SOSControl: Enhancing Human Motion Generation Through Saliency-Aware Symbolic Orientation and Timing Control

Ho Yin Au, Junkun Jiang, Jie Chen[†]

Department of Computer Science, Hong Kong Baptist University
{cshyau, csjkjiang, chenjie}@comp.hkbu.edu.hk

Abstract

Traditional text-to-motion frameworks often lack precise control, and existing approaches based on joint keyframe locations provide only positional guidance, making it challenging and unintuitive to specify body part orientations and motion timing. To address these limitations, we introduce the Salient Orientation Symbolic (SOS) script, a programmable symbolic framework for specifying body part orientations and motion timing at keyframes. We further propose an automatic SOS extraction pipeline that employs temporally-constrained agglomerative clustering for frame saliency detection and a Saliency-based Masking Scheme (SMS) to generate sparse, interpretable SOS scripts directly from motion data. Moreover, we present the SOSControl framework, which treats the available orientation symbols in the sparse SOS script as salient and prioritizes satisfying these constraints during motion generation. By incorporating SMS-based data augmentation and gradient-based iterative optimization, the framework enhances alignment with user-specified constraints. Additionally, it employs a ControlNet-based ACTOR-PAE Decoder to ensure smooth and natural motion outputs. Extensive experiments demonstrate that the SOS extraction pipeline generates human-interpretable scripts with symbolic annotations at salient keyframes, while the SOSControl framework outperforms existing baselines in motion quality, controllability, and generalizability with respect to motion timing and body part orientation control.

Code — https://github.com/asdryau/SOSControl

Introduction

Text-conditioned human motion generation has received substantial research focus due to its potential to produce diverse humanoid motions guided by text prompts, with promising applications in media content creation, robotics, and human-AI collaboration. However, since text descriptions are often subjective and ambiguous, traditional text-to-motion frameworks lack precise control over body part orientation and timing, prompting the integration of additional conditioning signals to enhance motion controllability.

Recent research has explored the use of joint keyframe locations to enhance controllability in motion generation. However, this approach often offers limited control over body part orientation and timing, and accurately defining plausible keyframe locations remains complex. For example, specifying only the final fist location for a *squatted forward punch* may produce incorrect arm orientation, such as an unintended *lower punch*, if the system adjusts primarily through shoulder rotation rather than coordinating the entire body. Also, the model may misinterpret end locations as intermediate waypoints, resulting in overshooting and disrupting the intended punch timing. Moreover, ensuring that specified 3D joint locations are accurately placed and physically executable requires extensive manual adjustments in animation tools, such as frequent switching between camera views, and a thorough understanding of motion dynamics (e.g., adjusting for appropriate movement speed and physical balance), making the workflow time-consuming and impractical for industrial animation pipelines.

To tackle these challenges, we introduce the Salient Orientation Symbolic (SOS) script, a programmable symbolic framework designed to define and represent body part orientations and motion timing within motion sequences. Inspired by Labanotation (Guest 2013), the SOS script uses orientation symbols to annotate individual body parts at high saliency keyframes. As shown in Fig. 1, the SOS script is represented as a symbolic staff, offering an intuitive, programmable interface for adjusting orientation and timing through drag-and-drop symbol placement. Additionally, we propose an automatic extraction pipeline that generates SOS scripts directly from motion sequences, utilizing temporally-constrained agglomerative clustering to extract frame saliency. Leveraging this saliency information, our Saliency-based Masking Scheme (SMS) adaptively filters the extracted orientation features below user-defined saliency thresholds, highlighting key motion moments and producing a sparse, human-interpretable SOS script. Users can adjust these thresholds at test time to interactively control the SOS script's sparsity for customizable visualization.

Building on this symbolic representation, we introduce SOSControl, which integrates the SOS script with language-guided motion generation. Since SOS scripts emphasize retaining high-saliency keyframes, the SOSControl framework considers the available orientation symbols in the input sparse SOS script as salient and prioritizes satisfying these constraints during the motion generation process. Extend-

[†]Corresponding Author

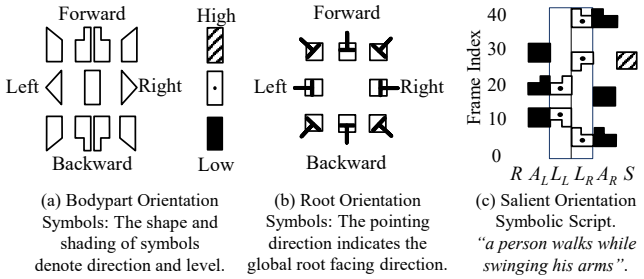
ing ControlNet-based motion diffusion methods, SOSControl incorporates SMS into data preprocessing and augmentation, enabling the model to prioritize retained symbols and precisely synchronize motion peaks with intended timings. The differentiable orientation feature extraction pipeline further supports gradient-based iterative optimization, ensuring precise alignment with input orientations. Additionally, the ControlNet-based ACTOR-PAE Decoder regularizes motion outputs for smoothness and naturalness, allowing stable iterative optimization during inference. To our knowledge, this is the first approach to use saliency information from agglomerative clustering for enhanced motion control. Our contributions are as follows:

- We introduce the SOS script, a programmable symbolic framework with an intuitive staff-based interface for representing body part orientations and motion timing using orientation symbols annotated at keyframes.
- We propose an automatic SOS extraction pipeline, which uses temporally-constrained agglomerative clustering to identify frame saliency, and applies a Saliency-based Masking Scheme (SMS) to adaptively filter orientation features, producing sparse and interpretable SOS scripts.
- We present the SOSControl framework, which integrates SOS scripts into motion generation using SMS-based data augmentation, gradient-based optimization, and employs a ControlNet-based ACTOR-PAE Decoder for smooth and natural motion outputs.

Related Works

Motion Diffusion with Textual Descriptions. Diffusion frameworks (Song, Meng, and Ermon 2020; Ho, Jain, and Abbeel 2020) have proven effective in generating high-quality outputs across diverse domains. In text-to-motion generation, foundational diffusion frameworks like MLD (Chen et al. 2023), MDM (Tevet et al. 2023), and MotionDiffuse (Zhang et al. 2024) offer great extensibility by enabling precise and flexible control through detailed text descriptions analysis. AttT2M (Zhong et al. 2023), FineMoGen (Zhang et al. 2023), and CoMo (Huang et al. 2024) further enhance motion control by establishing more accurate associations between text inputs and specific body parts. Meanwhile, GraphMotion (Jin et al. 2023) and Fg-T2M (Wang et al. 2023) utilize text-based semantic graphs to analyze input text comprehensively, uncovering detailed relationships between body parts and motion specifications to improve the accuracy and contextual relevance of the produced motion. However, achieving detailed motion control using freeform text remains challenging, as users must verbosely specify all body part conditions, joint locations, and motion semantics in a paragraph. Moreover, the text processing pipelines in these models may filter or misinterpret instructions, resulting in generated motions that may not fully reflect the user's intent.

Integrating Control Signals in Motion Diffusion. In addition to detailed text analysis, motion diffusion frameworks can be extended to incorporate additional control signals, such as joint keyframe locations, into the generation process. For instance, PriorMDM (Shafir et al. 2024) extends MDM by iteratively imputing user-specified root trajectories during motion sampling for trajectory-guided generation. Similarly, GMD (Karunratanakul et al. 2023) refines diffused motion by inferring the root trajectory from the generated motion, comparing it to the user-specified trajectory, and applying gradient-based optimization for refinement. Recent methods further enhance keyframe alignment through iterative optimization. For example, OmniControl (Xie et al. 2024) performs diffusion-time optimization by computing gradients at each diffusion step to improve alignment with keyframe specifications, while leveraging the ControlNet architecture to ensure motion coherence among unspecified frames and joints. In contrast, TLControl (Wan et al. 2024) trains a transformer to denoise VQ-VAE encoded body part tokens using both text and trajectory inputs, and employs test-time iterative optimization to better align the output motion with user-specified conditions.



(a) Bodypart Orientation Symbols: The shape and shading of symbols denote direction and level.

(b) Root Orientation Symbols: The pointing direction indicates the global root facing direction.

(c) Salient Orientation Symbolic Script. *"a person walks while swinging his arms".*

Figure 1: Salient Orientation Script (SOS) Illustration: (a) Body Part and (b) Root Orientation Symbols specify keyframe states for effective motion control. (c) SOS example as a staff highlights its programmable interface potential.

Abstract and Interpretable Motion Descriptors. Labanotation (Guest 2013) is a widely adopted symbolic system for recording body part movements using orientation symbols on a staff. While direct extraction of Labanotation from motion has not yet been achieved, various studies have focused on extracting related spatial features such as joint positions and orientations. For example, PoseScript (Delmas et al. 2022) and CoMo (Huang et al. 2024) convert these features into body part-level text descriptions to improve text-motion alignment. Meanwhile, KP (Liu et al. 2024), HL (Li et al. 2024), and PL (Jiang et al. 2024) transform these features into abstract, interpretable representations for motion understanding, which can also be used as control signals. However, these approaches do not address motion saliency detection, and their representations are often described in a verbose, frame-by-frame manner, making the process of programming these features highly labor-intensive.

Preliminary

Motion Data Representation

Due to the increasing popularity of using the SMPL humanoid model (Loper et al. 2015), human motions are typically standardized to the SMPL skeleton and represented as frame sequences comprising a global root trajectory and 24 joint rotations. Traditional motion generative models con-

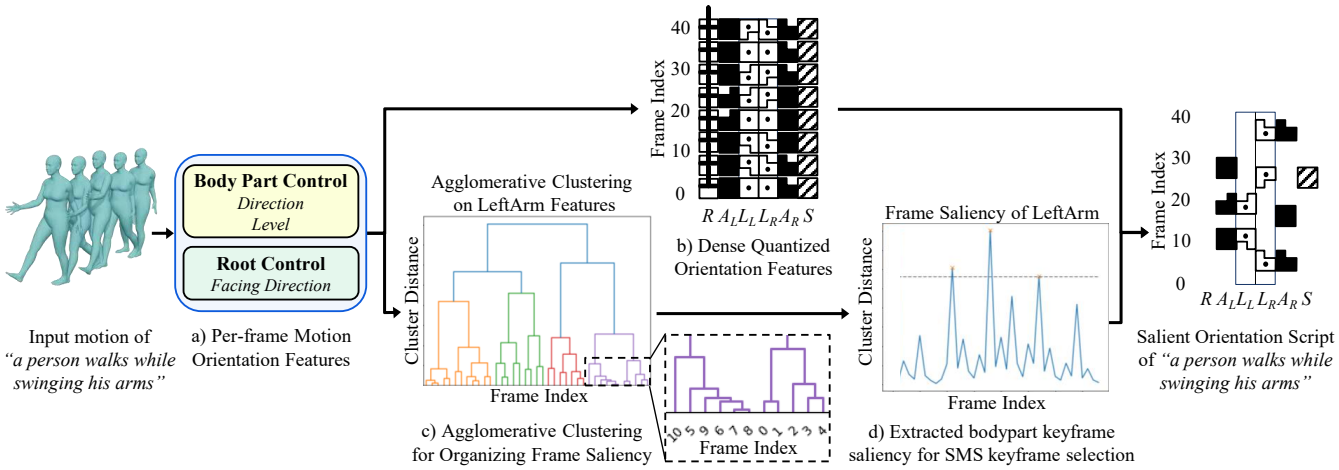


Figure 2: Overview of the SOS extraction pipeline: (a) extract per-frame orientation features, then (b) quantize them according to symbol categories from Fig. 1. Agglomerative clustering is applied in (c) to derive the per-frame orientation feature into frame saliency shown in (d). Symbols selected from frames with saliency above a threshold compose the final SOS.

vert SMPL motion data into 263 parameters based on the HumanML3D (Guo et al. 2022) pose format, which includes root-centric velocity, joint velocities, joint rotations, and foot contact information. To incorporate the missing root orientation, we add a 6D rotation (Zhou et al. 2019), resulting in a motion representation of $\mathbf{x} \in \mathbf{R}^{T\times 269}$.

Kinematic Feature Extraction

Feature extraction algorithms for body part orientation are often developed based on the orientation concept in Labanotation. For example, KP (Liu et al. 2024) extracts reference vectors $\mathbf{r}_t \in \mathbf{R}^{T\times 3\times 3}$ to define egocentric directions for each motion frame t, which are then used to compute the Pairwise Relative Position Phrase (PRPP) (Liu et al. 2024):

$$\mathbf{o}_t^J = \text{PRPP}(e^J, a^J)_t = (\mathbf{l}_t(e^J) - \mathbf{l}_t(a^J)) \cdot \mathbf{r}_t, \quad (1)$$

where $\mathbf{o}_t^J$ represents the relative position between the end joint e^J and the anchor joint a^J at frame t, serving as the orientation feature for body part J. Here, $\mathbf{l}$ denotes local joint trajectories obtained via forward kinematics with a zeroed global root trajectory. The dot product with reference vectors $\mathbf{r}_t$ transforms relative positions into egocentric directions.

Salient Orientation Symbols

Inspired by Labanotation (Guest 2013), we propose the Salient Orientation Symbolic (SOS) script as an abstract motion representation designed for motion control. Using body part frame saliency extracted through temporally-constrained agglomerative clustering, SOS is depicted as a sparse and concise symbolic staff that runs vertically up. As shown in Fig. 1(c), the staff contains six columns corresponding to body parts: *Root* (R) *Left Arm* (A_L), *Left Leg* (L_L), *Right Leg* (L_R), *Right Arm* (A_R), and *Spine* (S). Fig. 1(a) presents the eight root direction symbols, while Fig. 1(b) depicts the 26 body part orientation symbols, where shape denotes direction and shading represents level. The vertical position of each symbol on the staff reflects the orientation state of the corresponding body part at each frame.

We developed a pipeline to extract the SOS script from motion data in four main steps: First, kinematic feature extraction transforms raw motion signals into orientation features. Second, spatial feature quantization maps these features to discrete labels. Third, hierarchical temporal saliency detection analyzes orientation features for each body part, constructing a segment tree to detect frame saliency. Finally, by applying a Saliency-based Masking Scheme (SMS) to the quantized features based on the detected saliency, the SOS script can be synthesized at various levels of granularity.

Kinematic Feature Extraction

We extract per-frame motion orientation features $\mathbf{o} \in \mathbf{R}^{T\times 6\times 3}$ to represent the orientation state in the SOS script. These features are represented as three-dimensional directional vectors for six body parts, capturing the root facing direction and the body part orientation. For the *Root*, we use the horizontal facing direction $\mathbf{o}^R = \mathbf{r}_t^f \in \mathbf{R}^{T\times 1\times 3}$, obtained by projecting the forward reference vector in $\mathbf{r}_t$ onto the ground plane. For the remaining five body parts, such as *Left Arm*, the orientation $\mathbf{o}^{A_L} \in \mathbf{R}^{T\times 1\times 3}$ is computed using the PRPP between the end joint (e.g., *left wrist* e^{A_L}) and the anchor joint (e.g., *left shoulder* a^{A_L}) for each body part.

Hierarchical Temporal Saliency Detection

Hierarchical temporal saliency detection organizes the orientation features $\mathbf{o}$ to extract body part keyframes, resulting in a sparse and easily manipulable representation. We apply temporally-constrained agglomerative clustering to group these features into a bottom-up tree of connected segments. For each body part, we calculate the frame-level central finite difference of the normalized dot product between $\mathbf{o}$ with 26 unit-norm direction vectors, using this as input to the *scikit-learn* agglomerative clustering algorithm. Following the approach in Librosa (McFee et al. 2015), we con-

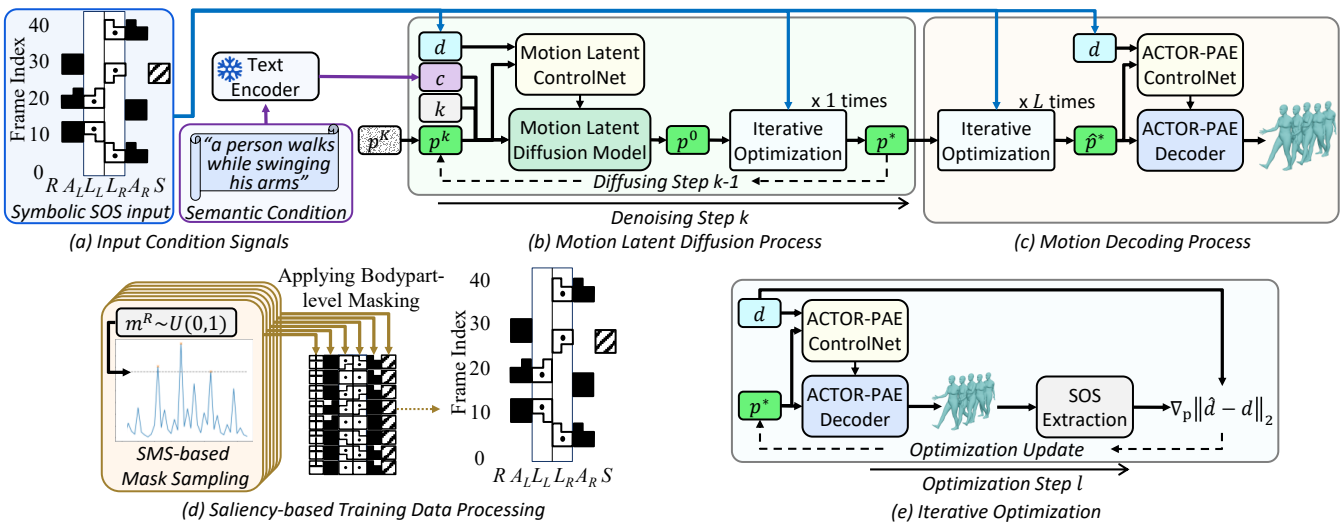


Figure 3: Overview of the SOSControl pipeline: (a) Start by obtaining a semantic condition and a Symbolic Orientation Script (SOS) from user input, (b) perform motion periodic latent diffusion, and (c) decode the resulting motion latent back to motion. Both (b) and (c) utilize ControlNet to incorporate the SOS condition into the trained models and apply iterative optimization in (e) to refine the model output, ensuring better alignment with the input SOS conditions. To improve model adaptability to diverse user-provided SOS scripts, (d) perform SMS-based mask sampling to generate SOS at varying levels of granularity, enabling the trained models to handle different saliency variations more robustly.

strain the connectivity matrix of the clustering algorithm so that merging occurs only between adjacent segments, ensuring that clusters correspond to temporally contiguous motion segments. Fig. 2(c) shows a dendrogram of this bottom-up segmentation, where the y-axis indicates cluster distance and the x-axis indicates frame indices.

After constructing the segmentation tree, each node comprises two temporally adjacent segment subnodes and a merging distance that indicates the value at which the segments are joined. We compute the body part keyframe saliency signal by assigning the merging distance at each node as the saliency value to the first frame of the subsequent segment subnode. Figure 2(d) shows the resulting saliency signal, obtained by processing all nodes in a bottom-up manner within the hierarchical tree. For instance, the saliency value at frame 18 for the *Left Arm* is the highest, indicating a key moment when the arm reaches its peak during a swinging motion. Further details on saliency value extraction are provided in the Supplementary Material.

Spatial Feature Quantization

The spatial feature quantization process transforms $\mathbf{o}$ into discrete intervals, enabling symbolic representation as shown in Fig. 2(b). Based on the Labanotation orientation scheme in Fig. 1(a), we define 26 unit-norm direction vectors $\mathbf{u} \in \mathbb{R}^{26\times 3}$, each representing a specific orientations (e.g. $(1, 0, 0)$ for the Right-Middle symbol). To increase sensitivity to horizontal movement compared to vertical movement, we increase the weight on the upward axis (e.g. $(0, \frac{1}{\sqrt{10}}, \frac{3}{\sqrt{10}})$ for the Forward-Top symbol). To quantize the orientation features into symbols, we compute the dot product between the orientation vector $\mathbf{o}$ and each of the 26 direction vectors in $\mathbf{u}$:

$$\mathbf{q} = softmax(\frac{\mathbf{o}}{||\mathbf{o}||} \cdot \mathbf{u}^T) \cdot \mathbf{u}, \tag{2}$$

where the softmax operation ensures the quantization operation remains differentiable, allowing for iterative optimization during diffusion-time and test-time. For symbol recognition and visualization, the softmax can be replaced with an argmax operation. Finally, by selecting symbols from the quantized orientation features at keyframes identified by temporal saliency detection, we synthesize the final SOS staff, as shown on the right in Fig. 2.

Saliency-Based Masking Scheme

Based on the body part keyframe saliency extracted from agglomerative clustering, we apply a Saliency-based Masking Scheme (SMS) that selects body part keyframes by masking all the body part frames with saliency values below a specified threshold. For instance, setting the threshold to 0.7 of the maximum saliency across all body parts results in the selection of three LeftArm keyframes, as shown in Fig. 2(d). This enables customizable levels of detail in the SOS script, supporting analytical visualization and data augmentation.

Motion Periodic Latent Diffusion With Saliency Orientation Script

To facilitate motion generation from the SOS script, we propose a two-stage framework, as shown in Fig. 3. In the first stage, the motion periodic latent $\mathbf{p}$ is denoised using the input text and SOS script. In the second stage, the denoised

latent is decoded back into human motion. To enhance alignment between the decoded motion and the input SOS, we integrate ControlNet and iterative optimization into both motion diffusion and decoding processes. In the following sections, we first introduce the basic framework for periodic latent diffusion. We then describe the ControlNet adaptation and iterative optimization strategies used to align the generated motion with the input SOS script. Finally, we present a saliency-based training data processing approach, which enables the model to accommodate user-provided SOS scripts with varying levels of granularity.

Basic Periodic Latent Diffusion

We perform motion latent diffusion in a periodic latent space defined by the ACTOR-PAE, which promotes the generation of smooth, natural motions and provides a regularized foundation for iterative optimization during motion decoding.

ACTOR-PAE and Periodic Latent Space. Following the approach in (Au et al. 2025), we construct the ACTOR-PAE model by integrating the periodic parameterization from PAE (Starke, Mason, and Komura 2022) with the motion autoencoder architecture from ACTOR (Petrovich, Black, and Varol 2021). Specifically, the ACTOR-PAE encoder $\mathcal{P}_E$ processes motion $\mathbf{x}$ into four phase parameters $\mathbf{f}, \mathbf{a}, \mathbf{b}, \mathbf{s} \in \mathbb{R}^P$, which are then used to generate a periodic signal $\mathbf{p} \in \mathbb{R}^{T \times P}$:

$$\mathbf{p} = \mathbf{a} \sin(\mathbf{f} \cdot (N - \mathbf{s})) + \mathbf{b}, \tag{3}$$

where $N \in \mathbb{R}^T$ denotes the time difference of each frame relative to the center of the motion sequence. The ACTOR-PAE decoder $\mathcal{P}_D$ then utilizes $\mathbf{p}$ to reconstruct the motion $\hat{\mathbf{x}}$. The model is trained using mean squared error (MSE) loss.

Periodic Latent Diffusion. Following MDM (Tevet et al. 2023), we develop the Motion Latent Diffusion Model $\mathcal{D}^-$, which adopts a transformer encoder architecture to denoise the periodic latent $\mathbf{p}^k$ at diffusion step k, conditioned on the input text $\mathbf{c}$. The training losses for $\mathcal{D}^-$ is as follows:

$$\mathcal{L}_{\mathcal{D}^-} = ||\mathbf{p}^0 - \mathcal{D}^-(k, \mathbf{c}, \mathbf{p}^k)||_2. \tag{4}$$

During inference, the model predicts the clean latent $\mathbf{p}^0$ from the diffused latent $\mathbf{p}^k$ at each diffusion step k. DDIM-Scheduler (Song, Meng, and Ermon 2020) is employed to diffuse $\mathbf{p}^0$ back to $\mathbf{p}^{k-1}$ for the next diffusion step. After K diffusion iterations, the final clean latent $\mathbf{p}^0$ is decoded into motion $\hat{\mathbf{x}}$ using the ACTOR-PAE decoder $\mathcal{P}_D$.

Injecting SOS Guidance to the Diffusion Process

In the basic periodic latent diffusion framework, $\mathcal{D}^-$ serves only as a foundational model and does not condition on the SOS input $\mathbf{d}$. To address this, we incorporate ControlNet and iterative optimization during both diffusion-time (Fig. 3(b)) and test-time (Fig. 3(c)), enabling the motion generation process to be guided and aligned with the input SOS scripts.

ControlNet Adaptation to the Diffusion Model. After training $\mathcal{D}^-$, we adapt the model to incorporate the SOS input $\mathbf{d}$ following the approach described in ControlNet (Zhang, Rao, and Agrawala 2023) and OmniControl (Xie et al. 2024). This yields $\mathcal{D}^+$, which consists of both the frozen original Motion Latent Diffusion Model $\mathcal{D}^-$ and a trainable copy of $\mathcal{D}^-$, referred to as the Motion Latent ControlNet. The trainable component is then finetuned to convert $\mathbf{d}$ into a guidance signal for the diffusion process. The training losses for $\mathcal{D}^+$ is as follows:

$$\mathcal{L}_{\mathcal{D}^+} = ||\mathbf{p}^0 - \mathcal{D}^+(k, \mathbf{c}, \mathbf{d}, \mathbf{p}^k)||_2. \tag{5}$$

In addition to the Diffusion Model $\mathcal{D}^-$, ACTOR-PAE Decoder $\mathcal{P}_D$ also employs the transformer encoder architecture, enabling the application of the same ControlNet adaptation to obtain $\mathcal{P}_D^+$. By leveraging the guidance signal derived from the input SOS script $\mathbf{d}$ during motion decoding, the decoded motion becomes better aligned with the input. This enhanced alignment also leads to improved performance in the iterative optimization process.

Iterative Optimization The denoised periodic signal $\mathbf{p}$ is decoded by $\mathcal{P}_D^+$ to produce the output motion $\hat{\mathbf{x}} = \mathcal{P}_D^+(\mathbf{p}^*, \mathbf{d}))$. The quantized orientation features $\hat{\mathbf{q}}$ are then estimated from $\hat{\mathbf{x}}$ using the extraction pipeline illustrated in Fig. 2. By comparing the orientation features of the output motion and the input SOS, the difference can be minimized through iterative optimization using gradient descent. Specifically, each gradient descent step is given by:

$$\mathbf{p}^* = \mathbf{p}^* - \nabla_{\mathbf{P}} ||\mathcal{M}_{\mathbf{d}}(\hat{\mathbf{q}}) - \mathbf{d}||_2. \tag{6}$$

Here, $\mathcal{M}_{\mathbf{d}}$ denotes the SMS mask of the input SOS script $\mathbf{d}$, ensuring gradient update to the visible salient regions only.

Saliency-Based Training Data Augmentation

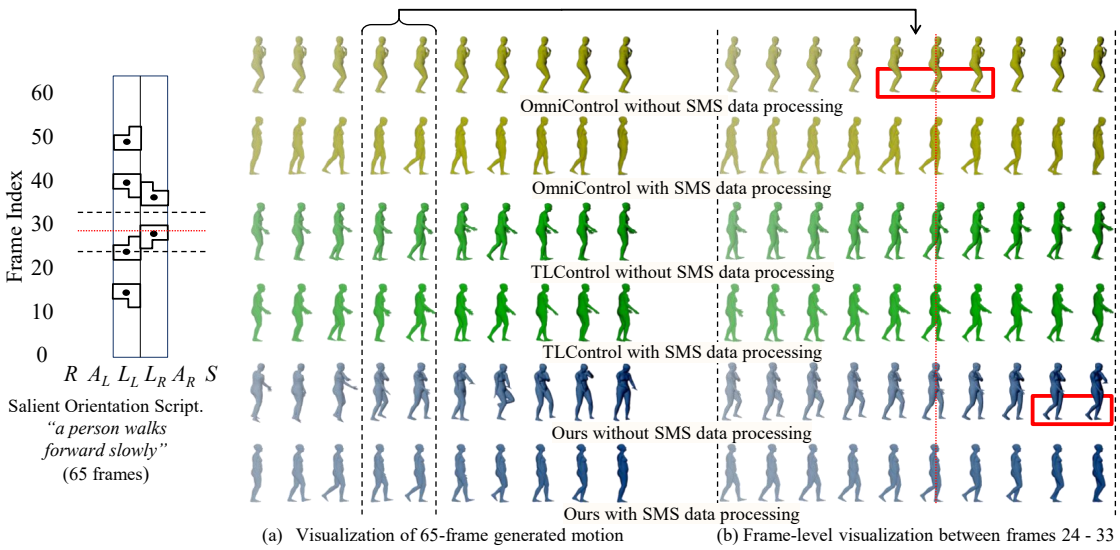
Following the SOS extraction pipeline in Fig. 2, synthesizing the final SOS scripts requires selecting an appropriate saliency threshold for each body part. However, since user-provided SOS scripts can vary in granularity, training with a fixed threshold is suboptimal. To address this, we use SMS-based mask sampling to generate scripts at multiple levels of granularity. For each body part (e.g. *Root (R)*), we uniformly sample a saliency percentile, $m^R \sim \mathcal{U}(0, 1)$, as the threshold for SMS-based SOS script synthesis. This data augmentation introduces diverse saliency variations, enabling the model to generalize to various body part and symbol combinations in user-provided SOS scripts and improving motion generation under sparse symbolic control.

Experiments

Datasets and Evaluation Metrics

We utilize the HumanML3D (Guo et al. 2022) dataset for training and evaluation. Following the data processing methods from related works (Tevet et al. 2023; Karunratanakul et al. 2023; Xie et al. 2024), subsequence lengths are set between 40 and 196 frames. All models are trained on the same dataset and representation to ensure a fair comparison.

For evaluation, we use *Fréchet Inception Distance* (*FID*) and *Multimodal Distance* (*MMD*) from the T2M (Guo et al. 2022) evaluation protocol to assess motion quality and text-motion alignment. To measure control signal alignment, we report SOS accuracy (*SOS-Acc*) by comparing generated motions to the target SOS input. We also measure *L2 loss in 6D rotation* (Zhou et al. 2019) (*L2-Rot6D*) between the generated and source motions used for the SOS input.



(a) Visualization of 65-frame generated motion (b) Frame-level visualization between frames 24 - 33

Figure 4: Visualization of SOS-conditioned motion generation results with and without SMS training data processing. (a) presents the motion sequences from left to right, while (b) provides a detailed view of the range between frames 24 and 33.

Comparison Models

We compare the performance of our method with trajectory-conditioned motion generation models described in the Related Works Section, including GMD, OmniControl, and TLControl, as well as the baseline MDM model, which is conditioned solely on text. Note that GMD 1-stage aligns motion directly to the input SOS script via direct diffusion-time optimization, whereas GMD 2-stage first generates a per-frame quantized orientation script before applying diffusion-time optimization. To ensure a fair comparison, we adapt these trajectory-conditioned models to use the SOS data representation and adjust their ControlNet adaptation and iterative optimization settings as needed. Further details are provided in the supplementary material.

Motion Generation With Saliency Control

The objective of this experiment is to generate motion conditioned on SOS. In the experimental setup, SOS is extracted from each motion in the test dataset using a saliency threshold of 0.9, and both the SOS and the corresponding textual conditions are provided as inputs to all evaluated methods.

The experimental results, summarized in Tab. 1, demonstrate that our method outperforms others in terms of control signal alignment, as evidenced by superior *SOS-Acc* scores. This highlights our method's effectiveness in leveraging information from SOS, resulting in generated motions that closely resemble the source motions, as indicated by *L2-Rot6D*. In contrast, the baseline MDM obtains the best *FID* and *MMD* scores, as it generates motions unconditionally and is not required to adapt outputs based on SOS. Additionally, Fig. 4 shows that the generated motions are well-aligned with the body part orientations specified in the input SOS while maintaining natural movement.

We further assess the impact of saliency in the SOS script during model training. Specifically, we replace the SMS-based training data processing with random masking at a ratio of 0.8 on masking quantized orientation features (w/o SMS data proc.). As shown in Tab. 1, omitting SMS results in degraded *L2-Rot6D*, *FID*, and *MMD* scores. Additionally, OmniControl, which does not utilize test-time iterative optimization, achieves lower *SOS-Acc* compared to TLControl and our method, both of which employ this optimization. As illustrated in Fig.4, when the SOS script specifies that the *Right Leg* should swing back and reach its peak at frame 29, both OmniControl and our method without SMS data processing fail to achieve the movement peak at the designated frame, treating the orientation symbol as an intermediate waypoint instead. These results highlight the importance of SMS for accurately aligning motion with salient events.

	SOS-Acc ↑	L2-Rot6D ↓	FID ↓	MMD ↓
Baseline MDM (2023)	0.151	0.351	**2.592**	**6.001**
GMD 1-stage (2023)	0.113	0.427	25.669	7.835
GMD 2-stage (2023)	0.120	0.402	21.278	7.823
OmniControl (2024)	0.873	**0.325**	3.975	6.095
- w/o SMS data proc.	0.225	0.505	60.966	8.483
TLControl (2024)	0.982	0.341	11.132	7.066
- w/o SMS data proc.	0.980	0.345	13.881	7.344
Ours	0.988	**0.325**	3.892	6.199
- w/o SMS data proc.	**0.991**	0.499	13.494	6.893

Table 1: Quantitative results for the **Motion Generation with SOS Control** on the HumanML3D test set. **Bold** and underline indicates the **best** and the second-best result.

Ablation Study of Iterative Optimization

We assess the effects of diffusion-time and test-time iterative optimizations on control signal alignment by evaluating four settings: no iterative optimization (no opt.), diffusion-time optimization (diff. opt.), test-time optimization (test. opt.), and both optimizations applied (both opt.). As shown

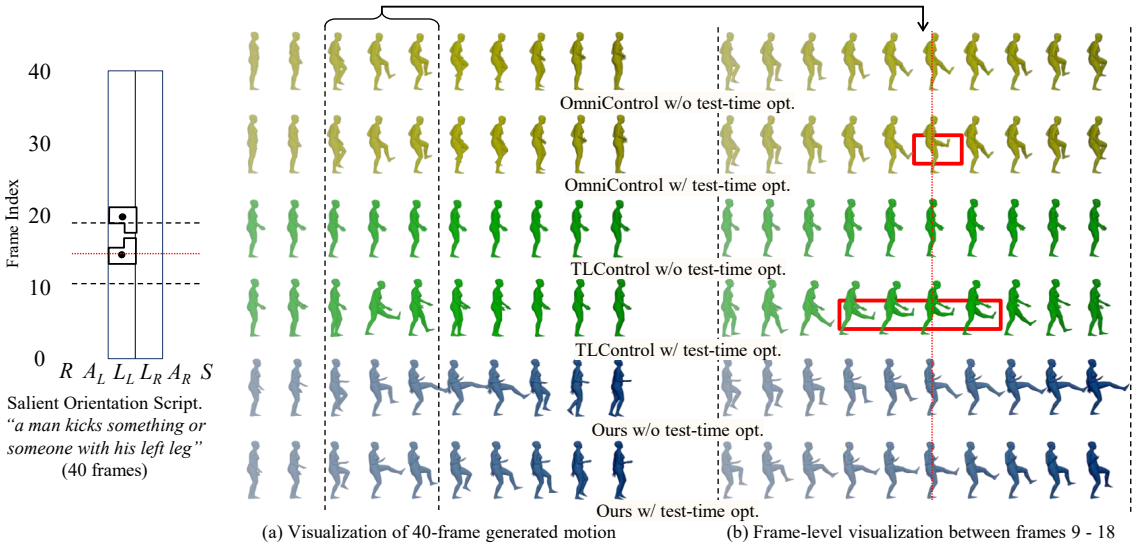


Figure 5: Visualization of SOS-conditioned motion generation results with and without test-time iterative optimization. (a) presents the motion sequences from left to right, while (b) provides a detailed view of the range between frames 9 and 18.

	SOS-Acc ↑	L2-Rot6D ↓	FID ↓	MMD ↓
GMD 1-stage (2023)				
- no opt.	0.111	0.425	24.999	7.871
- diff. opt.	0.113	0.427	25.669	7.835
OmniControl (2024)				
- no opt.	0.674	0.325	4.206	6.159
- diff. opt.	0.873	0.325	3.975	6.095
- test. opt.	0.956	**0.323**	**2.782**	5.992
- both opt.	0.956	**0.323**	3.025	**5.988**
TLControl (2024)				
- no opt.	0.162	0.334	42.012	8.015
- test. opt.	0.982	0.341	11.132	7.066
Ours				
- no opt.	0.531	0.329	5.570	6.382
- diff. opt.	0.535	0.329	5.187	6.335
- test. opt.	**0.988**	0.324	4.209	6.125
- both opt.	**0.988**	0.325	3.892	6.199

Table 2: Quantitative results for the impact of diffusion-time and test-time iterative optimizations.

in Tab. 2, test-time optimization delivers the most substantial improvements across all metrics. Diffusion-time optimization offers only marginal gains, likely because its adjustments can be overwritten in the diffusion model inference, limiting alignment to the input control signal. Notably, OmniControl achieves the best performance when both optimizations are applied. However, applying test-time optimization directly on the raw motion signal produces sparse guidance that affects only specific body part keyframes without propagating to adjacent frames, leading to inconsistencies between the modified part and overall motion. Fig. 5 illustrates these effects: test-time optimization in OmniControl only impacts frame 14, causing motion inconsistencies despite favorable metric values. In contrast, our method produces motions that closely follow the SOS specification and maintain overall consistency, benefiting from the ACTOR-PAE Decoder's ability to propagate sparse guidance to adjacent frames. While TLControl's VQ-VAE decoder also facilitate propagation, its limited codebook size restricts the output expressiveness, producing less smooth motions.

Additional Experiments and User Studies

In addition to experiments on SOS-conditioned motion generation and iterative optimization, we evaluate the effectiveness of each proposed module, analyze model performance under different saliency thresholds, and conduct experiments on the BABEL dataset. We also perform user studies to assess the interpretability of SOS extraction scripts and to evaluate the quality and control alignment of the generated motions. Detailed results from these experiments and user studies are presented in the supplementary material.

Conclusion

In this paper, we address the limitations of traditional text-to-motion generation by introducing the Salient Orientation Symbolic (SOS) script, a novel and programmable framework for specifying body part orientations and motion timing. By allowing automatic extraction of high-saliency keyframes, our approach facilitates the generation of saliency-aware sparse symbolic script. This provides an intuitive, user-programmable symbolic interface for interpretable visualization and precise motion control. Additionally, our SOSControl framework incorporates saliency-aware data augmentation and gradient-based iterative optimization to ensure that generated motions closely align with intended orientations and timings, while maintaining naturalness and smoothness. By bridging the gap between symbolic motion representation and language-guided generation, our contributions establish a foundation for more efficient and accessible workflows in animation and robotics.

Acknowledgments

This research was supported by the Theme-based Research Scheme, Research Grants Council of Hong Kong (T45-205/21-N), and the Guangdong and Hong Kong Universities "1+1+1" Joint Research Collaboration Scheme (2025A0505000003).

References

Au, H. Y.; Chen, J.; Jiang, J.; and Xiang, J. 2025. Deep Compositional Phase Diffusion for Long Motion Sequence Generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Chen, X.; Jiang, B.; Liu, W.; Huang, Z.; Fu, B.; Chen, T.; and Yu, G. 2023. Executing your Commands via Motion Diffusion in Latent Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18000–18010.

Delmas, G.; Weinzaepfel, P.; Lucas, T.; Moreno-Noguer, F.; and Rogez, G. 2022. Posescript: 3d human poses from natural language. In *European Conference on Computer Vision*, 346–362. Springer.

Guest, A. H. 2013. *Labanotation: the system of analyzing and recording movement*. Routledge.

Guo, C.; Zou, S.; Zuo, X.; Wang, S.; Ji, W.; Li, X.; and Cheng, L. 2022. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5152–5161.

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

Huang, Y.; Wan, W.; Yang, Y.; Callison-Burch, C.; Yatskar, M.; and Liu, L. 2024. Como: Controllable motion generation through language guided pose code editing. In *European Conference on Computer Vision*, 180–196. Springer.

Jiang, J.; Au, H. Y.; Chen, J.; Xiang, J.; and Chen, M. 2024. Motion Part-Level Interpolation and Manipulation over Automatic Symbolic Labanotation Annotation. In *2024 International Joint Conference on Neural Networks (IJCNN)*.

Jin, P.; Wu, Y.; Fan, Y.; Sun, Z.; Yang, W.; and Yuan, L. 2023. Act as you wish: Fine-grained control of motion diffusion model with hierarchical semantic graphs. *Advances in Neural Information Processing Systems*, 36: 15497–15518.

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

Li, L.; Yang, W.; Yu, X.; Xing, J.; and Zhang, X.-P. 2024. Translating Motion to Notation: Hand Labanotation for Intuitive and Comprehensive Hand Movement Documentation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 4092–4100.

Liu, X.; Li, Y.-L.; Zeng, A.; Zhou, Z.; You, Y.; and Lu, C. 2024. Bridging the gap between human motion and action semantics via kinematic phrases. In *European Conference on Computer Vision*, 223–240. Springer.

Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; and Black, M. J. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6): 248:1–248:16.

McFee, B.; Raffel, C.; Liang, D.; Ellis, D. P.; McVicar, M.; Battenberg, E.; and Nieto, O. 2015. librosa: Audio and Music Signal Analysis in Python. In *Proceedings of the 14th Python in Science Conference*, volume 8, 18–25.

Petrovich, M.; Black, M. J.; and Varol, G. 2021. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10985–10995.

Shafir, Y.; Tevet, G.; Kapon, R.; and Bermano, A. H. 2024. Human Motion Diffusion as a Generative Prior. In *Twelfth International Conference on Learning Representations*.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Starke, S.; Mason, I.; and Komura, T. 2022. Deepphase: Periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (TOG)*, 41(4): 1–13.

Tevet, G.; Raab, S.; Gordon, B.; Shafir, Y.; Cohen-or, D.; and Bermano, A. H. 2023. Human Motion Diffusion Model. In *The Eleventh International Conference on Learning Representations*.

Wan, W.; Dou, Z.; Komura, T.; Wang, W.; Jayaraman, D.; and Liu, L. 2024. Tlcontrol: Trajectory and language control for human motion synthesis. In *European Conference on Computer Vision*, 37–54. Springer.

Wang, Y.; Leng, Z.; Li, F. W.; Wu, S.-C.; and Liang, X. 2023. Fg-t2m: Fine-grained text-driven human motion generation via diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22035–22044.

Xie, Y.; Jampani, V.; Zhong, L.; Sun, D.; and Jiang, H. 2024. OmniControl: Control Any Joint at Any Time for Human Motion Generation. In *The Twelfth International Conference on Learning Representations*.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3836–3847.

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

Zhang, M.; Li, H.; Cai, Z.; Ren, J.; Yang, L.; and Liu, Z. 2023. Finemogen: Fine-grained spatio-temporal motion generation and editing. *Advances in Neural Information Processing Systems*, 36: 13981–13992.

Zhong, C.; Hu, L.; Zhang, Z.; and Xia, S. 2023. Attt2m: Text-driven human motion generation with multi-perspective attention mechanism. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 509–519.

Zhou, Y.; Barnes, C.; Lu, J.; Yang, J.; and Li, H. 2019. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5745–5753.