SPORT: From Zero-shot Prompts to Real-time Motion Generation

Bin Ji, Ye Pan*, Zhimeng Liu, Shuai Tan, and Xiaokang Yang

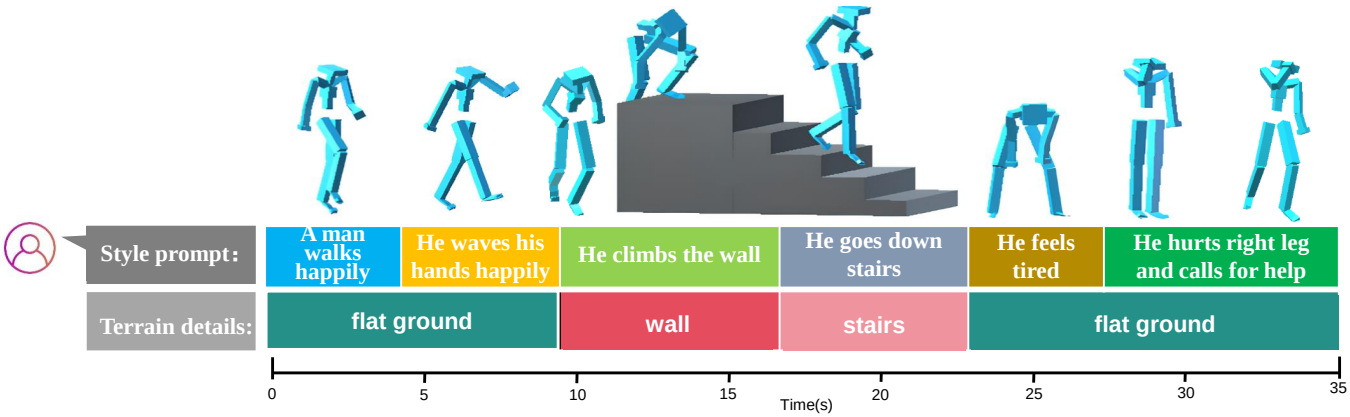


Fig. 1: Our proposed model can realize real-time zero-shot motion generation over various terrains by using composite and ever-changing text prompts. Further elaboration can be found in the supplementary video.

***Abstract*—Real-time motion generation has garnered significant attention within the fields of computer animation and gaming. Existing methods typically realize motion control via isolated style or content labels, resulting in short, simply motion clips. In this paper, we propose a motion generation framework, called SPORT ("from zero-Shot Prompt tO Real-Time motion generation"), for generating real-time and ever-changing motions using zero-shot prompts. SPORT consists of three primary components: (1) a body-part phase autoencoder that ensures smooth transitions between diverse motions; (2) a body-part content encoder that mitigates semantic gap between texts and motions; (3) a diffusion-based decoder that accelerates the denoising process while enhancing the diversity and realism of motions. Moreover, we develop a prototype for real-time application in Unity, demonstrating that our approach effectively considering the semantic gap caused by abstract style texts and rapidly changing terrains. Through qualitative and quantitative comparisons, we show that SPORT outperforms other approaches in terms of motion quality, style diversity and inference speed.**

***Index Terms*—Character animation, motion generation, diffusion models, contrastive learning, GPT-3.**

I. Introduction

CREATING real-time VR/AR human-like avatars is a hot research topic and has many applications in the field of computer animation and games. Recently, deep neural networks have become an attractive option to tackle this challenge. Utilizing the Mixture of Expert (MoE) architecture, several works, such as PFNN [1], MANN [2] and MVAE [3], have achieved notable success in learning motion periodicity, a crucial aspect for online generation of high-quality motions. However, these methods fall short in providing adequate control over motion content and style.

To incorporate style control into real-time motion synthesis, some methods [4], [5] retain phase networks to capture fundamental locomotion characteristics and further accomplish motion stylization on a frame-by-frame basis by employing residual connections. Nonetheless, it's worth noting that the complexity of the residual module is closely related to the quantity of style types, and it may struggle when dealing with an excessive number of style variations. Later, Mason et al. [6] propose a framework based on local phase network (LPN) [7], aiming to model 100 distinct performative styles using a feature-wise linear modulation (FiLM). This kind of example-based systems generate stylized motions by imitating provided motion clips. While effective for the styles included in the training dataset, these approaches lack a precise definition of motion content and style, and fail to facilitate the learning of disentangled style features. Consequently, these limitations hamper their ability to generalize to unseen styles.

Recently, numerous studies explore the zero-shot generation of motions from textual prompts using diffusion-based methods [8]–[11]. Their generation processes are conditioned on the semantic embeddings extracted from Contrastive-Language-Image-Pretraining (CLIP) model [12] or Large Language Models (LLM) [13]. In contrast to the aforementioned approaches, which directly utilize text embeddings from CLIP or LLMs, alternative methods such as GestureDiffuCLIP [14], TMR [15], and HumanTOMATO [16] pre-train a text encoder and a motion encoder in a contrastive manner [12]. They explicitly learn

B. Ji, Y. Pan, Z. Liu, S. Tan and X. Yang are with JHC & AI Institute, Shanghai Jiao Tong University, Shanghai, China. E-mail:{bin.ji, whitneypanye, 799734456, tanshuai0219, xkyang}@sjtu.edu.cn. Corresponding author: Ye Pan

text-motion aligned priors, resulting in a more motion-aware text embedding for driving motion generation. Despite their advancements, these methods face challenges in three aspects: (1) whole-body text-motion alignment remains difficult due to huge semantic gap; (2) the seq2seq architecture struggles to ensure seamless transition between different prompts without a large-scale cross-modal dataset; (3) the inference of diffusion model is computationally expensive, requiring iterations over thousands of timesteps to generate high-quality samples [17].

To address the semantic gap caused by the ambiguity in abstract descriptions (e.g. style text labels), we propose a fine-grained text-motion alignment using a novel body-part contrastive learning strategy. First, we employ GPT-3 [18] to extract body-part motion content descriptions from the abstract prompt. These descriptions are then converted into CLIP representations, which are aligned with the corresponding body-part motions via a contrastive objective. This strategy offers two advantages: (1) body-part descriptions are fundamental and concrete, making them easier to be aligned with motions; (2) GPT-3 can flexibly combine learned body-part descriptions based on user input prompts, enabling the generation of a diverse range of zero-shot content-specific motions, as well as more abstract stylistic movements. Additionally, given the limitations of text-based approaches in responding promptly to high-frequency, aperiodic motions on complex terrains, we also integrate terrain geometry to enhance terrain-adaptive motion generation.

Unlike the previous work [19], which employs a sequence-level framework for direct temporal-domain motion transitions, our model encodes motion sequences into a phase embedding space composed of finite channels. This encoding constructs decoupled motion embeddings by using motion's inherent periodicity, wherein smooth transitions are achieved by linear interpolation of motion embeddings corresponding to different actions. Therefore, our framework eliminates the need for large datasets with numerous transition motions and paired lengthy text descriptions, as required by [19]. The core of this framework lies in leveraging phase variables, a strategy shown to improve movement synchronization over time [1], [7], [20]. Recently, Starke et al. introduce the Periodic Autoencoder (PAE) [21] to learn the non-linear periodicity from large unstructured datasets in an unsupervised manner. In this paper, we propose the Body-Part Periodic Autoencoder (BP-PAE) to learn body-part local periodicity, ensuring coherent and plausible motion transitions while implementing specified body-part motion control.

To balance inference efficiency with high-quality sampling, we propose a novel diffusion-based framework. Our method enables real-time denoising in five inference steps by employing a compact neural network for noise prediction at each step. Additionally, we draw on Analytic-DPM [22] to analytically estimate variance per step, ensuring superior motion quality.

To this end, we propose SPORT, a method for real-time ever-changing motion generation. The architecture of SPORT comprises four key components: MoE-based motion encoder, body part-based text content encoder, recursive MoE-based content modulator and a diffusion-based motion decoder. The motion encoder, comprising a MoE model, is employed to encode motions into a non-linear phase embedding space. Simultaneously, a body part-based text content encoder is trained to extract content embeddings aligned with motion sequences using contrastive learning. These content embeddings then modify means and variances of the motion embeddings via adaptive instance normalization (AdaIN) layers [14], [23], [24] in a recursive MoE module. Finally, the modified motion embeddings are fed into the diffusion-based motion decoder, guiding the generation of probabilistic motions within only five inference steps.

In summary, contributions of our work are fourfold:

- We propose a prompt-conditioned framework for real-time motion generation, which is capable of generating ever-changing motions using zero-shot prompts.
- We develop a body-part contrastive learning strategy to bridge the semantic gap between texts and motions.
- We propose BP-PAE, an unsupervised method for learning body-part local periodicity, which improves seamless transitions between different motion types.
- We propose a diffusion model in the phase embedding space that optimally balances inference efficiency with the quality of the generated motions.

II. Related Work

A. Data-driven Motion Generation

Motion generation is a booming research area and has important application in VR/AR scenarios. The development of deep neural networks have opened up a new paradigm for the synthesis of motion conditioned on input frames. These approaches can be categorised into deterministic and probabilistic methods. For deterministic methods, Holden et al. [1] propose PFNN, a pioneering real-time framework that integrates phase-related features as parameters to synchronize locomotion with the timeline. After that, the concept of phase has been incorporated into MoEs scheme to tackle more intricate tasks like character-scene interactions [20] and multi-contact character movement [7]. However, deterministically predicted motions usually regress to the mean pose, which ignores diverse details of the motion.

In contrast to deterministic methods, probabilistic generative methods aim to describe a range of possible motions, which shows promise in preventing collapsing on a mean pose. Many GAN-related works have been implemented for motion-related tasks. For instance, Wang et al. [25] combine recurrent neural networks and adversarial training for motion modeling. Lee et al. [26] introduce a music-to-movement GAN (MM-GAN) designed to synthesize dance from music inputs. Li et al. [27] present GANimator, a generative model capable of synthesizing novel motions from a single, short motion clip. A different kind of generative model, known as normalizing flow, has also gained recent interest. Alexanderson et al. [28] utilize a flow-based model to generate attribute-controllable gesture animations. Ji et al. [29] propose FlowSMM to establish invertible transformations between motions and style latent codes. Nonetheless, it's worth noting that the computational demands of the methods based on both GANs and normalizing flows can be prohibitively high, making them unsuitable for

real-time animation systems. On the other hand, another VAE-architecture, MVAE [3] can address generation speed and motion quality simultaneously. Notably, there exists substantial evidence [30]–[32] indicating that VAE-based approaches face challenges in accurately capturing and representing diverse motion distributions, particularly in tightly-constrained tasks involving various styles or varying terrains.

Lately, diffusion models have attracted substantial interest within the realm of motion generation. Pioneering efforts such as MotionDiffuse [11] and MDM [9] focus on generating motions aligned with textual prompts using diffusion models. On one hand, some recent works attempt to enhance the model's capability to learn motion representations. For instance, MLD [10] employs a diffusion process on the motion latent space, aiming to acquire a representative and low-dimensional latent code for human motion. In contrast, TEDi [33] adapts diffusion to the temporal-axis of motion. Both methods face challenges when confronted with lengthy and constantly changing prompts. To tackle this issues, AMD [19] adopts a sequence-level autoregressive generation, enabling a seamless transition between different motion segments. However, this method relies on a large dataset that pairs long textual prompts with complex motions and is unsuitable for real-time applications due to its sequence-to-sequence generation strategy. On the other hand, some works focus on enhancing motion generation efficiency. EMDM [34] introduces a condition denoising diffusion GAN that models the complex denoising distribution and allows for a larger sampling step size. Nevertheless, EMDM is limited to sequence-level generation tasks. Additionally, certain endeavors [22], [35] focus on theoretically minimizing the number of iterative steps required for diffusion model inference, a strategy proven successful in image generation. In this work, to obtain the trade-off between quality and efficiency, we propose a autoregressive diffusion model within the latent phase manifold. For efficiency, we convert the diffusion model's input from a long time-series format to a compact phase-domain channel sequence. Additionally, our model reduces the denoising process to only five inference steps. To preserve generation quality, we draw inspiration from Analytic-DPM [22], employing analytical estimation of the optimal reverse variance at each inference step.

B. Style Transfer and Content Manipulation

Style transfer have been applied successfully in fields of image [23], talking face and human motion [24]. Early investigations use handcrafted features [36], [37] to establish a correspondence between content and style. Given the elusive nature of the term "style", contemporary methods have focused on learning style features through statistical learning techniques like linear time-invariant model [38], conditional restricted boltzmann machine [39] and mixtures of autoregressive models [40]. However, these methods require the prior specification of motion content, involving a laborious data collection process where actors must replicate the same motion content across different styles while adhering to identical steps.

Then, the emergence of deep learning has significantly advanced the task of style transfer. Holden et al. [41] propose a convolution autoencoder to encode motion styles into the Gram matrix [42] of hidden layers. Du et al. [43] improve the method by replacing the optimization procedure with a neural network. However, these methods are limited in their ability to learn a diverse range of style representations. Recently, inspired by Adaptive Instance Normalization (AdaIN) [23], Aberman et al. [24] disentangle style features from motion clips and inject them into the generation process with a temporally invariant AdaIN. Although this method can be generalized to unseen styles, the limited style features it learns may cause generated results to be overfitted to existing styles.

To realize real-time style transfer, some works [4], [5] design residual connections for style feature capture. However, these methods face limitations due to their single-style encoding capacity per residual branch, resulting in increased model complexity when scaling to multiple style categories. In contrast, Mason et al. propose a style modulation network [6] that employs feature-wise linear modulation (FiLM) [44] to effectively learn compact representations for up to 100 motion styles. However, the coupled nature of the learned style features presents challenges for generalization to unseen styles.

Recently, to achieve flexible manipulation of motion content using text prompts, methods like GestureDiffuCLIP [14] and TMR [15] align motion with textual descriptions explicitly. While GestureDiffuCLIP demonstrates promising capabilities in zero-shot style control of gestures, it exhibits two limitations: (1) it lacks a clear definition of motion content and style; (2) it directly partitions the full-body motion into several body parts, ignoring the connection between body part and global motion characteristics. To address these limitations, we define "motion content" as the fundamental movements of individual body parts and characterize "motion style" as high-level combinations of these basic movements. We then propose a contrastive learning strategy to align body-part motion content with textual descriptions in an embedding space. Both joint-wise and global transformed features are integrated into these embeddings. Subsequently, we employ GPT-3 to translate the abstract style prompts into specific body-part descriptions, employing the aligned content embeddings to generate the final stylized motion.

III. METHODOLOGY

As illustrated in Fig.2, we introduce our solution for real-time motion generation. Our system consists of four primary components. To manipulate motion content in a phase manifold, we employ a motion encoder based on MoE architecture to learn motion embeddings. Simultaneously, a body part-based content encoder is employed to extract text content embeddings aligned with motions at body part level. Following this, extracted content embeddings are fed into a recursive MoE-based content modulator to modify the motion embedding. This adaptation is achieved by using AdaIN layers within the phase manifold. Finally, the modified motion embedding is fed into the diffusion-based motion decoder to produce a probabilistic and realistic animation.

Our model operates in a frame-level autoregressive manner, where the generated output from the previous frame serves as

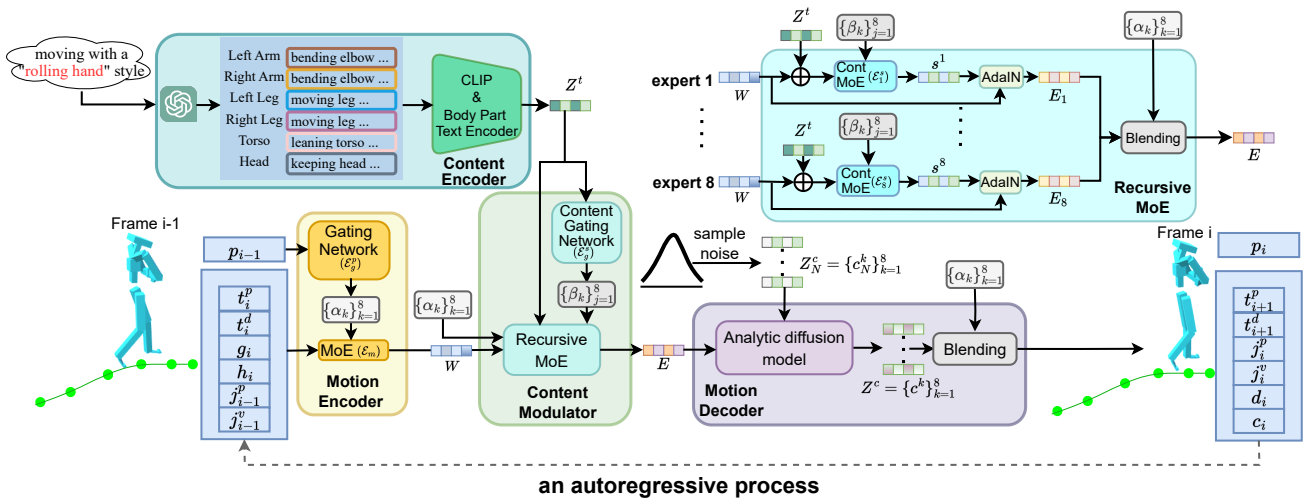


Fig. 2: **The framework of SPORT.** Our method consists of four components: (a) a motion encoder responsible for extracting the phasic motion embedding from the input variable; (b) a content encoder that utilizes GPT-3 guidance and pretrained text encoder to derive content embeddings aligned with textual descriptions at body part level; (c) a content modulator that integrates content embeddings into motion embeddings using AdaIN layers within the phase manifold; (d) a motion decoder that generates probabilistic and realistic results through a short denoising process. The whole system operates in an autoregressive fashion.

input for the motion encoder in the subsequent iteration. In the following sections, we will provide more details about the system's components and their training process.

A. Motion Representation

Our system's data formats closely resemble those of PFNN [1], comprising four components: character state, environmental information, phase variables and user control signals.

For frame i, the character state is represented by positions $\boldsymbol{j_i^p} \in \mathbb{R}^{3J}$ and velocities $\boldsymbol{j_i^v} \in \mathbb{R}^{3J}$ for J body joints, along with the root transformation $\boldsymbol{d_i} \in \mathbb{R}^3$ (containing root velocity $(r_i^x, r_i^z) \in \mathbb{R}^2$ in the horizontal XZ-plane and the angular velocity r_i^a around the Y-axis direction) and one-hot gait vectors $\boldsymbol{g_i} \in \mathbb{R}^{6L}$ that indicate the character's gait type (e.g., standing, walking, jogging, crouching, jumping and other). Here, $L = 12$ refers to a time window centered at frame i, capturing motion states from past and future every 10 frames.

Environmental information includes terrain heights $\boldsymbol{h_i} \in \mathbb{R}^{3L}$ around the trajectory, which are measured at three positions (left, right, and center) spaced 25 cm apart. It also features foot contact labels $\boldsymbol{c_i} \in \mathbb{R}^4$, which indicate whether each heel and toe joint is in contact with the ground.

Input phase variable $\boldsymbol{p_{i-1}} \in \mathbb{R}^{2\times C\times(L+1)}$ contains phase encodings from $C=5$ channels across the entire time window. In contrast, output phase variables $\boldsymbol{p_i} \in \mathbb{R}^{2\times C\times(0.5L+1)}$ only contains the future one-second time window. Both $\boldsymbol{p_i}$ and $\boldsymbol{p_{i-1}}$ are obtained through a pre-trained BP-PAE, with further details provided in the following sections.

For user control signals, trajectory positions $\boldsymbol{t_i^p} \in \mathbb{R}^{2L}$ and directions $\boldsymbol{t_i^d} \in \mathbb{R}^{2L}$ represent the user-controlled trajectory across the entire time window, projected onto the horizon X-Z plane. In contrast, the predicted trajectory variable $\boldsymbol{t_{i+1}^p} \in \mathbb{R}^L$ and $\boldsymbol{t_{i+1}^d} \in \mathbb{R}^L$ focus solely on the future half. Notably, all trajectory and body joint transformations are calculated relative to the root trajectory transformation.

Finally, the input variable for a single frame i is defined as $\boldsymbol{x_i} = \{\boldsymbol{t_i^p}, \boldsymbol{t_i^d}, \boldsymbol{h_i}, \boldsymbol{j_{i-1}^p}, \boldsymbol{j_{i-1}^v}, \boldsymbol{g_i}, \boldsymbol{p_{i-1}}\} \in \mathbb{R}^{472}$, while the complete parameterization of the output variable for frame i is given by $\boldsymbol{y_i} = \{\boldsymbol{t_{i+1}^p}, \boldsymbol{t_{i+1}^d}, \boldsymbol{j_i^p}, \boldsymbol{j_i^v}, \boldsymbol{d_i}, \boldsymbol{c_i}, \boldsymbol{p_i}\} \in \mathbb{R}^{287}$.

B. Motion Encoder

As is shown in Fig. 2, the motion encoder is composed of a 2-layer MoE network $\mathcal{E}_m$ and a gating network $\mathcal{E}_g^p$. According to [1], [2], the motion transition distribution exhibits multimodality, and the majority of its characteristics can be effectively represented by a set of time-varying principal component features. Considering a single network is prone to regress towards the average feature, $\mathcal{E}_m$ learns eight principle embedding components $\{\boldsymbol{W_k}\}_{k=1}^8 \in \mathbb{R}^{8\times 1024}$ of the motion's non-linear periodicity with eight branches of neural network experts. These component vectors are collectively utilized to reconstruct the motion embeddings, denoted as $\boldsymbol{W} = \sum_{k=1}^8 \alpha_k \boldsymbol{W_k} \in \mathbb{R}^{1024}$, where the phase blending coefficients $\{\alpha_k\}_{k=1}^8 \in \mathbb{R}^8$ are computed by the gating network $\mathcal{E}_g^p$.

$\mathcal{E}_g^p$ takes as input the phase variable $\boldsymbol{p_{i-1}}$, which is derived from a pretrained Body Part Periodic Autoencoder (BP-PAE). Starke et al. [21] were the first to introduce periodic autoencoders (PAEs), allowing motion to be clustered on a continuous multi-dimensional phase manifold. Given the velocity sequence $J_i^v \in \mathbb{R}^{\mathcal{T}\times 3J}$ centered at $\boldsymbol{j_i^v}$, PAE encodes J_i^v into lower-dimensional phase channels, denoted as $L_i = E_{pae}(J_i^v) \in \mathbb{R}^{\mathcal{T}\times C}$, where the time window $\mathcal{T}_{-1s}^{1s} = 121$ and C represents the number of phase channels. L_i is then parameterized by several sinusoidal functions, characterized by amplitude $\boldsymbol{a_i} \in \mathbb{R}^5$, frequency $\boldsymbol{f_i} \in \mathbb{R}^5$, offset $\boldsymbol{b_i} \in \mathbb{R}^5$, and phase shift $\boldsymbol{s_i} \in \mathbb{R}^5$. Finally, each phase encoding $\mathcal{P}_i \in \mathbb{R}^{5\times 2}$ at frame i is represented as:

$$\mathcal{P}_i^{2j-1} = \boldsymbol{a_i^j} \cdot sin(2\pi \cdot \boldsymbol{s_i^j}), \quad \mathcal{P}_i^{2j} = \boldsymbol{a_i^j} \cdot cos(2\pi \cdot \boldsymbol{s_i^j}), \quad (1)$$

where j is the channel index. To enhance the temporal alignment of movements, the final phase variable $\boldsymbol{p_i}$ contains phase

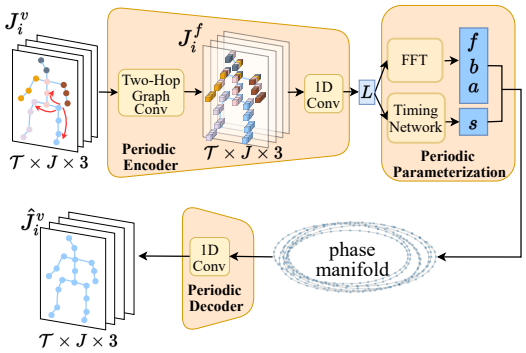


Fig. 3: **The training process of BP-PAE.** The process involves: a periodic encoder, a periodic parameterization process and a periodic decoder. To learn nonlinear periodic features at both the body-part and whole-body levels, the periodic encoder applies two-hop graph convolution. In the illustration, we take the left hip joint as an example. Its adjacent joints in two hops include joints within the body part (left knee and left ankle) and joints around hip (right hip, spine, and hip).

encodings sampled every 10 frames over the time window $\mathcal{T}_{-1s}^{1s}$ as $p_i = \{\mathcal{P}_{i-60}, ..., \mathcal{P}_{i-10}, \mathcal{P}_i, \mathcal{P}_{i+10}, ..., \mathcal{P}_{i+60}\}$.

Since PAE learns full-body periodic features in an unsupervised manner, it struggles to capture local periodicity, which is crucial for maintaining the consistency of other body parts when modifying specific body-part action. To address this, we introduce the Body Part Periodic Autoencoder (BP-PAE) to incorporate body-part periodicity into the unsupervised learning process. We achieve this by utilizing graph convolution within two-hop joint neighborhoods to encode each joint and its correlation with other inner-part joints into a learnable token. This produces a feature sequence $J_i^f = E_{graph}(J_i^v) \in \mathbb{R}^{\mathcal{T} \times J \times 3}$, which is subsequently processed by $E_{pae}(\cdot)$ as described earlier. Due to the articulated topology of the human skeleton, the correlation features learned in the two-hop joint neighborhood only involve joints in the same body part (body part features) or extra spinal joints (global transformation features). As a result, BP-PAE enables the learning of nonlinear periodic features at both the body-part and whole-body levels.

C. Content Encoder

Our goal is to establish a joint embedding space for text prompts and motion, forming the foundation for text-driven motion control. Current methods [8], [14], [15] suffer from a notable semantic gap between textual representations and corresponding movements. This disparity stems from two factors: (1) the complexity in describing quickly changing movements, (e.g., interactive actions on complex terrain); (2) the ambiguity in abstract descriptions.

To address the first challenge, we incorporate terrain topology information $\boldsymbol{h_i}$ into the input variable $\boldsymbol{x_i}$. To further resolve the textual ambiguity, we employ a multi-faceted approach: initially, we leverage GPT-3 to convert the abstract prompt into more concise and precise body-part content descriptions. Subsequently, we introduce a body-part contrastive learning strategy to achieve fine-grained text-motion alignment. Additionally, We train an additional motion sequence decoder to make body-part content embeddings more conductive to whole-body motion generations. Notably, we opt for contrastive learning in the temporal domain rather than in the phase manifold. In the temporal domain, segmenting the entire body into distinct body parts is straightforward. Conversely, in the phase manifold, unsupervised learning of periodicity lacks a clear definition of body parts, making text-motion alignment at the body-part level challenging.

As is shown in Fig. 4, the training of text-motion alignment occurs at the body-part level. Following TEMOS [45], we apply a transformer-based VAE encoder-decoder architecture. Our proposed motion sequence encoder is composed of a Joint Transformer Encoder $\mathcal{E}_J$ and a Temporal Transformer Encoder $\mathcal{E}_T$, designed to capture the hierarchical spatio-temporal structure of the motion sequence. Within $\mathcal{E}_J$, we incorporate a novel body part-aware attention mechanism that aggregates inner-part joints' features and global motion features into each body part's learnable token with 64 dimensions. Subsequently, the body part embeddings of length T, denoted as $Z_{1:T}^J \in \mathbb{R}^{T \times 6 \times 64}$, are fed into $\mathcal{E}_T$ to integrate each body part's temporal features into their respective VAE distribution parameter tokens $\{\mu_i^p + \Sigma_i^p\}_{i=0}^5 \in \mathbb{R}^{2 \times 6 \times 64}$ isolately.

Simultaneously, we train a text encoder to map the abstract prompt into a shared VAE parameter embedding space. Inspired by SINC [46], we initially employ GPT-3 to convert the abstract prompt into body-part content description. However, unlike SINC, our approach instruct GPT-3 to describe the content of all six body parts (left arm, right arm, left leg, right leg, torso and head), rather than restricting it to specific parts associated with the target content. This process is accomplished through a structured prompt comprising two components: (a) Prompt Specification, which provides GPT-3 with lists of body joints, verbs, adverbs and adjectives for use in characterizing movement or positioning, and (b) Question-Answer Examples, which guide GPT-3 in generating descriptive contents. Questions are formatted as "Describe the $< style/action >$ into six body parts with provided lists of body joints, verbs, adverbs and adjectives", and the answers furnish detailed descriptions involving six body parts and relevant verbs/adverbs/adjectives. After providing some question-answer examples, GPT-3 autonomously extracts descriptive texts from the input abstract prompt. We then take the CLIP text embeddings of these descriptions $\{\boldsymbol{z_i^c}\}_{i=0}^5 \in \mathbb{R}^{6 \times 512}$ as the language prior and translate them into body-part VAE parameters $\{\mu_i^t + \Sigma_i^t\}_{i=0}^5 \in \mathbb{R}^{2 \times 6 \times 64}$ via a body-part MLP.

Subsequent to the operation of vector concatenation, we can obtain $\mu^p + \Sigma^p \in \mathbb{R}^{2 \times 384}$ and $\mu^t + \Sigma^t \in \mathbb{R}^{2 \times 384}$, which represent the VAE parameters for body-part motion sequences and textual descriptions, respectively. To improve the alignment, we train these encoders via an InfoNCE-based contrastive loss [15]. Given N_b pairs of VAE parameters $(\mu_0^p + \Sigma_0^p, \mu_0^t + \Sigma_0^t), ..., (\mu_{N_b-1}^p + \Sigma_{N_b-1}^p, \mu_{N_b-1}^t + \Sigma_{N_b-1}^t)$, the objective during training is to maximize the similarity of the parameters within the same pair $(\mu_i^p + \Sigma_i^p, \mu_i^t + \Sigma_i^t)$ and simultaneously maximize the differences between parameters from different pairs $(\mu_i^p + \Sigma_i^p, \mu_j^t + \Sigma_j^t)_{i \neq j}$. The contrastive loss

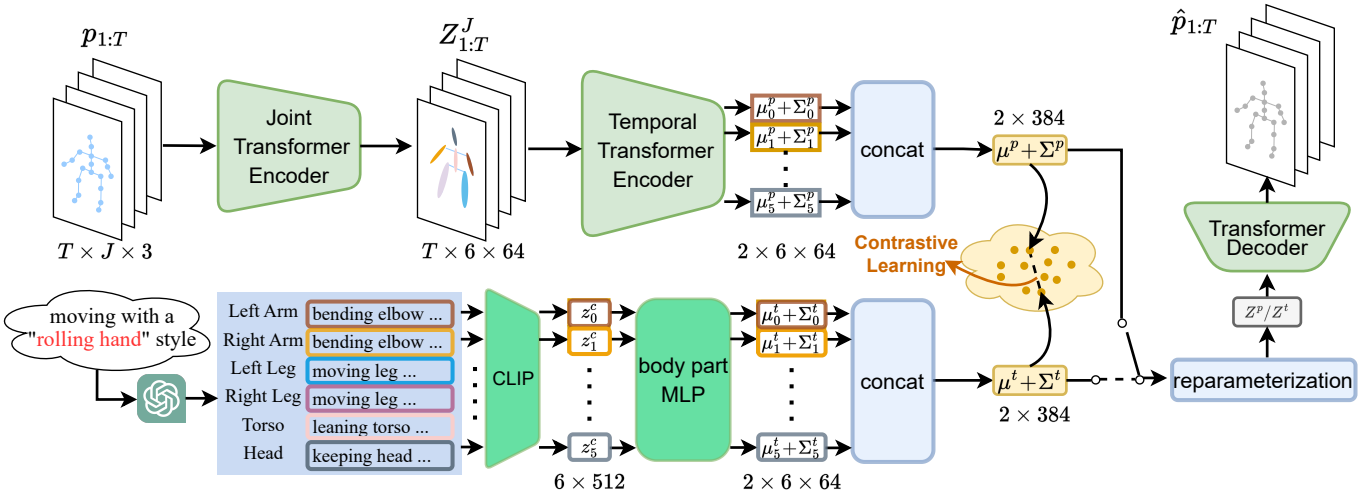


Fig. 4: **The training process of the text-motion joint embedding space.** Initially, a joint transformer encoder is utilized to transform a motion sequence $p_{1:T}$ into body part embeddings $Z^J_{1:T}$, which are then aggregated into sequence VAE parameters $\{\mu^p_i + \Sigma^p_i\}^5_{i=0}$ for 6 body parts via a temporal transformer encoder. Concurrently, after the steps of GPT-guidance and CLIP-based text encoding, a body part MLP is trained to extract textual VAE parameters $\{\mu^t_i + \Sigma^t_i\}^5_{i=0}$ for 6 body parts. The training of these encoders involves a explicit supervision for aligning text and motion through a contrastive loss as well as a implicit supervision for the generated motions derived from the joint content embedding space using a reconstruction loss.

can be defined as:

$$\mathcal{L}_{con} = -\frac{1}{2N_b}\sum_i (\log \frac{\exp(S_{ii}/\tau)}{\sum_j \exp(S_{ij}/\tau)} + \log \frac{\exp(S_{ii}/\tau)}{\sum_j \exp(S_{ji}/\tau)}), \tag{2}$$

where $S_{ij} = \cos(\mu^p_i + \Sigma^p_i, \mu^t_j + \Sigma^t_j)$ computes the pairwise cosine similarities and τ is the temperature hyperparameter.

Moreover, we implicitly conduct text-motion alignment supervision by sampling sequence embedding $\boldsymbol{Z^p} \in \mathbb{R}^{384}$ and text content embedding $\boldsymbol{Z^t} \in \mathbb{R}^{384}$ from the VAE latent distributions $\mathcal{N}(\boldsymbol{\mu^p}, \boldsymbol{\Sigma^p})$ and $\mathcal{N}(\boldsymbol{\mu^t}, \boldsymbol{\Sigma^t})$, and feed them into a transformer-based motion sequence decoder $\mathcal{D}_s$ to reconstruct the motion sequence $\hat{p}_{1:T}$. The reconstruction loss is defined as:

$$\mathcal{L}_{recon} = \mathcal{L}_1(p_{1:T}, p^p_{1:T}) + \mathcal{L}_1(p_{1:T}, p^t_{1:T}), \tag{3}$$

where $\mathcal{L}_1$ denotes the smooth L1 Loss, $p^p_{1:T}$ and $p^t_{1:T}$ respectively refer to the reconstructed motion sequences originating from distributions $\mathcal{N}(\boldsymbol{\mu^p}, \boldsymbol{\Sigma^p})$ and $\mathcal{N}(\boldsymbol{\mu^t}, \boldsymbol{\Sigma^t})$.

D. Content Modulator

In the content modulator, we propose a recursive MoE to constructs a content sub-space within the phase embedding space. Specifically, in Fig. 2, recursive MoE operates in two recursive steps. In the inner step, we create a Cont MoE $\mathcal{E}^s_k$ within k-th expert branch to learn content components $\{\boldsymbol{s^k_j}\}^8_{j=1} \in \mathbb{R}^{8\times 2048}$, derived from the fusion of motion embedding $\boldsymbol{W}$ and text content embedding $\boldsymbol{Z^t}$. The assumption is made that dynamic blending of content components suffices to represent the motion content feature of k-th expert branch. Utilizing a uniform set of content blending coefficients $\{\beta_j\}^8_{j=1} \in \mathbb{R}^8$ across all experts, the motion content feature of k-th expert branch, denoted as $\boldsymbol{s^k} = \sum^8_{j=1} \beta_j \boldsymbol{s^k_j} \in \mathbb{R}^{2048}$, is extracted, where $\{\beta_j\}^8_{j=1} \in \mathbb{R}^8$ are computed by the content gating network $\mathcal{E}^s_g(\cdot)$, which takes $\boldsymbol{Z^t}$ as input. This approach allows us to encode all content characteristics within the phase manifold using only eight Cont MoEs, denoted as $\{\mathcal{E}^s_k\}^8_{k=1}$. In the outer step, k-th expert initially modifies the mean and variance of the original motion embedding $\boldsymbol{W}$ via an AdaIN layer, producing the modified motion embedding $\boldsymbol{E_k} \in \mathbb{R}^{1024}$ of k-th expert. The full-body modified motion embedding is then obtained by blending $\{\boldsymbol{E_k}\}^8_{k=1} \in \mathbb{R}^{8\times 1024}$ with the phase blending coefficients $\{\boldsymbol{\alpha_k}\}^8_{k=1} \in \mathbb{R}^8$, denoted as $\boldsymbol{E} = \sum^8_{k=1} \alpha_k \boldsymbol{E_k} \in \mathbb{R}^{1024}$.

E. Motion Decoder

In this section, we introduce a diffusion model to generate the output variable $\boldsymbol{y_i}$. Unlike previous work, we apply a frame-level diffusion process conditioned on the modified full-body motion embedding $\boldsymbol{E}$ to compute the modified motion code $Z^c_0 = \{\boldsymbol{c^k_0}\}^8_{k=1} \in \mathbb{R}^{8\times 287}$ across only eight compact phase channels, rather than over a longer time sequence. Formally, the motion decoder computes:

$$Z^c_0 = \mathcal{G}(\boldsymbol{E}, Z^c_N), \tag{4}$$

where $Z^c_N \sim \mathcal{N}(0, \mathbf{I})$ is the input Gaussian noise. Finally, the output variable $\boldsymbol{y}_i$ is a linear combination of these modified motion codes, denoted as $\boldsymbol{y}_i = \sum^8_{k=1} \alpha_k \boldsymbol{c^k_0}$.

Diffusion models begin with a forward process $q(Z^c_{1:N}|Z^c_0)$ to introduce noise into the motion code distribution $q(Z^c_0)$ and then reverse the forward process to recover it. Since the forward process involves N steps, the reverse process also follows a sampling procedure of N steps. Recently, DDIM [47] accelerates the reverse process to within $dim(\tau)$ steps:

$$\begin{aligned} &q_\lambda(Z^c_{\tau_{n-1}}|Z^c_{\tau_n}) \approx q_\lambda(Z^c_{\tau_{n-1}}|Z^c_{\tau_n}, Z^c_0 = \hat{\mu}_\theta(Z^c_{\tau_n}, \tau_n)) \\ &= \mathcal{N}(Z^c_{\tau_{n-1}}|\tfrac{\sqrt{\bar{\alpha}_{\tau_{n-1}}}}{\sqrt{\bar{\alpha}_{\tau_n}}}\left(Z^c_{\tau_n} - \left(\sqrt{\bar{\beta}_{\tau_n}} - \tfrac{\sqrt{\bar{\alpha}_{\tau_n}}}{\sqrt{\bar{\alpha}_{\tau_{n-1}}}}\sqrt{\bar{\beta}_{\tau_{n-1}} - \lambda^2_{\tau_n}}\right)\cdot\epsilon_\theta, \lambda^2_{\tau_n}\mathbf{I}\right), \end{aligned} \tag{5}$$

$$\hat{\mu}_\theta(Z^c_{\tau_n}, \tau_n) = \frac{1}{\sqrt{\bar{\alpha}_{\tau_n}}}(Z^c_{\tau_n} - \sqrt{\bar{\beta}_{\tau_n}}\epsilon_\theta(Z^c_{\tau_n}, \tau_n)), \tag{6}$$

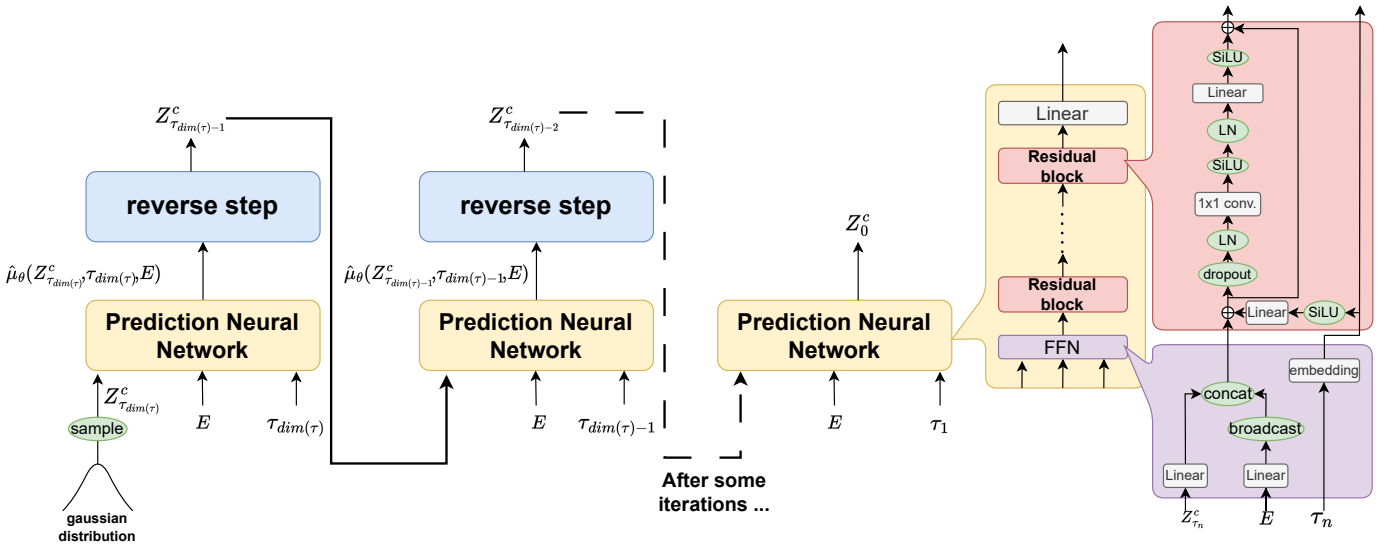


Fig. 5: **(Left) The generative process of the diffusion model.** The accelerated generation progress starts with a sub-sequences of $[1, ..., N]$, denoted as $\tau = [\tau_1, \tau_2, ..., \tau_{dim(\tau)}]$. The initial iteration step involves the following key actions: (a) sampling the random noise $Z^c_{\tau_{dim(\tau)}}$; (b) utilizing the prediction Neural Network $\hat{\mu}_\theta$ to estimate $\hat{Z}^c_0$ conditioned on: the noise $Z^c_{\tau_{dim(\tau)}}$, the modified motion embedding E and the index of step $\tau_{dim(\tau)}$; (c) obtaining the state of next step $Z^c_{\tau_{dim(\tau)-1}}$, following the Eq. 5. By repeating these iterative steps $dim(\tau)$ times, we can finally obtain the final Z^c_0. **(Right) The architecture of the prediction neural network.** The model has two kinds of components: (a) a feedforward network that handles multimodal inputs; (b) residual block designed to deals with serial and spatial information, utilizing 1D convolution and FC layers, respectively.

where α_n and β_n are scalars, with $\alpha_n = 1 - \beta_n$, $\overline{\alpha}_n = \prod_{i=1}^{n} \alpha_i$ and $\overline{\alpha}_n = 1 - \overline{\beta}_n$. $\mathbf{I}$ is the identity matrix. Z^c_N tends to an isotropic Gaussian distribution when $N \longrightarrow \infty$. $\tau = [\tau_1, \tau_2, ..., \tau_{dim(\tau)}]$ is a sub-sequence of $[1, ..., N]$. $\lambda = (\lambda_1, \cdots, \lambda_N) \in \mathbb{R}^N_{\geq 0}$ represents a set of unspecified non-negative coefficients. $\epsilon_\theta(Z^c_{\tau_n}, \tau_n)$ is a function approximator intended to predict noise from $Z^c_{\tau_n}$ and step index τ_n.

While reducing the iteration count in the reverse process can significantly improve computational efficiency, it often compromises the quality of generated results. To maintain comparable performance while accelerating the reverse process, we adopt the optimal reverse variance proposed by [22] to replace the covariance term $\lambda^2_{\tau_n}\mathbf{I}$ in Eq. 5 with $(\lambda^2_{\tau_n} + (\sqrt{\overline{\alpha}_{\tau_{n-1}}} - \sqrt{\overline{\alpha}_{\tau_n}}\kappa_{\tau_n})^2 \bar{\sigma}^2_{\tau_n})\mathbf{I}$, where $\kappa_{\tau_n} = \frac{\sqrt{\overline{\beta}_{\tau_{n-1}} - \lambda^2_{\tau_n}}}{\sqrt{\overline{\beta}_{\tau_n}}}$, $\bar{\sigma}^2_{\tau_n}$ can be estimated using the Monte Carlo method: $\bar{\sigma}^2_{\tau_n} = \frac{\bar{\beta}_{\tau_n}}{\overline{\alpha}_{\tau_n}} \left(1 - \frac{1}{d}\mathbb{E}_{Z^c_{\tau_n} \sim q(Z^c_{\tau_n})} \left[\|\epsilon_\theta\|^2\right]\right)$, d is the dimension of $Z^c_{\tau_n}$.

In our approach, we use a diffusion model to generate the modified motion content code Z^c conditioned on the modified motion embedding $\boldsymbol{E}$. Consequently, the above reverse step $q_\lambda(Z^c_{\tau_{n-1}}|Z^c_{\tau_n})$ takes a conditional form: $q_\lambda(Z^c_{\tau_{n-1}}|Z^c_{\tau_n}, \boldsymbol{E})$. The details of the generative process are shown in Fig. 5.

Following [48], we propose a neural network to directly predict $\hat{\mu}_\theta(Z^c_{\tau_n}, \tau_n, \boldsymbol{E})$ rather than the noise term $\epsilon_\theta(Z^c_{\tau_n}, \tau_n, \boldsymbol{E})$. $\epsilon_\theta(Z^c_n, n, \boldsymbol{E})$ can be achieved in reverse as detailed in Eq. 6. Our design for $\hat{\mu}_\theta(\cdot)$ follows two principles: (1) it should be lightweight for real-time operation; (2) it should be compatible with ONNX format for Unity integration. As a result, $\hat{\mu}_\theta(\cdot)$ is built using common operations such as 1D Convolution (Conv1d) layers, Fully Connected (FC) layers, Layer Normalization (LN) and Sigmoid Linear Unit (SiLU) functions.

As depicted in Fig. 5, $\hat{\mu}_\theta(\cdot)$ comprises two main modules: a feedforward network (FFN) and residual blocks. The FFN functions as an encoder, hierarchically fusing the multimodal inputs $(Z^c_{\tau_n}, \boldsymbol{E}, \tau_n)$. First, FC layers project $Z^c_{\tau_n}$ and $\boldsymbol{E}$ into high-dimensional feature spaces independently. Then, $\boldsymbol{E}$ is expanded to a length of 8, allowing it to be concatenated with $Z^c_{\tau_n}$ and ensuring that each expert component of $Z^c_{\tau_n}$ effectively incorporates motion features during the denoising process. To provide the residual blocks with iteration order information, the FFN also transforms τ_n into positional embeddings. Each residual block contains a Conv1d layer and a FC layer, responsible for extracting serial and spatial information from Z^c_0. To avoid vanishing gradient problem, skip-connections are employed, as detailed in [49].

F. Losses

The training process of SPORT primarily comprises three distinct stages. In the first stage, we only train the content encoder via a body-part contrastive learning strategy mentioned in Sec. III-C. In addition to $\mathcal{L}_{con}$ and $\mathcal{L}_{recon}$, inspired by TEMOS [45], we further supervise the VAE architecture by leverage a Kullback-leibler (KL) divergence loss $\mathcal{L}_{KL}$ and an embedding similarity loss $\mathcal{L}_{emb}$. Specifically,

$$\mathcal{L}_{stage1} = \mathcal{L}_{recon} + \lambda_{con}\mathcal{L}_{con} + \lambda_{KL}\mathcal{L}_{KL} + \lambda_{emb}\mathcal{L}_{emb}, \quad (7)$$

$$\mathcal{L}_{KL} = KL(\phi^t, \phi^p) + KL(\phi^p, \phi^t) + KL(\phi^t, \psi) + KL(\phi^p, \psi), \quad (8)$$

$$\mathcal{L}_{emb} = \mathcal{L}_1(\boldsymbol{Z^p}, \boldsymbol{Z^t}), \quad (9)$$

where $\phi^t = \mathcal{N}(\boldsymbol{\mu^t}, \boldsymbol{\Sigma^t})$, $\phi^p = \mathcal{N}(\boldsymbol{\mu^p}, \boldsymbol{\Sigma^p})$, $\psi = \mathcal{N}(0, I)$.

Due to the challenges of joint training, in the second stage, we replace the diffusion-based motion decoder with a 1-layer MoE-based decoder $\mathcal{E}_d$ to train all modules except the motion

decoder and pretrained content encoder. The loss function for this stage combines reconstruction and contact losses:

$$\mathcal{L}_{stage2} = \mathcal{L}_{rec} + \mathcal{L}_{contact}, \tag{10}$$

$$\mathcal{L}_{rec} = \mathcal{L}_{mse}(\boldsymbol{M_i}, \hat{\boldsymbol{M}}_{\boldsymbol{i}}), \tag{11}$$

$$\mathcal{L}_{contact} = \mathcal{L}_{mse}(\boldsymbol{c_i}, \delta(\hat{\boldsymbol{f}}_{\boldsymbol{v}}^{\boldsymbol{2}})), \tag{12}$$

$$\delta(\hat{f}_v^2) = \begin{cases} 1, & \hat{f}_v^2 \leq 0.02, \\ 0, & \hat{f}_v^2 > 0.02. \end{cases} \tag{13}$$

where $\boldsymbol{M_i} = \{\boldsymbol{t}_{\boldsymbol{i+1}}^{\boldsymbol{p}}, \boldsymbol{t}_{\boldsymbol{i+1}}^{\boldsymbol{d}}, \boldsymbol{j}_{\boldsymbol{i}}^{\boldsymbol{p}}, \boldsymbol{j}_{\boldsymbol{i}}^{\boldsymbol{v}}, \boldsymbol{d_i}, \boldsymbol{p_i}\}$ contains information regarding trajectory, joints, displacement and phase, while $\hat{\boldsymbol{M}}_{\boldsymbol{i}}$ denotes predicted results from SPORT. $\mathcal{L}_{mse}$ is the standard MSE loss, $\hat{\boldsymbol{f}}_{\boldsymbol{v}} \in \mathbb{R}^4$ refers to predicted foot velocities (left/right ankle/feet) derived from $(\hat{\boldsymbol{j}}_{\boldsymbol{i-1}}^{\boldsymbol{p}}, \hat{\boldsymbol{j}}_{\boldsymbol{i}}^{\boldsymbol{p}})$.

In the third stage, we initially train the $\hat{\mu}(\cdot)$ of motion decoder using the following loss function:

$$\mathcal{L}_{stage3} = \mathcal{L}_{mse}(Z^c, \hat{Z}^c), \tag{14}$$

where Z^c is the output from the pretrained $\mathcal{E}_d$ and $\hat{Z}^c$ is generated by the motion decoder. After 10 epochs, we apply additional $\mathcal{L}_{stage2}$ supervision to refine $\hat{\mu}(\cdot)$.

G. Implementation Details

We configure $\mathcal{E}_J$ with 1 transformer layer, 1 heads, a dropout rate of 0.1 and a feedforward dimension of 256. For both $\mathcal{E}_T$ and $\mathcal{D}_S$, we use 6 transformer layers, 4 attention heads, a dropout rate of 0.1 and a feedforward dimension of 1024. In the latent diffusion model, the forward process consists of $N = 1000$ steps, with the non-negative coefficient computed as $\lambda_n = \frac{\bar{\beta}_{n-1}}{\bar{\beta}_n}\beta_n$. The reverse process has been accelerated to $dim(\tau) = 5$ iterative steps, and $\hat{\mu}(\cdot)$ employs 12 residual blocks. During training, we set the sequence length T to 40 and the batch size to 1024. The loss weights are configured as follows: λ_{con} to $1e-2$, while λ_{emb} and λ_{KL} were set to $1e-5$. We employ a dropout rate of 0.3 with the initial learning rate and weight decay both set to $1e-4$. In the first training stage, we use the AdamW optimizer with the fixed learning rate. In the second stage, following [21], we employ the AdamWR optimizer with cosine annealing warm-restart scheduling, configuring the iterations to 10 and the restart factor to 2.0. In the final training stage, we revert to the AdamW optimizer, starting with a learning rate of $4e-4$ and reducing it to $1e-5$ after 373,500 steps. Our model is implemented using PyTorch. We further develop an ONNX version suitable for execution in Unity.

IV. Experiment and Evaluation

A. Datasets

To evaluate SPORT's ability to address the semantic gap in long-term motion generation, we conduct experiments using mixed batches of stylized and terrain-fitting datasets. The first dataset, 100STYLE [6], consists of 100 styles of locomotion performed on flat terrain. The second dataset, a terrain-fitting dataset [1], covers fundamental motion scenarios (e.g., walking, jogging, crouching, jumping, beam balancing and climbing) across various terrains. To ensure compatibility with 100STYLE, we downsample the terrain-fitting dataset from 120fps to 60fps. The textual annotation process is conducted differentially across the datasets. For 100STYLE, we annotate body-part content associated with each stylized motion, utilizing the predefined vocabulary introduced in the prompt specification process (Sec. III-C). Conversely, the terrain-fitting dataset is annotated solely for body-part descriptions corresponding to the motion content. Furthermore, we augment each frame with phase variables by utilizing BP-PAE.

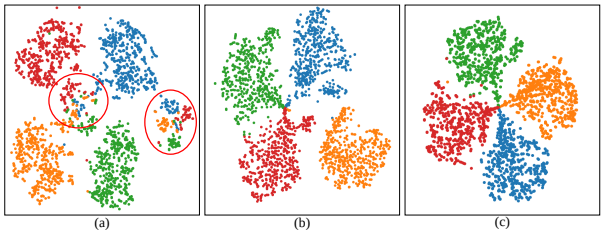


Fig. 6: **The comparison of motion embedding spaces.** Components obtained from PFNN (a), PAE (b) and BP-PAE (c) are assessed within the 2D embedding space via t-SNE. The latent space of PFNN exhibits two distinct regions where feature coupling is evident. In contrast, the motion embeddings derived from both PAE and BP-PAE display a more pronounced clustering with a noticeable decoupling effect.

B. Ablation Study

To put the performance of SPORT in perspective, we conduct thorough evaluations to validate various design choices of our framework, including the effectiveness of BP-PAE, diffusion models, MoE and body part-based content encoder.

1) The Evaluation on BP-PAE: As described in Sec. III-B, the motion embedding space is constructed through the 2-layer MoE $\mathcal{E}_m$, where the gating network $\mathcal{E}_g^p$ leverages periodic features extracted from phase variables to allocate feature subdomains to individual expert branches of $\mathcal{E}_m$ for specialized learning. Our approach leverages BP-PAE to generate phase variables that encapsulate both body part-aware and non-linear periodicity. The non-linear periodicity ensures expert branches to learn principal motion patterns in a decoupled manner, while the body part-aware periodicity ensures temporal coherence during body part-level motion editing.

To validate BP-PAE's capability in learning non-linear periodic features, we conduct a comparative analysis of motion embedding spaces, as illustrated in Fig. 6. The analysis employs identical MoE network architectures while varying the phase variable constraints across three conditions. The analysis examines phase variables derived from: (1) the footstep phase methodology proposed in PFNN [1]; (2) the PAE framework; (3) our proposed BP-PAE. To ensure experimental parity, given the limited periodic features inherent in footstep phases, we standardize the number of expert networks in the MoE architecture to four across all experimental conditions. The experimental results, visualized in Fig. 6, demonstrate notable differences in embedding space characteristics. The PFNN-based motion embedding space (Fig. 6 (a)) exhibits four distinct modes. However, feature coupling is evident in

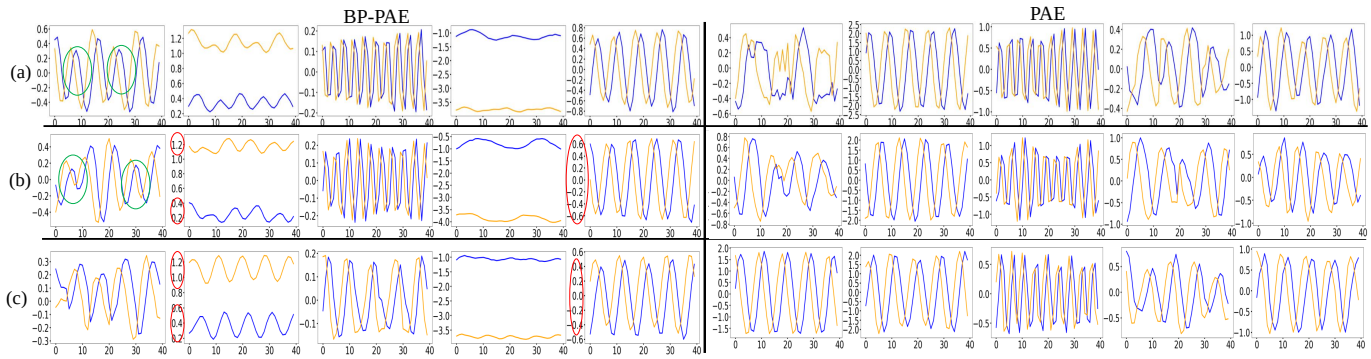


Fig. 7: **The comparison of phase variables.** We compare phase variables generated by BP-PAE and PAE, analyzing amplitude and frequency characteristics across three styles: (a) akimbo with single-hand elevation above the head, (b) standard akimbo and (c) floor sweeping. For BP-PAE, kinematically similar motions—(a) and (b)—demonstrate obvious variation only in the first high-frequency channel (denoted by green annotations), while maintaining consistency across other channels. Motion (b) and (c), which are semantically distinct, exhibit similarity only in the fourth low-frequency channel, with differentiation across remaining channels. PAE, however, produces obvious differentiation even between kinematically similar motions (a) and (b).

two regions. In contrast, both PAE and BP-PAE frameworks (Fig. 6 (b) and (c), respectively) facilitate more effective mode separation, demonstrating superior decoupling of motion features within their respective embedding spaces.

To further evaluate BP-PAE's capacity in unsupervised learning of body part-aware periodicity, we conduct a comparative analysis between phase variables generated by BP-PAE and PAE, as depicted in Fig. 7. The analysis focuses on three motion types: akimbo with one hand elevated above the head, standard akimbo and floor sweeping. The analysis reveals that BP-PAE successfully differentiates between high- and low-frequency features through unsupervised learning, resulting in the formation of two distinct low-frequency channels and three high-frequency channels. The low-frequency components exhibit larger magnitude characteristics, whereas the high-frequency components display more modest magnitude profiles. Furthermore, these frequency components demonstrate adaptive variation that correlates with the degree of differentiation between motions: kinematically similar motions that vary by a single body part primarily exhibit differences within a single high-frequency channel of the phase space, while significantly different motions reveal considerable variations across multiple channels. In contrast, PAE's inability to decouple body-part periodicity from whole-body periodicity leads to the generation of mixed high- and low-frequency characteristics. This limitation results in substantial phase variable differentiation, even among similar motions. Subsequent visualization analysis suggests that this characteristic compromises PAE's capacity to generate high-quality motions.

2) The Evaluation on Diffusion Model: To assess the effectiveness of the autoregressive diffusion model, we maintain the other components of SPORT as constant while introducing variations in the motion decoder (diffusion model or 1-layer MoE decoder $\mathcal{D}_{MoE}$). Our evaluation focuses on three key aspects: motion quality, diversity and model complexity.

Assessing motion quality in a quantitative manner remains a challenging task. One feasible solution involves employing the Fréchet Motion Distance (FMD) [50], [51], which compares the distribution of the generated motions with that of the dataset. This comparison gauges the overall ability of the diffusion model in learning the authentic data distribution. Moreover, following MANN, we assess the local motion quality of each frame by measuring the foot skating artifacts. The amount of foot skating s_f is calculated by adding the horizontal component of the foot velocity v_f if the foot height h_f is within a threshold $H = 2.5cm$: $s_f = v_f \cdot clamp(2 - 2^{h_f/H}, 0, 1)$.

To evaluate motion diversity, inspired by the metric of multimoldality [52], we propose to randomly sample a set of 40 motion clips with the size of 180 frames, denoted as $p_{i,t}$, where i is the index of clip and t is the index of frame. For each motion clip, we keep the control variables $\{t^p, t^d, t^g, t^h\}$ of $p_{i,t}$ unchanged and generate a pair of motion clips like $(\hat{p}^1_{i,t}, \hat{p}^2_{i,t})$. The diversity of this motion set is formalized as: $diversity = \frac{1}{7200} \sum_{i=1}^{40} \sum_{t=1}^{180} \left\| \hat{p}^1_{i,t} - \hat{p}^2_{i,t} \right\|_2$.

As presented in Tab. I, the integration of a diffusion model has notably improved the learning of motion distribution, as indicated by the FMD scores (0.96 vs. 4.33). However, this improvement comes at the expense of increased foot skating (0.16 vs. 0.13), extended runtime (52.09 ms vs. 12.29 ms) and a larger model parameter count (55.46M vs. 47.98M). Our supplementary video provides further evidence that SPORT, aided by the diffusion model, can generate superior motions in challenging terrain scenarios. In the context of motion generations, we argue that the FMD metric should be prioritized, as foot skating can be addressed through post-processing and the current hardware capabilities are sufficient to execute our model. Specifically, the ONNX version of SPORT achieves a 24 FPS performance on a laptop equipped with an RTX 2060 graphics card. Conversely, when evaluating the diversity, we observe that the diversity score of SPORT does not exhibit a significant increase compared to that of SPORT-w/o-Diffusion (2.47×10^{-2} vs. 0.0). This distinction is discernible in the supplementary video when the character remains stationary. However, when the avatar is in motion, the contrast between these two models appears less pronounced.

TABLE I: Comparison on FMD, foot skating, diversity, runtime, number of parameters, number of styles, motions across different terrains and transitions between different styles

Models	FMD ↓	Foot Skating ↓	Diversity ↑	Runtime (ms) ↓	#Params ↓	#Styles ↑	Multi-terrain?	Seamless Trans?
Ours-4Experts	1.27	0.21	2.32×10^{-2}	46.27	**31.49M**	100+	✓	✓
Ours-4ContExperts	1.18	0.17	2.28×10^{-2}	49.67	47.05M	100+	✓	✓
Ours-w/o-Diffusion	4.33	**0.13**	0.00	**12.29**	47.98M	100+	✓	✓
Ours-PAE	0.99	0.19	2.41×10^{-2}	52.09	55.46M	100+	✓	✓
Ours-w/o-ContEnc	1.06	0.23	2.18×10^{-2}	53.65	66.64M	100+	✓	✓
Ours	**0.96**	0.16	$\mathbf{2.47 \times 10^{-2}}$	52.09	55.46M	100+	✓	✓
StyleERD [5]	2.31	0.57	0.00	22.85	**0.48M**	7	×	×
Mason et al. [4]	6.98	0.27	0.00	16.31	4.32M	58	×	×
Mason et al. [6]	3.15	0.20	0.00	**8.30**	18.42M	100	×	×
Motion Puzzle [50]	9.44	0.41	0.00	95.20	37.16M	**100+**	×	×
Ours(100STYLE)	**1.76**	**0.11**	$\mathbf{2.31 \times 10^{-2}}$	52.09	55.46M	**100+**	×	✓
PFNN	10.25	**0.03**	0.00	**7.55**	**2.39M**	1	✓	×
Ours(PFNN dataset)	**1.94**	0.09	$\mathbf{2.16 \times 10^{-2}}$	52.09	47.71M	1	✓	✓

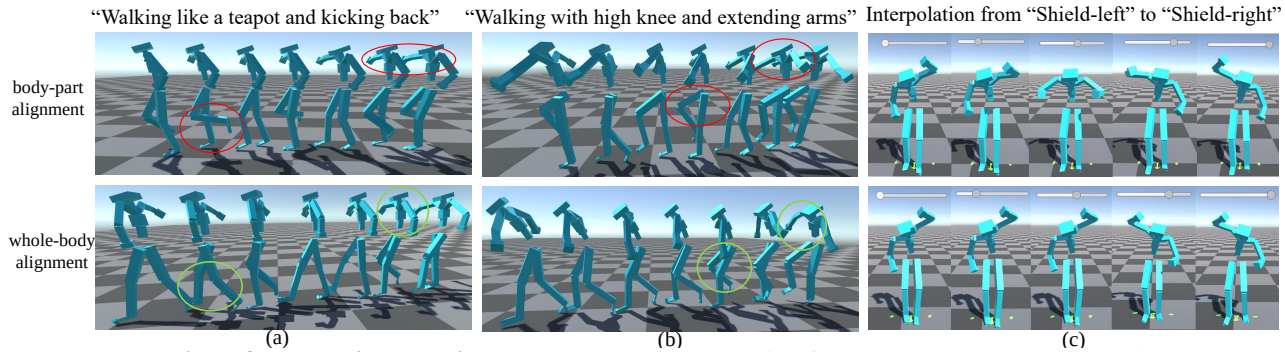


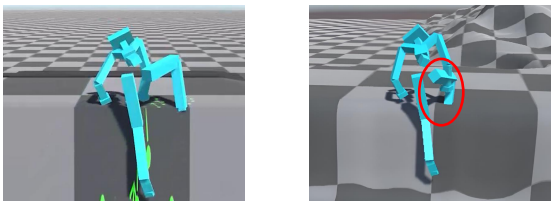
Fig. 8: **The evaluation of contrastive learning.** The comparative analysis of body-part and whole-body alignment strategies indicates that body-part contrastive learning significantly enhances the SPORT framework by: (a)-(b) enabling the generation of unseen motions, and (c) facilitating precise linear interpolation for smooth transitions between divergent styles.

3) The Evaluation on MoE: In the motion encoder and content modulator, we employ a standard MoE and a recursive MoE to learn motion and content embeddings, respectively. To evaluate the efficacy of these modules, we conduct comparative experiments with reduced expert branches (from 8 to 4), denoted as SPORT-4Experts and SPORT-4ContExperts. As presented in Table I, while the original SPORT demonstrates superior performance in both motion quality and diversity indicators, SPORT-4Experts outperforms both SPORT-4ContExperts and the original SPORT in terms of runtime (46.27ms vs. 49.67ms vs. 52.09ms) and model parameter count (31.49M vs. 47.05M vs. 55.46M). Visualization analysis indicates that both reduced-expert models can generate the majority of style actions present in the 100STYLE, with limitations primarily in dynamic body part styles. Specifically, SPORT-4Experts exhibits constraints in rolling and flapping arm motions, while SPORT-4ContExperts demonstrates limitations exclusively in rolling arm motions. Notably, both models maintain the capability to generate unseen style motions.

4) The Evaluation on Body-part Contrastive Learning: The content encoder creates a compact and decoupled alignment space that enhances continuity by clustering semantically similar motion embeddings and facilitates seamless transitions between different prompts. We conduct two analyses to validate our approach. First, to evaluate compactness, we examine a variant without the content encoder, where CLIP embeddings for six body parts are directly utilized as the content embedding. While this approach maintains decoupling capabilities and enabled body part-level motion editing, it increases the content embedding dimension from 384 to 3072, resulting in a significant expansion of model parameters compared to SPORT (66.64M vs. 55.46M in Tab. I). Second, to assess decoupling effectiveness, we implement a full-body contrastive learning strategy following [46], where GPT-3 generated full-body motion text prompts for each style. This approach requires modifications to both text and motion sequence encoders to process full-body motion. Experimental results in Fig. 8 indicate that body-part contrastive learning effectively constructs a decoupled text-motion alignment space with two principal capabilities: (1) synthesizing unseen motions through local prompt amalgamation, and (2) executing precise linear interpolation facilitating smooth transitions between different types of motions. Conversely, the whole-body alignment paradigm demonstrates limitations in zero-shot stylized motion generation and seamless inter-type transitions.

C. Comparative Performance

To evaluate SPORT's effectiveness in addressing the semantic gap caused by abstract input prompt, we first benchmark our approach against SOTA real-time style transfer methods, including styleERD [5], Mason et al. [4], [6] and MotionPuzzle [50]. A quantitative evaluation is conducted using three types of metrics: (1) motion quality, including FMD, foot skating and seamless transitions between different styles; (2)



(a) (b)

Fig. 9: **Visual comparison with PFNN.** When executing the wall climbing action, the animation generated by SPORT (a) seems more plausible than the one produced by PFNN (b).

TABLE II: Likert scale markers to assess the rationality

Extremely Unsatisfied	Physically implausible poses
Slightly Unsatisfied	Clipping/foot-sliding/jitter
Mediocre	Reasonable yet uninspiring results
Satisfied	Natural character-terrain interactions
Highly Satisfied	Vivid character-terrain interactions

generalization capabilities, encompassing motion diversity, the breadth of style representation and multi-terrain adaptability; (3) model complexity, considering parametric complexity and runtime efficiency.

Results in Tab. I validate the superior performance of our approach in both quality and generalization when compared to all baseline methods. StyleERD and the work of [6] surpass our results in terms of model parameters and runtime respectively. While it is worth noting that styleERD is limited to handling only 7 periodic styles and [6] does not extend to accommodate unseen styles. Additionally, Motion Puzzle can achieve zero-shot stylized generation with fewer model parameters. However, the supplementary video reveals that the motion generated by Motion Puzzle exhibits unnatural movements like foot-sliding and lacks seamless transition between different styles. Despite our method being more time-consuming and leveraging a larger number of model parameters compared to other methods, it adeptly captures decoupled content features at body part level, thereby composing high-quality zero-shot stylized motions. Moreover, considering our earlier discussion in Sec. IV-B2, where we emphasized that our model can be readily deployed on a laptop equipped with an RTX 2060 graphics card and run in real time, we believe the trade-off between performance and model complexity is justified.

Furthermore, to assess SPORT's effectiveness in addressing the semantic gap caused by rapidly changing terrains, we conducted a comparative analysis with PFNN, focusing on motion performance over complex terrains. While PFNN slightly outperforms our model in terms of foot skating, our method surpasses PFNN in terms of FMD and diversity. This distinction becomes more apparent in Fig. 9, where our model can generate more reasonable animations during interactions with varying terrains. Additionally, in Sec. IV-D, we perform a user study to perceptually evaluate their motion quality.

D. User Study

We recruited a group of 20 participants, with 14 of them being male, from Shanghai Jiao Tong University to participate in our experiment. They were unaware to the purposes of the experiment. The average age was 25.5 (SD=2.4). The

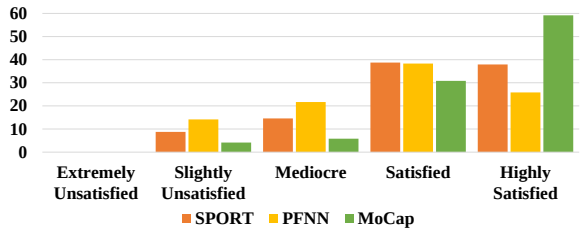


Fig. 10: **Responses on the rationality of motions over complex terrains.** The evaluation compared motions across different terrains with those of PFNN and MoCap.

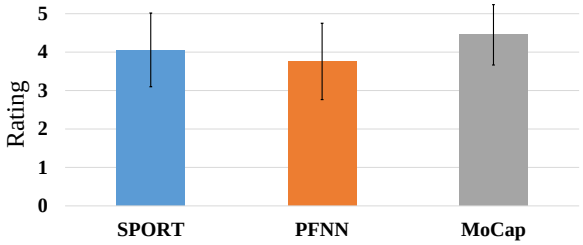


Fig. 11: **Mean ratings of the user study.** Mean ratings with standard deviation bars are calculated to assess the performance of models.

experiment was approved by Shanghai Jiao Tong University Research Ethics Committee.

We measure motion rationality over complex terrains with a trial that followed the within-subject design. In this trial, participants were asked to rate three sets of twelve animations, each set comprising two animations for six motions: hurdling, balance beam traversal, ascending stairs, platform jumps, hill climbing, and wall scaling. Three different methods were employed in generating these animations. Participants who rated the MoCap clips poorly (scores$\leq$ 2) were asked to test again. The presentation order of animations war randomized. Evaluations were based on a five-point Likert sacle in Tab. II.

The results are visualized in Fig. 10. Feedback collected for SPORT indicated a strong preference for high ratings ("Satisfied":38.75%,"Highly Satisfied":37.92%), surpassing the scores of PFNN ("Satisfied":38.33%,"Highly Satisfied":25.83%). Participants observed that animations produced by PFNN often faced issues with clipping, particularly noticeable when characters climbed walls or slopes. Specifically, the characters' legs occasionally penetrated the model in such scenarios, and their feet exhibited unrealistic heights during climbing actions. On the other hand, SPORT received a lower percentage of "highly satisfied" responses (37.92%), compared to MoCap clips (59.17%). Participants noted that while our generated animations were acceptable, they exhibited a more cyclical pattern and lacked nuanced expressions.

Fig. 11 illustrates meaning ratings of the user study. SPORT received a higher rating compared to PFNN. Notably, there was an obvious disparity between the outcomes of FlowSMM and MoCap. To assess the statistical significance of these differences, we conducted a series of statistical tests. First, we performed a one-way repeated measures ANOVA, followed by a post-hoc Tukey HSD test. Prior to these operations, we conducted Kruskal-Wallis and Bartlett's tests to confirm

the normality and homoscedasticity assumptions of ANOVA. Additionally, we examined the boxplot, which did not detect any data outliers. The results of ANOVA show significant main effects among these models $F(2,237)=231.60$, $p<0.001$). Then Tukey's HSD test indicated that all differences between the models were statistically significant ($p<0.001$).

V. Conclusion

This paper introduces SPORT, a real-time motion generation framework designed to generate ever-changing motions using composite prompts and terrain geometries. This framework enables users to input extensive text prompts and interactively control the creation of stylized animations through a gamepad in an autoregressive manner. By incorporating BP-PAE, SPORT enhances the learning of local periodicity, which improves seamless transitions between different motion types. Furthermore, the application of body-part contrastive learning enables the model to capture fundamental motion features, bridging the semantic gap between texts and motions. The use of a diffusion model further improves the generation of probabilistic and realistic motions. Notably, SPORT circumvents the challenges associated with sequence processing and eliminates the need for large-scale cross-modal datasets. A prototype of SPORT has been implemented in Unity, showcasing its robustness in handling zero-shot style prompts and complex terrains. Through both qualitative and quantitative comparisons, we demonstrate that SPORT outperforms current SOTA methods in terms of quality, generalization and inference speed.

The synchronous integration of motion equilibrium maintenance and style transfer in the multi-terrain scenes presents a fundamental challenge. Consequently, SPORT adopts a conservative approach. It only seeks to demonstrate that, in cases where text prompts fail to consistently respond to varying terrains, terrain-specific actions can still be performed by incorporating terrain geometries. In the future, we plan to explore the use of large-scale motion-language models, such as MotionGPT [53] and advanced multimodal architectures like GPT4 to develop a controller capable of comprehensively evaluating both the input task prompt and terrain signals, thereby offering more precise control instructions. Furthermore, most existing datasets focus on either style or terrain individually. To bridge this disparity, we aim to collect a dataset that incorporates both style and terrain aspects simultaneously.

Acknowledgments

This work is supported by National Natural Science Foundation of China (NSFC, NO. 62472285), Shanghai Municipal Science and Technology Major Project (No. 2021SHZDZX0102), and the Open Project Program of the State Key Laboratory of CAD&CG (Grant No. A2305) Zhejiang University.

References

[1] D. Holden, T. Komura, and J. Saito, "Phase-functioned neural networks for character control," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–13, 2017.

[2] H. Zhang, S. Starke, T. Komura, and J. Saito, "Mode-adaptive neural networks for quadruped motion control," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–11, 2018.

[3] H. Y. Ling, F. Zinno, G. Cheng, and M. Van De Panne, "Character controllers using motion vaes," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 40–1, 2020.

[4] I. Mason, S. Starke, H. Zhang, H. Bilen, and T. Komura, "Few-shot learning of homogeneous human locomotion styles," in *Computer Graphics Forum*, vol. 37, no. 7. Wiley Online Library, 2018, pp. 143–153.

[5] T. Tao, X. Zhan, Z. Chen, and M. van de Panne, "Style-erd: Responsive and coherent online motion style transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6593–6603.

[6] I. Mason, S. Starke, and T. Komura, "Real-time style modelling of human locomotion via feature-wise transformations and local motion phases," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 5, no. 1, pp. 1–18, 2022.

[7] S. Starke, Y. Zhao, T. Komura, and K. Zaman, "Local motion phases for learning multi-contact character movements," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 54–1, 2020.

[8] 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*. Springer, 2022, pp. 358–374.

[9] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-or, and A. H. Bermano, "Human motion diffusion model," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=SJ1kSyO2jwu

[10] 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*, 2023, pp. 18 000–18 010.

[11] 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.

[12] 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*. PMLR, 2021, pp. 8748–8763.

[13] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[14] T. Ao, Z. Zhang, and L. Liu, "Gesturediffuclip: Gesture diffusion model with clip latents," *ACM Trans. Graph.*

[15] M. Petrovich, M. J. Black, and G. Varol, "TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis," in *International Conference on Computer Vision (ICCV)*, 2023.

[16] S. Lu, L.-H. Chen, A. Zeng, J. Lin, R. Zhang, L. Zhang, and H.-Y. Shum, "Humantomato: Text-aligned whole-body motion generation," *arxiv:2310.12978*, 2023.

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

[18] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[19] B. Han, H. Peng, M. Dong, C. Xu, Y. Ren, Y. Shen, and Y. Li, "Amd autoregressive motion diffusion," *arXiv preprint arXiv:2305.09381*, 2023.

[20] S. Starke, H. Zhang, T. Komura, and J. Saito, "Neural state machine for character-scene interactions." *ACM Trans. Graph.*, vol. 38, no. 6, pp. 209–1, 2019.

[21] S. Starke, I. Mason, and T. Komura, "Deepphase: periodic autoencoders for learning motion phase manifolds," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–13, 2022.

[22] F. Bao, C. Li, J. Zhu, and B. Zhang, "Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models," *arXiv preprint arXiv:2201.06503*, 2022.

[23] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[24] 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)*, vol. 39, no. 4, pp. 64–1, 2020.

[25] 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*, vol. 27, no. 1, pp. 14–28, 2019.

[26] 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*, vol. 32, 2019.

[27] 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)*, vol. 41, no. 4, p. 138, 2022.

[28] S. Alexanderson, G. E. Henter, T. Kucherenko, and J. Beskow, "Style-controllable speech-driven gesture synthesis using normalising flows," in *Computer Graphics Forum*, vol. 39, no. 2. Wiley Online Library, 2020, pp. 487–496.

[29] B. Ji, Y. Pan, Y. Yan, R. Chen, and X. Yang, "Stylevr: Stylizing character animations with normalizing flows," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[30] D. Greenwood, S. Laycock, and I. Matthews, "Predicting head pose from speech with a conditional variational autoencoder," ser. INTERSPEECH'17. Grenoble, France: ISCA, 2017, pp. 3991–3995.

[31] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Two distributed-state models for generating high-dimensional time series." *Journal of Machine Learning Research*, vol. 12, no. 3, 2011.

[32] D. Greenwood, S. Laycock, and I. Matthews, "Predicting head pose in dyadic conversation," ser. IVA'17. Cham, Switzerland: Springer, 2017, pp. 160–169.

[33] Z. Zhang, R. Liu, K. Aberman, and R. Hanocka, "Tedi: Temporally-entangled diffusion for long-term motion synthesis," *arXiv preprint arXiv:2307.15042*, 2023.

[34] W. Zhou, Z. Dou, Z. Cao, Z. Liao, J. Wang, W. Wang, Y. Liu, T. Komura, W. Wang, and L. Liu, "Emdm: Efficient motion diffusion model for fast, high-quality motion generation," *arXiv preprint arXiv:2312.02256*, 2023.

[35] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5775–5787, 2022.

[36] K. Amaya, A. Bruderlin, and T. Calvert, "Emotion from motion," in *Graphics interface*, vol. 96. Toronto, Canada, 1996, pp. 222–229.

[37] M. E. Yumer and N. J. Mitra, "Spectral style transfer for human motion between independent actions," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–8, 2016.

[38] E. Hsu, K. Pulli, and J. Popović, "Style translation for human motion," in *ACM SIGGRAPH 2005 Papers*, 2005, pp. 1082–1089.

[39] 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*, 2009, pp. 1025–1032.

[40] S. Xia, C. Wang, J. Chai, and J. Hodgins, "Realtime style transfer for unlabeled heterogeneous human motion," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 1–10, 2015.

[41] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–11, 2016.

[42] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[43] H. Du, E. Herrmann, J. Sprenger, N. Cheema, S. Hosseini, K. Fischer, and P. Slusallek, "Stylistic locomotion modeling with conditional variational autoencoder." in *Eurographics (Short Papers)*, 2019, pp. 9–12.

[44] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[45] M. Petrovich, M. J. Black, and G. Varol, "TEMOS: Generating diverse human motions from textual descriptions," in *European Conference on Computer Vision (ECCV)*, 2022.

[46] N. Athanasiou, M. Petrovich, M. J. Black, and G. Varol, "SINC: Spatial composition of 3D human motions for simultaneous action generation," in *ICCV*, 2023.

[47] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[48] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[50] D.-K. Jang, S. Park, and S.-H. Lee, "Motion puzzle: Arbitrary motion style transfer by body part," *ACM Transactions on Graphics*, vol. 41, no. 3, pp. 1–16, jun 2022. [Online]. Available: https://doi.org/10.1145%2F3516429

[51] X. Tang, L. Wu, H. Wang, B. Hu, X. Gong, Y. Liao, S. Li, Q. Kou, and X. Jin, "Rsmt: Real-time stylized motion transition for characters," *arXiv preprint arXiv:2306.11970*, 2023.

[52] C. Guo, X. Zuo, S. Wang, S. Zou, Q. Sun, A. Deng, M. Gong, and L. Cheng, "Action2motion: Conditioned generation of 3d human motions," in *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*, 2020.

[53] B. Jiang, X. Chen, W. Liu, J. Yu, G. Yu, and T. Chen, "Motiongpt: Human motion as a foreign language," *arXiv preprint arXiv:2306.14795*, 2023.

Bin Ji is a member of Character Lab, Shanghai Jiao Tong University, Shanghai, China. He received the B.S. degree in mechanical engineering from Xidian University and the M.S. degree in Computer Science and Engineering from Shanghai Jiao Tong University. He is currently pursuing the Ph.D. degree in Computer Science and Engineering at Shanghai Jiao Tong University. His research interest includes computer graphics and computer vision.

Ye Pan is currently an Associate Professor with Shanghai Jiao Tong University. Her research interests include AR/VR, 3D animations, and computer graphics. Previously, she was an Associate Research Scientist at Disney Research Los Angeles. She received the B.Sc. degree from Purdue/UESTC in 2010 and the Ph.D. degree in computer graphics from the University College London (UCL) in 2015. She has served as Associate Editor of the International Journal of Human Computer Studies, and a regular member of IEEE VR TPC.

Zhimeng Liu is a member of Character Lab, Shanghai Jiao Tong University, Shanghai, China. He is currently pursuing the B.S. degree in French and Information Engineering at Shanghai Jiao Tong University. His research interest includes computer graphics and virtual reality.

Shuai Tan is a member of Character Lab, Shanghai Jiao Tong University, Shanghai, China. He received the B.S. degree in software engineering from Sichuan University. He is currently pursuing the Ph.D. degree in Computer Science and Engineering at Shanghai Jiao Tong University. His research interest includes computer vision and multi-modal learning.

Xiaokang Yang (Fellow, IEEE) received the B.S. degree from Xiamen University, Xiamen, China, in 1994, the M.S. degree from the Chinese Academy of Sciences, Beijing, China, in 1997, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2000. He is currently a Distinguished Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. His research interests include visual signal processing and communication, media analysis and retrieval, and pattern recognition. Dr. Yang serves as an Associate Editor of the IEEE Transactions on Multimedia and the IEEE Signal Processing Letters.