texts can be applied to different characters interacting with the scenes.

6.6.1 Scene Navigation

As shown in Fig. 6.6, using the prompt *"walks down some stairs"*, the framework produces a motion sequence where the character descends a staircase. With knowledge of the staircase location, *global alignment* adjusts the generated sequence to match the scene geometry. For trajectory-based navigation, locomotion clips are generated from simple directional prompts such as *"walk forward"*, *"run forward"*, or *"turn left"*. These units are refined and blended via *kinematic refinement* to satisfy spatial constraints, and further connected with *dynamic transitions* to form long, continuous navigation paths. This experiment illustrates that the proposed framework enables smooth and controllable scene traversal.

6.6.2 Scene Interaction

The framework also supports interactive motions, e.g., knocking on a door or jumping over obstacles. In the right-hand scenario of Fig. 6.6, the text prompt *"breaks into a running jump, lands."* generates a jumping motion. *Global alignment* ensures the apex of the jump coincides with the obstacle's location, allowing the character to clear

it. *Dynamic transition* is then applied to seamlessly connect the jump with preceding and subsequent motions, yielding natural interactions.

6.6.3 Non-trivial Motion Generation

By enriching prompts with stylistic adjectives, the MDD framework can produce complex, stylized motions with specific posture, speed, or amplitude preferences. For example, the middle scenario in Fig. 6.6 shows a salsa dance generated from the prompt *"A person is doing a salsa dance moving their legs and arms."* Given a stage-and-audience layout, *global alignment* places the performance at the stage center and orients the dancer toward the audience, while *kinematic refinement* allows fine control of dance tempo and stylistic details.

The results across these prototypes demonstrate the flexibility and effectiveness of MDD-based motion generation for VR applications.

Chapter 7

Multi-task Scene-aware Text-to-Motion

7.1 Overview

In this chapter, I introduce the **Multi-task Scene-aware Text-to-Motion (MTSA-T2M)** framework, which is designed to generate motion sequences that simultaneously respect textual descriptions **c** and interact coherently with a given scene **s**. Unlike conventional text-to-motion models that typically focus on producing single-task or isolated motions, MTSA-T2M enables the end-to-end synthesis of sequences containing an arbitrary number of tasks, allowing characters to perform diverse motions consistent with the scene layout.

As illustrated in Figure 7.1, the framework is composed of three core modules: (1) a **Prompt Decomposer** which could automatically segment a long and complex textual description into a sequence of shorter sub-prompts, each corresponding to a sub-task; (2) a **Sub-task Motion Planner**, which interprets each sub-prompt under

the given road map $\mathbf{R}$ and object height maps $\{\mathbf{H}^o\}$, where $\mathbf{H}^o$ denotes the height map of a single object and the braces $\{\cdot\}$ represent the set of all objects, performs spatio-temporal reasoning such as path planning on the scene map, and produces motion guidance for execution; and (3) a **Scene-Aligned Diffusion Model**, which synthesize motion segments conditioned on both the sub-task descriptions and the scene guidance, ensuring that the generated sequences are coherent with the textual intent and physically compatible with the scene.

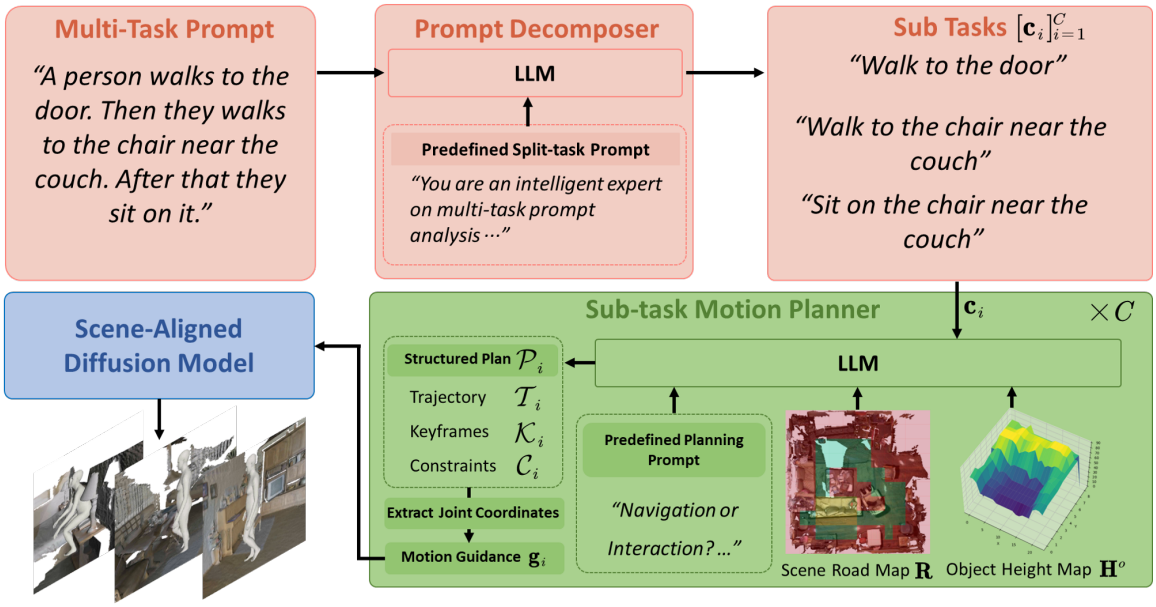

Figure 7.1: Framework of Multi-task Scene-aware Text-to-Motion (MTSA-T2M)

7.2 Prompt Decomposer

In existing text-to-motion pipelines, multi-task scenarios are often handled through manually designed stages, where each stage corresponds to a specific type of task.

While effective in restricted domains, this approach does not generalize well to open-ended user instructions, which frequently involve long and varied textual descriptions with multiple implicit actions. To overcome this limitation, I introduce the **Prompt Decomposer**, a module designed to automatically decompose complex prompts into atomic sub-tasks.

Given an input description $\mathbf{c}$ that contains multiple actions (e.g., "A person walks to the door, then they turn and walk towards the sofa. After that, they sits on the sofa."), the decomposer leverages large language models (LLMs, e.g., GPT-4o [63]) combined with carefully engineered prompts to segment the text into a sequence of shorter, semantically consistent sub-descriptions $[\mathbf{c}_i]$. Each sub-prompt corresponds to one elementary action, such as "Walk to the door" or "Sit on the sofa". The design of the decomposition prompt (see Appendix A) follows three key principles. First, it enforces atomic granularity, requiring each output to capture a single action unit. This prevents ambiguity and ensures that no two distinct actions are merged. Second, it incorporates explicit handling of temporal connectors (e.g., "then", "after that", "and"), treating them as natural breakpoints for segmentation. Third, it specifies a structured output format in JSON arrays, which enables direct integration with downstream planning modules. By following these principles, the decomposer reliably translates long-form user descriptions into a set of actionable sub-prompts $[\mathbf{c}_i]_{i=1}^{C}$, where the subscript i indexes each sub-task and the superscript C denotes the total number of decomposed sub-prompts, which can be independently processed in later stages.

As a result, the Prompt Decomposer provides the foundation for handling complex user instructions in an automated manner, ensuring scalability to diverse multi-task

scenarios without the need for manually predefined motion stages.

7.3 Sub-task Motion Planner

For the i-th sub-prompt $\mathbf{c}_i$ generated by the Prompt Decomposer, the LLM-driven Sub-task Motion Planner designed in this section produces a sparse 3D skeletal sequence as motion guidance $\mathbf{g}_i \in \mathbb{R}^{T\times J\times 3}$, where T is the number of frames, J is the number of body joints, and the last axis stores global coordinates (x, y, z). The LLM employed here is the same as that used in the Prompt Decomposer described in Section 7.2, ensuring consistent language understanding and reasoning throughout the entire pipeline.It will guide the Scene-Aligned Diffusion Model introduced in the next section to generate the corresponding motion $\mathbf{m}_i$. Specifically, given the sub-prompt $\mathbf{c}_i$ and the scene information represented by the road map $\mathbf{R}$ and the set of object height maps $\{\mathbf{H}^o\}_{o=1}^{O}$, the Sub-task Motion Planner first generates a structured plan $\mathcal{P}_i = (\mathcal{T}_i, \mathcal{K}_i, \mathcal{C}_i)$, where $\mathcal{T}_i$ sketches the root trajectory on the ground plane, $\mathcal{K}_i$ provides a compact set of keyframes containing pelvis positions and optionally selected joints, and $\mathcal{C}_i$ encodes interaction constraints such as the contact joint, the approach direction, and the target grid cell. Finally, the motion guidance $\mathbf{g}_i$ is reconstructed based on the generated plan $\mathcal{P}_i$.

The planner utilizes two types of maps at different reasoning processes. When reasoning over scene traversal, it relies on the road map $\mathbf{R} \in \{0, 1, 2\}^{H\times W}$, which serves as a global grid representation of the scene. Cells with value 0 denote traversable areas, 1 indicate obstacles, and 2 mark target regions. When reasoning about precise object–scene contacts, the planner first identifies the target object among all candidates and selects its corresponding upper-surface height map $\mathbf{H}^o \in \mathbb{R}^{h_o\times w_o}$. Each

object's 3D bounding box is projected onto the scene coordinate system, where x and y represent horizontal floor positions and z denotes vertical height. The resulting height map $\mathbf{H}^o$ is defined over the object's footprint, with each grid cell storing the elevation of the top surface along the z-axis. This dual-map design enables the planner to reason globally about navigation and locally about contact interactions within the same framework.

Depending on whether the motion requires fine-grained interaction inferred from the prompt semantics, the planner categorizes each sub-task into either a *navigation* or an *interaction* type (see Appendix B). For navigation-type motions, when generating the trajectory $\mathcal{T}_i$ within the plan $\mathcal{P}_i$, the planner determines an appropriate number of keyframes and samples an initial point in a traversable area (cell value = 0) on the road map $\mathbf{R}$, maintaining a reasonable distance from the target region. It then searches for an optimal trajectory that avoids obstacles (cell value = 1) and terminates at a feasible end point near the target area (cell value = 2). The corresponding keyframes $\mathcal{K}_i$ include motion phrases decomposed by the LLM (e.g., for the instruction "walk to the door," the motion is divided into "start walking," "midway walking," and "reaching the door"), the motion tendency (such as *toward* or *away*), and the 3D coordinates of the pelvis joint. The motion phrases and motion tendency mainly serve to help the LLM maintain semantic coherence and logical continuity during plan generation, resulting in smoother and more natural motion transitions. The final pelvis coordinates will later be used to construct the motion guidance $\mathbf{g}_i$. For interaction-type motions, the planner additionally generates interaction constraints $\mathcal{C}_i$, including the selected interaction joint, the approach direction, and the specific

contact region determined from the height map $\mathbf{H}^o$. The corresponding joint coordinates are also attached to $\mathcal{C}_i$, allowing precise alignment between the interacting body part and the target object surface.

Finally, by extracting the spatio-temporal coordinates of all joints contained in the plan $\mathcal{P}_i$ and padding the intermediate missing frames, a sparse motion guidance sequence $\mathbf{g}_i$ is obtained. It will condition the Scene-Aligned Diffusion Models at the level of trajectories, key poses, and contact goals, ensuring that the synthesized $\mathbf{m}_i$ is consistent with both the sub-prompt semantics and the scene constraints in Section 7.4.

7.4 Scene-Aligned Diffusion Model

Following the MDM-style denoising process [29] and the aligned generator adopted in TSTMotion [28], each motion instance $\mathbf{m}_i$ is generated through a diffusion sampling procedure under the scene-aware condition. Starting from isotropic Gaussian noise $\mathbf{m}_i^{(K)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, a transformer-based denoiser f_θ iteratively refines it into a clean motion sequence conditioned on the diffusion step k and the sub-prompt embedding $\mathbf{c}_i$.

At each diffusion step $k \in \{K, \ldots, 1\}$, the network predicts an estimate of the noise-free motion, denoted as

$$\widehat{\mathbf{m}}_i^{(0,k)} = f_\theta\big(\mathbf{m}_i^{(k)},\, k,\, \mathbf{c}_i\big), \tag{7.4.1}$$

where the superscript $(0, k)$ indicates that $\widehat{\mathbf{m}}_i^{(0,k)}$ is the denoised (i.e., step-0) motion estimated from the current noisy state at diffusion step k.

Given this estimate, the DDPM posterior mean [33] is computed as

$$\boldsymbol{\mu}_i^{(k)} = \frac{\sqrt{\bar{\alpha}_{k-1}}\,(1-\alpha_k)}{1-\bar{\alpha}_k}\,\widehat{\mathbf{m}}_i^{(0,k)} + \frac{\sqrt{\alpha_k}\,(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k}\,\mathbf{m}_i^{(k)}, \tag{7.4.2}$$

where $\alpha_k \in (0,1)$ is the diffusion coefficient and $\bar{\alpha}_k = \prod_{s=1}^{k} \alpha_s$ is the cumulative product controlling the noise schedule.

The next state is then sampled from this posterior as

$$\mathbf{m}_i^{(k-1)} = \boldsymbol{\mu}_i^{(k)} + \sigma_k\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{7.4.3}$$

where $\sigma_k^2 = 1 - \alpha_k$ denotes the variance term that determines the stochasticity at step k.

After K denoising steps, the final motion $\mathbf{m}_i^{(0)}$ (or simply $\mathbf{m}_i$) is obtained as the generated motion sequence corresponding to the i-th sub-prompt.

To align the reverse diffusion process with the scene-based plan, I incorporate the motion guidance $\mathbf{g}_i \in \mathbb{R}^{T\times J\times 3}$ generated by the planner in the previous subsection. At each diffusion step k, an alignment energy encourages the denoiser's clean prediction to conform to the scene-conditioned plan and the intended contacts:

$$\mathcal{E}_{\text{align}}^{(k)} = \sum_{(t,j)\in\Omega_i} \Big\| \mathbf{g}_i[t,j] - FK\big(\widehat{\mathbf{m}}_i^{(0,k)}\big)[t,j] \Big\|_2^2, \tag{7.4.4}$$

where Ω_i denotes the index set of frame–joint pairs constrained by the plan, t and j refer to the frame index ($t=1,\ldots,T$) and the joint index ($j=1,\ldots,J$), respectively, and $FK(\cdot)$ denotes the forward kinematics function that maps a motion representation (i.e., joint rotations) into the corresponding 3D skeleton sequence.

A small posterior update is then applied to refine the denoiser's prediction at diffusion step k:

$$\widehat{\mathbf{m}}_i^{(0,k)} \leftarrow \widehat{\mathbf{m}}_i^{(0,k)} - \lambda \nabla_{\widehat{\mathbf{m}}_i^{(0,k)}} \mathcal{E}_{\text{align}}^{(k)}, \tag{7.4.5}$$

where λ is a scalar hyperparameter controlling the guidance strength. This posterior refinement gently nudges the reverse diffusion toward motions consistent with the planner-generated scene-based plan $\mathbf{g}_i$, while preserving the stochastic nature of the diffusion process.

To further prevent interpenetration with the scene geometry, a soft non-collision constraint is incorporated during sampling. Let $\mathbf{P}$ denote the mesh of the given scene, and $\text{SMPL}\big(\widehat{\mathbf{m}}_i^{(0,k)}\big)$ represent the skinned human mesh reconstructed from the current clean motion prediction. Using a signed distance field $\text{SDF}(\cdot, \mathbf{P})$ that measures the signed distance between the body surface and the scene mesh, a penetration energy is defined as

$$\mathcal{E}_{\text{scene}}^{(k)} = \text{ReLU}\big(- \text{SDF}(\text{SMPL}(\widehat{\mathbf{m}}_i^{(0,k)}), \mathbf{P})\big), \tag{7.4.6}$$

where $\text{ReLU}(\cdot)$ excludes body points that are already outside the scene mesh, and $\text{SDF}(\cdot, \mathbf{P})$ is updated at each diffusion step to reflect the current body–scene configuration.

A small posterior update is then applied to the clean prediction at diffusion step k:

$$\widehat{\mathbf{m}}_i^{(0,k)} \leftarrow \widehat{\mathbf{m}}_i^{(0,k)} - \eta \nabla_{\widehat{\mathbf{m}}_i^{(0,k)}} \mathcal{E}_{\text{scene}}^{(k)}, \tag{7.4.7}$$

where η is a scalar hyperparameter controlling the repulsion strength. This update softly penalizes body–scene intersections, guiding the reverse diffusion toward physically plausible, non-colliding motions while preserving the stochasticity of sampling.

In practice, the inputs to the generator are the text embedding $\mathbf{c}_i$, the initial noise sample $\mathbf{m}_i^{(K)}$, and the planner-produced guidance $\mathbf{g}_i$; the output is the denoised motion $\mathbf{m}_i^{(0)}$. Both guidance terms act only at sampling time and therefore require no extra training, gradually steering the samples toward motions that satisfy the sub-prompt semantics and the scene constraints.

7.5 Experiments and Results

To assess the feasibility of my MTSA-T2M framework, this chapter focuses on qualitative evaluations. I generate multiple multi-task motion sequences in several scenes from the ScanNet [19] dataset. The selected prompts include varying numbers and orders of navigation and interaction tasks, as well as diverse object configurations. Among the evaluated scenes, one features a larger and more complex layout with multiple objects and interaction regions, while another represents a smaller and simpler environment. These results indicate that MTSA-T2M produces scene-aware, text-conditioned motions and shows strong generalization to varied and demanding settings.

7.5.1 Implementation Details

The qualitative experiments were implemented in PyTorch 1.7.1 and run on an Ubuntu 20.04 workstation with 32 GB RAM, an Intel(R) Core(TM) i9-13900 CPU @ 2.00 GHz, and an NVIDIA GeForce RTX 4080 GPU (16 GB). The LLM was GPT-4. The MDM component used its default settings, and following TSTMotion, the weighting in Eq. (7.4.5) was set to $\lambda = 2$, while the term in Eq. (7.4.7) was set to

$\eta = 0.5$.

7.5.2 Qualitative Result

Prompt	Result
A person walks to the door, then moves toward the chair near the shelf and climbs onto it.	
A person sits on the bed, then stands up and quickly runs toward the TV.	

Figure 7.2: Qualitative Result

Figure 7.2 visualizes MTSA-T2M outputs across three scenes and instructions. Overall, the framework reliably parses long prompts with multiple implicit tasks into sub-goals and generates scene–aligned motion segments for each, without manual specification of motion types or their sequence.

In the first example, the prompt represents a sequence involving multiple navigation actions followed by an interaction, within a relatively complex scene. To specify the target more precisely, the phrase "near the shelf" is used to locate the intended chair. The prompt is automatically decomposed into three sub-tasks: "Walk to the door," "Move to the chair near the shelf," and "Climb onto the chair near the shelf." The character successfully reaches the designated objects with accurate root trajectories, confirming the effectiveness of the navigation components in the Sub-task Motion Planner. In the final sub-task, the feet make stable contact with the chair surface,

indicating successful interaction control. Occasional interpenetration is observed in the middle of the sequence, suggesting that the current interaction constraints still have room for improvement.

For the second example, the sequence involves multiple interaction actions followed by a navigation task, taking place in a relatively simple scene that allows easier localization of the target objects. MTSA-T2M segments the prompt into three sub-tasks: "Sit on the bed," "Stand up from the bed," and "Run toward the TV." Although no explicit rule enforces spatial continuity between sub-tasks, the planner selects plausible contact points on the bed, resulting in visually coherent transitions between sitting and standing. In the final sub-task, the running motion demonstrates distinct dynamics compared to walking, indicating that the Scene-Aligned Diffusion Models can preserve the stylized behaviors inherited from the base MDM while remaining adaptive to the surrounding scene context.

Chapter 8

Conclusion and Future Work

8.1 Discussion

In this section, I will discuss several practical aspects that should be considered when applying the proposed methods to real-world scenarios. While this thesis primarily focuses on improving the efficiency and quality of end-to-end text-to-motion generation, it is also necessary to address potential challenges that arise in practical use, particularly cases where the generated motions might be uncomfortable to view or unsafe to experience.

Regarding the issue of motion comfort, the stochastic nature of motion generation may occasionally lead to unnatural postures or movements that deviate from common human motion habits. In addition, minor artifacts such as sliding, jittering, or jerking sometimes appear in generated sequences. In VR applications, these irregularities can induce motion sickness or discomfort, leading to unpleasant viewing. Therefore, ensuring motion comfort will be an essential direction for future improvement, as it is crucial for delivering a stable and reliable visual performance to users.

Regarding motion safety, on one hand, directly experiencing unverified generated motions may result in rotations that exceed the normal range of motion, posing risks if replicated by real humans. On the other hand, due to the complexity of the planning system and the inherent limitations in LLM reasoning, the model may occasionally produce unsafe plans that lead to penetration or excessive sliding, which could cause loss of balance or even physical hazards in real-world scenes. Therefore, ensuring the reliability and safety of generated motions must be treated as a critical priority for future development.

8.2 Conclusion

This thesis introduced two complementary frameworks for text-driven human motion generation. MDD treats text-to-motion as diffusion over compact, kinematics-aware codes: Kinematic Chain Quantization learns expressive discrete representations that capture both local joint dynamics and whole-body structure, and a Masked Deconstructed Diffusion Transformer performs parallel masked index refinement under textual conditioning; decoding the predicted codes yields motions that follow the description, remain temporally coherent, and can be produced efficiently.

Building on this core, MTSA-T2M addresses long, composite instructions in realistic environments: a prompt decomposer segments a complex description into sub-prompts, a sub-task motion planner reasons over the scene's road map and per-object height maps to produce actionable guidance, and scene-aligned diffusion modules synthesize motion segments that respect both language and geometry. Together, MDD improves semantic fidelity, diversity, and runtime through its discrete interface and parallel refinement, while MTSA-T2M delivers end-to-end multi-task generation with

consistent navigation and object interactions, modular components that can be upgraded independently, and robust generalization to varied scenes and instructions.

8.3 Future Work

For MDD, there are several limitations. Current public datasets typically provide motions at 20–30 FPS, which is suboptimal for synthesizing fast, high-amplitude movements with crisp timing. Moreover, when training across datasets with different kinematic hierarchies or skeleton conventions, the encoder layers in Kinematic Chain Quantization require manual customization. Going forward, I will pursue a temporal–kinematic agnostic pipeline: learning rate-invariant motion codes that can be decoded at arbitrary frame rates, and introducing an adaptive kinematic interface that maps diverse skeletal definitions into a shared canonical space with minimal hand-tuning. I also plan to explore mixed-resolution training and cross-skeleton alignment losses to improve robustness and transferability.

For MTSA-T2M, the main gaps lie in the coupling between sub-tasks and the breadth of evaluation. While the framework decomposes long prompts reliably, the transitions between segments rely on simple stitching and do not enforce stronger continuity constraints on velocity, contact persistence, or momentum. Quantitative assessment is also limited, focusing primarily on qualitative case studies. In future work, I will introduce an explicit transition module that predicts handoff states and timing between sub-tasks, along with soft constraints that preserve speed profiles, contact states, and heading consistency across boundaries. I will also develop standardized scene-aware benchmarks with automatic metrics for navigation success, contact accuracy, path efficiency, and collision rates, complemented by user studies to

assess perceived realism and instruction faithfulness.

For motion comfort and motion safety, future work can incorporate a post-check mechanism into the framework to ensure that generated motions comply with standard kinematic constraints and maintain natural, physically valid behavior. When scene awareness is involved, additional runtime validation can be performed to guarantee safe interaction with environmental geometry. Moreover, providing users with intermediate previews of generated motions can help them assess and adjust results early, preventing error accumulation and improving overall user control. Through these enhancements, the system can maintain both generation efficiency and realism while ensuring that the motions meet the comfort and safety requirements of real-world applications.

Appendix A

Prompt of Prompt Decomposer

You are an intelligent expert on multi-task motion caption analysis. I will provide you with a natural language caption describing a long and complex human motion sequence in a 3D environment. Your task is not to generate 3D motion directly, but to decompose the caption into multiple concise motion sub-captions, each representing one clear physical action.

Requirements: 1. Subtask granularity: - Each sub-caption must describe one atomic human action (e.g., “Walk to the door”, “Turn left”, “Sit on the sofa”). - If the caption includes temporal connectors such as “then”, “after that”, “and”, “subsequently”, “finally”, use them as natural split points.

2. Spatial awareness: - When generating each sub-caption, preserve object and location references from the original description (e.g., “Move to the chair near the shelf”, not just “Move to the chair”). - If multiple actions involve the same object, keep the object name consistent across the steps.

3. Output format: - Return the result as a JSON array of strings. - Each string starts with a capital letter and uses the base form of the verb (no -s or -ed).

4. Semantic clarity: - Do not merge distinct actions into one line. - Do not invent actions or objects not mentioned in the input caption.

Examples:

Example 1 Input: A woman runs to the kitchen, opens the refrigerator, and takes out a bottle of water.

Output: ["Run to the kitchen", "Open the refrigerator", "Take out the bottle of water"]

Example 2 Input: The man walks to the desk near the window, pulls the chair, and sits down on it.

Output: ["Walk to the desk near the window", "Pull the chair near the window", "Sit on the chair near the window"]

Example 3 Input: A person moves across the living room, picks up the cup on the table, and places it on the shelf beside the TV.

Output: ["Move across the living room", "Pick up the cup on the table", "Place the cup on the shelf beside the TV"]

Example 4 Input: The child crawls to the toy box, opens the lid, and puts a doll inside.

Output: ["Crawl to the toy box", "Open the lid of the toy box", "Put the doll inside the toy box"]

Appendix B

Prompts of Sub-task Motion Planner

B.1 Navigation

[12]

Standardized Output Schema (use null where not applicable): { “task”: “navigation” — “interaction”, “interaction_joint”: “pelvis” — “left_hand” — “right_hand” — “left_foot” — “right_foot”, “motion_tendency”: “toward” — “away” — null, “interaction_direction”: “left” — “right” — “top” — “bottom” — “left-top” — “left-bottom” — “right-top” — “right-bottom” — null, “start”: [x, y] — null, “end”: [x, y] — null, “interaction_grid”: [i, j] — null } (If any original example shows a different field set, adapt it to this unified schema while preserving semantics.)

[navigation_plan]

You are an intelligent expert on the interaction between humans and 3D objects. You will be provided with a caption of a 3D human motion and the corresponding

target information in the 3D scene. Your job is to generate 3D human motion in the format of skeleton joints for this person, which must be aligned with the given caption and target. Here are more details: 1. Coordinate System: The coordinate system of the 3D scene includes x, y and z-axis, and every 100 units means 1 meter length. The positive z-axis represents height. The person moves on the XOY plane and his "pelvis" should be around 85 units at z-axis when standing upright. 2. Target Object: The relevant information of the target is also presented. 3. Motion Tendency: You first need to determine the motion tendency, that is, whether the motion is "toward" or "away" from the target. If the motion is like "walk to", the motion tendency is "toward"; If the motion is like "walk away", the motion tendency is "away". 4. Motion Orientation: You must judge this person's motion orientation. If the motion is like "walk to" or "walk away": the motion orientation is set as "forward". If the motion is like "walk backwards": the motion orientation is set as "backward". 5. Motion Start and End: The start of the motion on XOY plane should be the "motion_start" of the target. The end of the motion on XOY plane should be the "motion_end" of the target. 6. Motion Keyframes: You must determine how many frames are in the motion according to the caption. This motion contains a minimum of 40 frames and a maximum of 100 frames, whose frame rate is 20 frames per second. Then you must analyze which frames are more key, and then generate the motion of "pelvis". Importantly, you must provide the start frame and the end frame of the motion. You must rationally plan the trajectory of this motion according to the above requirements and your analysis results. Before you start to generate a new motion, I will first offer an example:

Scene Scope: {x_min:0, x_max:630, y_min:0, y_max:500, z_min:0, z_max:250}

Target: { 'target': {'label': 'door', 'midpoint': [18, 379, 106], 'x_min': 10, 'x_max': 26, 'y_min': 331, 'y_max': 427, 'z_min': 14, 'z_max': 198}, 'motion_start': [15, 15], 'motion_end': [15, 335] }

Caption: walk to the door near the sofa.

Analysis: 1. Motion Tendency: Since the motion is "walk to", the motion tendency is "toward". 2. Motion Orientation: Since the motion is "walk to", the motion orientation is "forward". 3, Motion Start and End: Since the motion is "walk to", the joint "pelvis" should starts at [15,15,85] and ends at [15,335,85]. 4. Motion Keyframes: Since the start is away from the end, so we can consider it to take 80 frames. And we consider that the start, mid and the end frame are keyframes, so we provide the "pelvis" position in these frames.

Result: { "motion":"walk to the door", "motion_tendency":"toward", "motion_orientation":"forward",
"keyframe_1":{"state":"starting to walk","pelvis":[15,15,85],},
"keyframe_40":{"state":"midway walking","pelvis":[78,175,65],},
"keyframe_80":{"state":"reaching the door ","pelvis":[15,335,85],},
}

[navigation_target]

You are an intelligent expert on the interaction between humans and 3D objects. You will be provided with a caption of a human motion and the corresponding target object in the 3D scene. Your job is to reason the trajectory of the motion. Here are more details: 1. Road Map; The 3D scene will be projected onto the XOY plane and recorded in the form of a matrix. If the gird value of the road map is 0, the grid is walk-able for motion. If the gird value of the road map is 1, there are some

obstacles in this grid. If the gird value of the road map is 2, the target is inside this grid. 2. Interaction Joint: The joint used to interact with the target is “pelvis”. 3. Motion Tendency: You first need to determine the motion tendency, that is, whether the motion is “toward” or “away” from the target. If the motion is like “walk to”, the motion tendency is “toward”; If the motion is like “walk away”, the motion tendency is “away”; 4. Motion Trajectory: You need to first locate the target, and then reason trajectory the of the interaction joint “pelvis” on the road map. If the motion tendency is “toward”, then the end of motion should be at girds with value 2, the start of motion should be with value 0 and be close to the center of the scene; If the motion tendency is “away”, then the start of motion should be at girds with value 2, the end of motion should be with value 0 and away form the target’s grids. The distance between the start and the end should be moderate. Unless necessary, it is best not for this person to walk diagonally. There should be no obstacles between the start and end of the motion. Before you start to analyze the target, I will first offer an example:

Road Map of the Scene: [[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],]

(The illustration of this sample road map can be seen in Figure B.1.)

Caption: walk to the chair.

Analysis: 1. Interaction Joint: Since the motion is “walk to”, the interaction joint is “pelvis”. 2. Motion Tendency: Since the motion is “walk to”, the motion

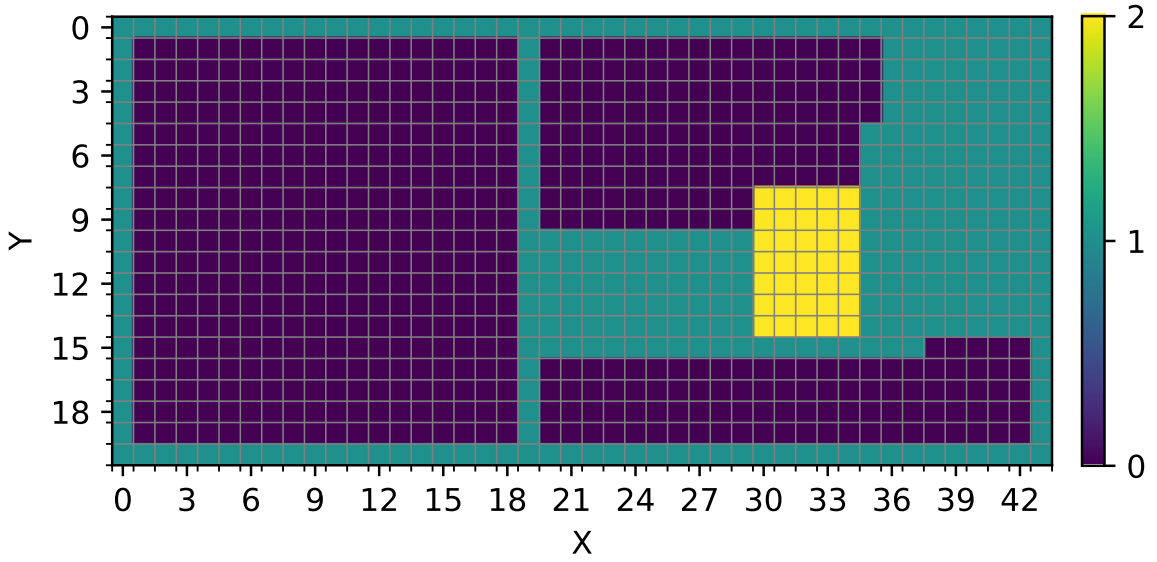

Figure B.1: Illustration of sample road map.

tendency is “toward”. 3. Motion Trajectory: It can be noted that the target locates from grid [30,8] to grid [34,14], and the motion tendency is “toward”. Therefore, the start can be grid [30,1], and end can be grid [30,8]. Note that there are no obstacles (grids with value 1) on the road.

Result: { “interaction_joint”:“pelvis”, “motion_start”: [30,1], “motion_end”: [30,8], }

B.2 Interaction

Standardized Output Schema (use null where not applicable): { “task”: “navigation” — “interaction”, “interaction_joint”: “pelvis” — “left_hand” — “right_hand” — “left_foot” — “right_foot”, “motion_tendency”: “toward” — “away” — null, “interaction_direction”: “left” — “right” — “top” — “bottom” — “left-top” — “left-bottom” — “right-top” — “right-bottom” — null, “start”: [x, y] — null, “end”: [x,

y] — null, “interaction_grid”: [i, j] — null } (If any original example shows a different field set, adapt it to this unified schema while preserving semantics.)

[interaction_scene]

You are an intelligent expert on the interaction between humans and 3D objects. You will be provided with a caption of a human motion and the corresponding target object in the 3D scene. Your job is to reason the feasible interaction direction of the motion. Here are more details: 1. Road Map; The 3D scene will be projected onto the XOY plane and recorded in the form of a matrix. If the gird value of the road map is 0, the grid is walk-able for motion. If the gird value of the road map is 1, there are some obstacles in this grid. If the gird value of the road map is 2, the target is inside this grid. 2. Feasible Interaction Direction: You can select the feasible interaction directions from “left”, “right”, “top”, “bottom”, “left-top” “left-bottom”, “right-top” and “right-bottom” of the target in the roadmap. First you should determine whether the target is at the boundary of the scene. If so, motion can only occur inside the scene. Secondly, you should determine whether there are many walk-able grids around the target. If so, the motion should occur in this direction. If you are unsure, you should provide as many interaction directions as possible. Before you start to analyze the target, I will first offer an example:

Road Map of the Scene: [[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0, 0, 1], [1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],]

(The illustration of this sample road map can be seen in Figure B.1.)

Caption: sit on the toilet.

Analysis: Since the target is not at the boundary of the scene; there are many walk-able areas at the “left” direction and few walk-able areas at “top” direction of the target; so the feasible interaction direction can be “left” and “left-top”.

Result: { “feasible_interaction_direction”:[“left”,“left-top”] }

[interaction_target]

You are an intelligent expert on the interaction between humans and 3D objects. You will be provided with a caption of a human motion and the corresponding target object in the 3D scene. Your job is to predict which part of the target, and from which direction of this target the motion should interact with. Here are more details: 1. Target Object: The target object will be projected onto the XOY plane and divided into multiple grids on the XOY plane. Each grid contains the height of the upper surface of the target in the grid, forming a height map. Each grid represents 100 square centimeters on the XOY plane. 2. Interaction Surface: For motion like “sit on the toilet”,“lie on bed”,“stand up from couch”,“stand on the table”, the interaction point should be the “top” surface of the target. 3. Interaction Joint: You should decide the decisive joint used to interact with the target, including “pelvis”, “left_hand”, “right_hand”, “left_foot” and “right_foot”. 4. Interaction Direction: You need to predict on which direction of the height map the person should interact with. Namely, the start and the end of the motion are much closer to the interaction direction than other directions. You can select the interaction directions from “left”, “right”, “top”, “bottom”, “left-top” “left-bottom”, “right-top” and “right-bottom” of the height map. You had better select one interaction direction from the provided feasible interaction direction. If the target’s height map grids are all concentrated at

one height, the interaction direction can be any direction. If the target's height map does not have all grids concentrated at one height, the interaction direction is from the higher area of the height map to the lower area. 5. Interaction Grids: Based on the interaction direction, you need to predict which grid of the height map the person should interact with the interaction joint. Therefore, the height change of interaction grids and the surrounding grids should be relatively smooth; the height of this grid should appear many times in the height map; these grids should be connected to the boundaries of the target. Importantly, the interaction grids should be close to the interaction direction. Namely, if there are grids of multiple heights that meet the requirements, you must not select the grids with the highest heights. For example, if some grids have a height near 40, and some have a height near 70. You should select the grid with a height near 40.

Before you start to analyze the target, I will offer two examples:

Example 1: Height Map of Target: [[70, 70, 65, 45, 45, 45, 45, 45, 40, 40, 40, 40, 40, 40, 35, 35, 40, 40, 40, 60, 65, 65, 60], [75, 75, 70, 50, 50, 50, 50, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 55, 60, 65, 65, 65], [0, 70, 70, 50, 50, 50, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 65, 65, 70, 70, 65], [0, 75, 70, 60, 50, 55, 55, 50, 45, 45, 45, 45, 45, 45, 45, 45, 45, 50, 65, 75, 75, 70, 65], [75, 75, 80, 80, 70, 60, 55, 60, 60, 60, 60, 60, 45, 50, 55, 50, 45, 45, 65, 70, 70, 70, 70], [75, 90, 90, 90, 90, 90, 85, 70, 75, 80, 80, 85, 90, 70, 75, 65, 70, 70, 75, 75, 70, 70, 70], [80, 90, 90, 90, 90, 90, 90, 85, 85, 85, 90, 90, 90, 85, 85, 90, 90, 90, 90, 85, 85, 70, 70], [80, 80, 80, 80, 85, 90, 90, 85, 85, 85, 85, 85, 85, 85, 85, 90, 90, 90, 90, 90, 85, 70, 70], [80, 80, 80, 75, 75, 70, 85, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 65, 65, 60, 65, 65, 0],]

(The illustration of this sample height map can be seen in Figure B.2.)

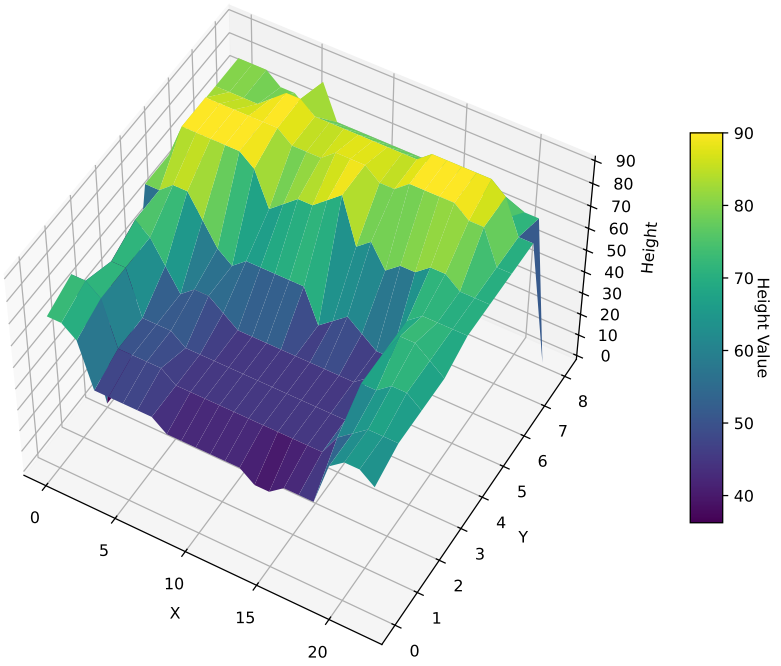

Figure B.2: Illustration of sample height map of couch.

Feasible Interaction Direction: [“top”]

Caption: sit on the couch.

Analysis: 1. Interaction Joint: Since the motion is “sit on the couch”, the interaction joint is “pelvis”. 2. Interaction Direction: The provided feasible interaction direction is [“top”]. THe height map of the target couch is mainly concentrated around 40 or 80, so the interaction direction is from 80 to 40 (i.e., “top”). Comprehensive analysis shows that the interaction direction should be “top”. 3. Interaction Grids: Since the motion is “sit on the couch”, the interaction surface is “top”. Since the height map of the couch and the motion of sitting, the grid [1,11] in the height of 45 and nearby grids are suitable interaction grids. This is because the grids at heights of 45 appear multiple times, are connected to each other, have a smooth

height change, and can be directly connected to the boundaries of the target. Importantly, the grids at heights near 90 also meet these requirements, but they are with the highest height. So we not select the grids at the height near 90.

Result: { "interaction_joint":"pelvis", "interaction_grid":[1,11], "interaction_direction":"top", }

Example 2: Height Map of Target: [[40, 40, 45, 45, 45, 45, 45, 50, 50, 55, 55, 55, 55, 55, 50, 50, 45, 45, 45, 45, 45, 40, 40], [45, 45, 50, 50, 50, 50, 50, 55, 55, 55, 55, 55, 55, 55, 55, 55, 50, 50, 50, 50, 50, 45, 45], [45, 45, 50, 50, 50, 50, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 50, 50, 50, 50, 45, 45], [45, 45, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 45, 45], [45, 45, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 45, 45], [45, 45, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 45, 45], [45, 45, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 45, 45], [45, 45, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 45, 45], [40, 40, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 40, 40]]

(The illustration of this sample height map can be seen in Figure B.3.)

Feasible Interaction Direction: ["bottom", "bottom-right"]

Caption: lie on the bed.

Analysis: 1. Interaction Joint: Since the motion is "lie on the bed", the interaction joint is "pelvis". 2. Interaction Direction: The provided feasible interaction direction is ["bottom", "bottom-right"]. The height map of the bed is mainly concentrated around 50, so the interaction direction can be the long side of the target (e.g., "bottom" and "top"). Comprehensive analysis shows that the interaction direction should be "top". 3. Interaction Grids: Since the motion is "lie on the bed",

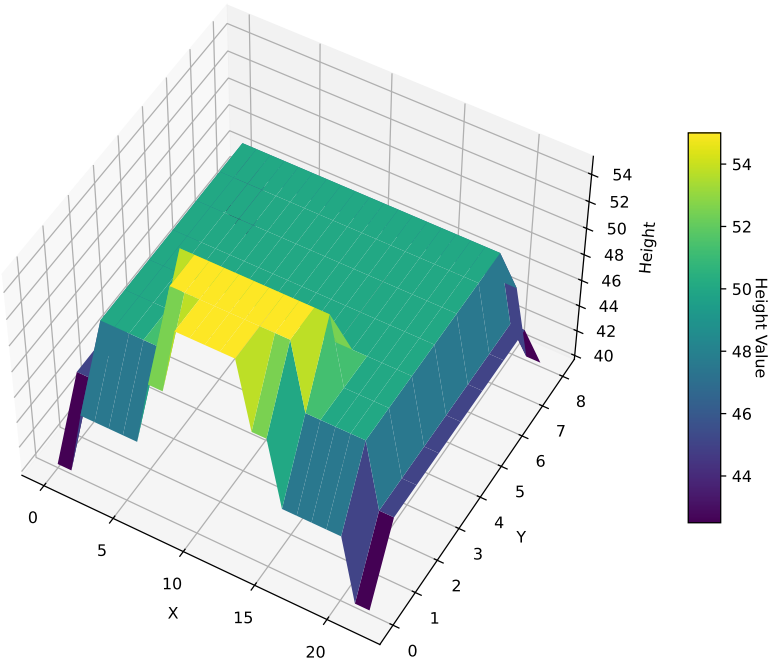

Figure B.3: Illustration of sample height map of bed.

the interaction surface is "top". Since the height map of the bed and the motion of lying, the grid [7,12] in the height of 55 and nearby grids are suitable interaction grids. This is because the grids are at the boundaries of the target.

Result: { "interaction_joint":"pelvis", "interaction_grid":[8,12], "interaction_direction":"bottom", }

[interaction_plan]

You are an intelligent expert on the interaction between humans and 3D objects. You will be provided with a caption of a 3D human motion and the corresponding target information in the 3D scene. Your job is to generate 3D human motion in the format of skeleton joints for this person, which must be aligned with the given caption and target. Here are more details: 1. Coordinate System: The coordinate system of the 3D scene includes x, y and z-axis, and every 100 units means 1 meter length.

The positive z-axis represents height. The person moves on the XOY plane and his "pelvis" should be around 85 units at z-axis when standing upright. 2. Target Object: The relevant information of the target is also presented, including interaction joint, interaction position and interaction direction. 3. Motion Tendency, Interaction Joint and Interaction Position: If the motion is like "sit" or "lie", the motion tendency is "toward" the target. Then the interaction joint should reach the interaction position at end. If the motion is like "stand", the motion tendency is "away" the target. Then the interaction joint should reach the interaction position at start. 4. Interaction Direction: If the given interaction direction includes "x_min" and motion tendency is "toward", the position of start on the x-axis should be smaller than the target's "x_min". If the given interaction direction includes "x_min" and motion tendency is "away", the position of end on the x-axis should be smaller than the target's "x_min". If the given interaction direction includes "x_max" and motion tendency is "toward", the position of start on the x-axis should be larger than the target's "x_max". If the given interaction direction includes "x_max" and motion tendency is "away", the position of end on the x-axis should be larger than the target's "x_max". If the given interaction direction includes "y_min" and motion tendency is "toward", the position of start on the y-axis should be smaller than the target's "y_min". If the given interaction direction includes "y_min" and motion tendency is "away", the position of end on the y-axis should be smaller than the target's "y_min". If the given interaction direction includes "y_max" and motion tendency is "toward", the position of start on the y-axis should be larger than the target's "y_max". If the given interaction direction includes "y_max" and motion tendency is "away", the position of end on the y-axis should be larger than the target's "y_max". 5. Motion

Orientation: You must judge this person's motion orientation. If the motion is like "stand": the motion orientation is set as "forward". If the motion is like "sit" or "lie": the motion orientation is set as "backward". 6. Motion Keyframes: You must determine how many frames are in the motion according to the caption. This motion contains a minimum of 40 frames and a maximum of 100 frames, whose frame rate is 20 frames per second. Then you must analyze which frames are more key, and then generate the motion of "pelvis" and the interaction joint in these keyframes. Importantly, you must provide the start frame and the end frame of the motion. 7. Motion Trajectory: You must rationally plan the trajectory of this motion according to the above requirements and your analysis results. The distance from the start to the end is moderate. Before you start to generate a new motion, I will offer two examples:

Example 1: Target: { "target": {"label":"toilet","midpoint": [98, 77, 35],"x_min": 78,"x_max": 117,"y_min": 47,"y_max": 107,"z_min": 0,"z_max": 70}, "interaction_joint":"pelvis", "interaction_position":[78,77,45], "interaction_direction":"y_min", }

Caption: stand up from the toilet away from the curtain.

Analysis: 1. Motion Tendency, Interaction Joint and Interaction Position: Since the motion is "stand up from", the motion tendency is "away", the interaction joint "pelvis" should reach the interaction position [78,77,45] at start. 2. Interaction direction: Since the given interaction direction includes "y_min" and the motion tendency is "away", the position of end on the y-axis should be smaller than the target's "y_min". 3. Motion Orientation: Since the motion is "stand", the motion orientation is "forward". 4. Motion Keyframes: Since the motion is simple, so we

can consider it to take 50 frames. And we consider that the start, mid and the end frame are keyframes, so we provide the "pelvis" position in these frames. 5. Motion Trajectory: The interaction joint "pelvis" should be interaction position [78,77,45] at start; the end should be at some distance head of the start along the interaction direction and out of the target bounding box, which can be [38,77,85];

Result: { "motion":"stand up from the toilet", "motion_tendency":"away", "motion_orientation":"forward",
"keyframe_1":{"state":"starting to stand up","pelvis":[78,77,45],},
"keyframe_25":{"state":"midway standing up","pelvis":[78,57,65],},
"keyframe_50":{"state":"completing the standing ","pelvis":[78,37,85],},
}

Example 2: Target: { "target": {"label":"bed","midpoint": [96, 239, 36],"x_min": 45,"x_max": 147,"y_min": 95,"y_max": 383,"z_min": 0,"z_max": 72}, "interaction_joint":"pelvis", "interaction_position":[140,220,45], "interaction_direction":"x_max",
}

Caption: lie on the bed close to the door.

Analysis: 1. Motion Tendency, Interaction Joint and Interaction Position: Since the motion is "lie on the bed", the motion tendency is "toward", the interaction joint "pelvis" should reach the interaction position [140,220,45] at end. 2. Interaction direction: Since the given interaction direction includes "x_max" and the motion tendency is "toward", the position of start on the x-axis should be bigger than the target's "x_max". 3. Motion Orientation: Since the motion is "lie", the motion orientation is "backward". 4. Motion Keyframes: Since the motion is simple, so we can consider it to take 60 frames. And we consider that the start, mid and the end

frame are keyframes, so we provide the "pelvis" position in these frames. 5. Motion Trajectory: The interaction joint "pelvis" should be interaction position [140,220,45] at end; the start should be at some distance head of the end along the interaction direction and out of the target bounding box, which can be [155,220,85];

Result: { "motion":"lie on the bed", "motion_tendency":"toward", "motion_orientation":"backward",

"keyframe_1":{"state":"starting to lie down","pelvis":[155,220,85],},

"keyframe_25":{"state":"midway lying down","pelvis":[145,220,65],},

"keyframe_50":{"state":"completing the lying","pelvis":[140,220,45],},

}

Bibliography

[1] K. Aberman, Y. Weng, D. Lischinski, D. Cohen-Or, and B. Chen. Unpaired motion style transfer from video to animation. ACM Transactions on Graphics (TOG), 39(4):64–1, 2020.

[2] O. Alemi, J. Françoise, and P. Pasquier. Groovenet: Real-time music-driven dance movement generation using artificial neural networks. networks, 8(17): 26, 2017.

[3] S. Alexanderson, G. E. Henter, T. Kucherenko, and J. Beskow. Style-controllable speech-driven gesture synthesis using normalising flows. In Computer Graphics Forum, volume 39, pages 487–496. Wiley Online Library, 2020.

[4] S. Alexanderson, R. Nagy, J. Beskow, and G. E. Henter. Listen, denoise, action! audio-driven motion synthesis with diffusion models. ACM Transactions on Graphics (TOG), 42(4):1–20, 2023.

[5] T. Ao, Q. Gao, Y. Lou, B. Chen, and L. Liu. Rhythmic gesticulator: Rhythm-aware co-speech gesture synthesis with hierarchical neural embeddings. ACM Transactions on Graphics (TOG), 41(6):1–19, 2022.

[6] T. Ao, Z. Zhang, and L. Liu. Gesturediffuclip: Gesture diffusion model with clip latents. ACM Transactions on Graphics (TOG), 42(4):1–18, 2023.

[7] A. Aristidou, A. Yiannakidis, K. Aberman, D. Cohen-Or, A. Shamir, and Y. Chrysanthou. Rhythm is a dancer: Music-driven motion synthesis with global structure. IEEE transactions on visualization and computer graphics, 2022.

[8] U. Bhattacharya, N. Rewkowski, A. Banerjee, P. Guhan, A. Bera, and D. Manocha. Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents. In 2021 IEEE virtual reality and 3D user interfaces (VR), pages 1–10. IEEE, 2021.

[9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[10] Z. Cen, H. Pi, S. Peng, Z. Shen, M. Yang, S. Zhu, H. Bao, and X. Zhou. Generating human motion in 3d scenes from text descriptions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

[11] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman. Maskgit: Masked

generative image transformer. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 11305–11315, 2022. doi: 10.1109/CVPR52688.2022.01103.

[12] J. Chen, F. Liu, and Y. Wang. MDD: Masked Deconstructed Diffusion for 3D Human Motion Generation from Text . In 2025 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR), pages 61–72, 2025. Project page: `https://krishlo-chen.github.io/MDD/`.

[13] X. Chen, B. Jiang, W. Liu, Z. Huang, B. Fu, T. Chen, and G. 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.

[14] X. Chen, Z. Liu, S. Xie, and K. He. Deconstructing denoising diffusion models for self-supervised learning. arXiv preprint arXiv:2401.14404, 2024.

[15] Y. Cheng. and Y. Wang. Transformer-based two-level approach for music-driven dance choreography. In Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - GRAPP, pages 127–139. INSTICC, 2024.

[16] Y. Cheng, Y. Jiang, and Y. Wang. Music-stylized hierarchical dance synthesis with user control. In Computer Graphics International, 2024.

[17] L. Crnkovic-Friis and L. Crnkovic-Friis. Generative choreography using deep learning. arXiv preprint arXiv:1605.06921, 2016.

[18] R. Dabral, M. H. Mughal, V. Golyanik, and C. 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.

[19] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[20] Y. Ferstl and R. McDonnell. Iva: Investigating the use of recurrent motion modelling for speech gesture generation. In IVA '18 Proceedings of the 18th International Conference on Intelligent Virtual Agents, Nov 2018. URL `https://trinityspeechgesture.scss.tcd.ie`.

[21] Y. Ferstl, M. Neff, and R. McDonnell. Adversarial gesture generation with realistic gesture phasing. Computers & Graphics, 89:117–130, 2020.

[22] B. Foundation. Blender, 2024.

[23] S. Ghorbani, Y. Ferstl, D. Holden, N. F. Troje, and M.-A. Carbonneau. Zeroeggs: Zero-shot example-based gesture generation from speech. In Computer Graphics Forum, volume 42, pages 206–216. Wiley Online Library, 2023.

[24] C. Guo, S. Zou, X. Zuo, S. Wang, W. Ji, X. Li, and L. 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.

[25] C. Guo, S. Zou, X. Zuo, S. Wang, W. Ji, X. Li, and L. 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.

[26] C. Guo, X. Zuo, S. Wang, and L. 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.

[27] C. Guo, Y. Mu, M. G. Javed, S. Wang, and L. Cheng. Momask: Generative masked modeling of 3d human motions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1900–1910, June 2024.

[28] Z. Guo, H. Qu, H. Rahmani, D. Soh, P. Hu, Q. Ke, and J. Liu. Tstmotion: Training-free scene-aware text-to-motion generation, 2025. URL https://arxiv.org/abs/2505.01182.

[29] T. Guy, R. Sigal, G. Brian, S. Yoni, C.-o. Daniel, and B. Amit Haim. Human motion diffusion model. In The Eleventh International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=SJ1kSyO2jwu.

[30] I. Habibie, W. Xu, D. Mehta, L. Liu, H.-P. Seidel, G. Pons-Moll, M. Elgharib, and C. Theobalt. Learning speech-driven 3d conversational gestures from video. In Proceedings of the 21st ACM International Conference on Intelligent Virtual Agents, pages 101–108, 2021.

[31] I. Habibie, M. Elgharib, K. Sarkar, A. Abdullah, S. Nyatsanga, M. Neff, and C. Theobalt. A motion matching-based framework for controllable gesture synthesis from speech. In ACM SIGGRAPH 2022 Conference Proceedings, pages 1–9, 2022.

[32] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[33] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020.

[34] D. Holden, J. Saito, T. Komura, and T. Joyce. Learning motion manifolds with convolutional autoencoders. In SIGGRAPH Asia 2015 technical briefs, pages 1–4. 2015.

[35] D. Holden, J. Saito, and T. Komura. A deep learning framework for character motion synthesis and editing. ACM Transactions on Graphics (TOG), 35(4): 1–11, 2016.

[36] D. Holden, T. Komura, and J. Saito. Phase-functioned neural networks for character control. ACM Transactions on Graphics (TOG), 36(4):1–13, 2017.

[37] R. Huang, H. Hu, W. Wu, K. Sawada, M. Zhang, and D. Jiang. Dance revolution: Long-term dance generation with music via curriculum learning. arXiv preprint arXiv:2006.06119, 2020.

[38] Z. Huang et al. Scenediffuser: Diffusion-based generation, optimization, and

planning in 3d scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

[39] N. Jiang, Z. Zhang, H. Li, X. Ma, Z. Wang, Y. Chen, T. Liu, Y. Zhu, and S. Huang. Scaling up dynamic human-scene interaction modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

[40] K. Karunratanakul, K. Preechakul, S. Suwajanakorn, and S. Tang. Guided motion diffusion for controllable human motion synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2151–2162, 2023.

[41] H. Kong, K. Gong, D. Lian, M. B. Mi, and X. Wang. Priority-centric human motion generation in discrete latent space. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 14806–14816, 2023.

[42] T. Kucherenko, D. Hasegawa, G. E. Henter, N. Kaneko, and H. Kjellström. Analyzing input and output representations for speech-driven gesture generation. In Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents, pages 97–104, 2019.

[43] H.-Y. Lee, X. Yang, M.-Y. Liu, T.-C. Wang, Y.-D. Lu, M.-H. Yang, and J. Kautz. Dancing to music. Advances in neural information processing systems, 32, 2019.

[44] S. Levine, C. Theobalt, and V. Koltun. Real-time prosody-driven synthesis of body language. In ACM SIGGRAPH Asia 2009 papers, pages 1–10. 2009.

[45] S. Levine, P. Krähenbühl, S. Thrun, and V. Koltun. Gesture controllers. In ACM SIGGRAPH 2010 papers, pages 1–11. 2010.

[46] B. Li, Y. Zhao, S. Zhelun, and L. Sheng. Danceformer: Music conditioned 3d dance generation with parametric motion transformer. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 1272–1279, 2022.

[47] J. Li, Y. Yin, H. Chu, Y. Zhou, T. Wang, S. Fidler, and H. Li. Learning to generate diverse dance motions with transformer. arXiv preprint arXiv:2008.08171, 2020.

[48] J. Li, D. Kang, W. Pei, X. Zhe, Y. Zhang, Z. He, and L. Bao. Audio2gestures: Generating diverse gestures from speech audio with conditional variational autoencoders. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 11293–11302, 2021.

[49] L. Li and A. Dai. Genzi: Zero-shot 3d human-scene interaction generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

[50] P. Li, K. Aberman, Z. Zhang, R. Hanocka, and O. Sorkine-Hornung. Ganimator: Neural motion synthesis from a single sequence. ACM Transactions on Graphics (TOG), 41(4):1–12, 2022.

[51] R. Li, S. Yang, D. A. Ross, and A. Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 13401–13412, 2021.

[52] T. Li, H. Chang, S. Mishra, H. Zhang, D. Katabi, and D. Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2142–2152, 2023.

[53] Z. Li, Y. Zhou, S. Xiao, C. He, Z. Huang, and H. Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. arXiv preprint arXiv:1707.05363, 2017.

[54] H. Liu, Z. Zhu, N. Iwamoto, Y. Peng, Z. Li, Y. Zhou, E. Bozkurt, and B. Zheng. Beat: A large-scale semantic and emotional multi-modal dataset for conversational gestures synthesis. In European conference on computer vision, pages 612–630. Springer, 2022.

[55] H. Liu, Z. Zhu, G. Becherini, Y. Peng, M. Su, Y. Zhou, N. Iwamoto, B. Zheng, and M. J. Black. Emage: Towards unified holistic co-speech gesture generation via masked audio gesture modeling. arXiv preprint arXiv:2401.00374, 2023.

[56] L. Liu and J. Hodgins. Learning to schedule control fragments for physics-based characters using deep q-learning. ACM Transactions on Graphics (TOG), 36 (3):29, 2017.

[57] L. Liu and J. Hodgins. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. ACM Trans. Graph., 37(4):142:1–142:14, July 2018.

[58] X. Liu, Q. Wu, H. Zhou, Y. Du, W. Wu, D. Lin, and Z. Liu. Audio-driven

co-speech gesture video generation. Advances in Neural Information Processing Systems, 35:21386–21399, 2022.

[59] Y. Lou, L. Zhu, Y. Wang, X. Wang, and Y. Yang. Diversemotion: Towards diverse human motion generation via discrete diffusion. arXiv preprint arXiv:2309.01372, 2023.

[60] J. Martinez, H. H. Hoos, and J. J. Little. Stacked quantizers for compositional vector compression. arXiv preprint arXiv:1411.2173, 2014.

[61] J. Martinez, M. J. Black, and J. Romero. On human motion prediction using recurrent neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2891–2900, 2017.

[62] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In International conference on machine learning, pages 8162–8171. PMLR, 2021.

[63] OpenAI. Gpt-4o system card. `https://openai.com/index/gpt-4o-system-card/`, 2024. Accessed: 2025-10-09.

[64] L. Pan et al. Synthesizing physically plausible human motions in 3d scenes. In International Conference on 3D Vision (3DV), 2024.

[65] M. Papillon, M. Pettee, and N. Miolane. Pirounet: Creating dance through artist-centric deep learning. In International Conference on ArtsIT, Interactivity and Game Creation, pages 447–465. Springer, 2022.

[66] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. ACM Transactions On Graphics (TOG), 37(4):1–14, 2018.

[67] X. B. Peng, A. Kanazawa, J. Malik, P. Abbeel, and S. Levine. Sfv: Reinforcement learning of physical skills from videos. ACM Transactions On Graphics (TOG), 37(6):1–14, 2018.

[68] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. ACM Transactions on Graphics (ToG), 40(4):1–20, 2021.

[69] M. Petrovich, M. J. Black, and G. Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In International Conference on Computer Vision (ICCV), 2021.

[70] M. Petrovich, M. J. Black, and G. Varol. Temos: Generating diverse human motions from textual descriptions. In European Conference on Computer Vision, pages 480–497. Springer, 2022.

[71] M. Plappert, C. Mandery, and T. Asfour. The kit motion-language dataset. Big data, 4(4):236–252, 2016.

[72] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In International conference on machine learning, pages 8748–8763. PMLR, 2021.

[73] Z. Ren, Z. Pan, X. Zhou, and L. Kang. Diffusion motion: Generate text-guided 3d human motion by diffusion model. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[74] M. Shi, K. Aberman, A. Aristidou, T. Komura, D. Lischinski, D. Cohen-Or, and B. Chen. Motionet: 3d human motion reconstruction from monocular video with skeleton consistency. ACM Trans. Graph., 40(1), Sept. 2020. ISSN 0730-0301. doi: 10.1145/3407659. URL https://doi.org/10.1145/3407659.

[75] L. Siyao, W. Yu, T. Gu, C. Lin, Q. Wang, C. Qian, C. C. Loy, and Z. 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.

[76] H. J. Smith, C. Cao, M. Neff, and Y. Wang. Efficient neural networks for real-time motion style transfer. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 2(2):1–17, 2019.

[77] S. Starke, H. Zhang, T. Komura, and J. Saito. Neural state machine for character-scene interactions. In ACM SIGGRAPH Asia, 2019.

[78] T. Tang, J. Jia, and H. Mao. Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis. In Proceedings of the 26th ACM international conference on Multimedia, pages 1598–1606, 2018.

[79] G. W. Taylor and G. E. Hinton. Factored conditional restricted boltzmann

machines for modeling motion style. In Proceedings of the 26th annual international conference on machine learning, pages 1025–1032, 2009.

[80] U. Technologies. Unity 3d, 2024.

[81] G. Tevet, B. Gordon, A. Hertz, A. H. Bermano, and D. Cohen-Or. Motionclip: Exposing human motion generation to clip space. In European Conference on Computer Vision, pages 358–374. Springer, 2022.

[82] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. Advances in neural information processing systems, 30, 2017.

[83] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

[84] H. Wang, E. S. Ho, H. P. Shum, and Z. Zhu. Spatio-temporal manifold learning for human motions via long-horizon modeling. IEEE transactions on visualization and computer graphics, 27(1):216–227, 2019.

[85] Y. Wang and M. Neff. Deep signatures for indexing and retrieval in large motion databases. In Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, pages 37–45, 2015.

[86] Z. Wang, J. Chai, and S. Xia. Combining recurrent neural networks and adversarial training for human motion synthesis and control. IEEE transactions on visualization and computer graphics, 27(1):14–28, 2019.

[87] Z. Wang, Y. Chen, B. Jia, P. Li, J. Zhang, J. Zhang, T. Liu, Y. Zhu, W. Liang, and S. Huang. Move as you say, interact as you can: Language-guided human motion generation with scene affordance. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

[88] Z. Wang et al. Humanise: Language-conditioned human motion generation in 3d scenes. In Advances in Neural Information Processing Systems (NeurIPS), 2022.

[89] S. Xia, C. Wang, J. Chai, and J. Hodgins. Realtime style transfer for unlabeled heterogeneous human motion. ACM Transactions on Graphics (TOG), 34(4): 119, 2015.

[90] Z. Xiao, T. Wang, J. Wang, J. Cao, W. Zhang, B. Dai, D. Lin, and J. Pang. Unified human-scene interaction via prompted chain-of-contacts. In International Conference on Learning Representations (ICLR), 2024.

[91] S. Yan, Z. Li, Y. Xiong, H. Yan, and D. Lin. Convolutional sequence generation for skeleton-based action synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 4394–4402, 2019.

[92] H. Yi, H. Liang, Y. Liu, Q. Cao, Y. Wen, T. Bolkart, D. Tao, and M. J. Black. Generating holistic 3d human motion from speech. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 469–480, 2023.

[93] H. Yi, J. Thies, M. J. Black, X. B. Peng, and D. Rempe. Generating human interaction motions in scenes with text control. In ECCV, 2024.

[94] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu. Vector-quantized image modeling with improved vqgan. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=pfNyExj7z2.

[95] M. E. Yumer and N. J. Mitra. Spectral style transfer for human motion between independent actions. ACM Transactions on Graphics (TOG), 35(4):137, 2016.

[96] B. Zalán, M. Raphaël, V. Damien, K. Eugene, P. Olivier, S. Matt, T. Olivier, T. Marco, and Z. Neil. Audiolm: A language modeling approach to audio generation, 2022.

[97] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi. Soundstream: An end-to-end neural audio codec. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 30:495–507, 2021.

[98] J. Zhang, Y. Zhang, X. Cun, Y. Zhang, H. Zhao, H. Lu, X. Shen, and S. Ying. Generating human motion from textual descriptions with discrete representations. In 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 14730–14740, 2023. doi: 10.1109/CVPR52729.2023.01415.

[99] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. arXiv preprint arXiv:2208.15001, 2022.

[100] M. Zhang, X. Guo, L. Pan, Z. Cai, F. Hong, H. Li, L. Yang, and Z. Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In 2023 IEEE/CVF

International Conference on Computer Vision (ICCV), pages 364–373, 2023. doi: 10.1109/ICCV51070.2023.00040.

[101] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

[102] S. Zhang, M. J. Black, and S. Tang. Place: Proximity learning of articulation and contact in 3d environments. In International Conference on 3D Vision (3DV), 2020.

[103] K. Zhao, S. Wang, Y. Zhang, T. Beeler, and S. Tang. Compositional human-scene interaction synthesis with semantic control. In European Conference on Computer Vision (ECCV), 2022.

[104] W. Zhuang, C. Wang, S. Xia, J. Chai, and Y. Wang. Music2dance: Music-driven dance generation using wavenet. arXiv preprint arXiv:2002.03761, 3(4): 6, 2020.