

Unifying Precise Keyframes and Semantic Control via Multi-level Diffusion

Supplementary Material

In this supplementary material, we provide comprehensive details in support of the main paper. Specifically, we present additional experiments in Sec. 7, describe our motion representation in Sec. 8, and provide implementation details for the text-conditioned in-betweening task in Sec. 9, including the network architecture, loss functions, and training and inference procedures. We further detail our inversion-based motion editing pipeline in Sec. 10. Finally, Sec. 11 outlines the experimental settings, baseline configurations, and evaluation metrics.

7. Additional Experiments

Flexibility of partial joint control. Our method allows for partial joint conditioning (e.g., joint positions without rotations or contact labels) and achieves competitive performance (see Tab. 6). This setting is particularly challenging because such partial conditioning is not observed during training. The root trajectory is additionally required for effective control, as all other joints are represented relative to it. If the root is unavailable, it could be estimated, which we leave for future work.

Table 6. Performance of our method under partial joint setting, where the root joint and a random subset of the five end-effector positions are constrained, while other features (e.g., joint rotations or contact labels) remain unconstrained.

Keyframe Error ↓	R-Precision (Top 3) ↑	MM Dist ↓	FID ↓	Skating Ratio ↓
0.000	0.794	2.537	0.120	0.051

Visualization of motion-keyframe attention. To analyze the semantic and temporal structure captured by the learned keyframe representations, we visualize the attention map between motion features and learned keyframe features for the example in Fig. 5 (b), as shown in Fig. 7. The orange boxes highlight the alignment between the attention peaks and the keyframes, showing that the attention map accurately captures the timing of the keyframes. Moreover, high-attention regions highlighted by blue and pink dashed

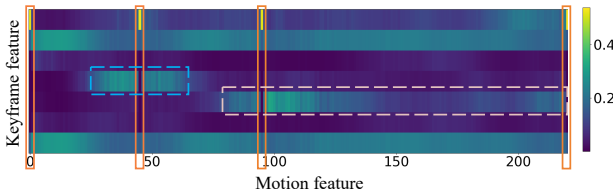


Figure 7. Visualization of the attention map between motion and keyframe features for the example in Fig. 5 (b).

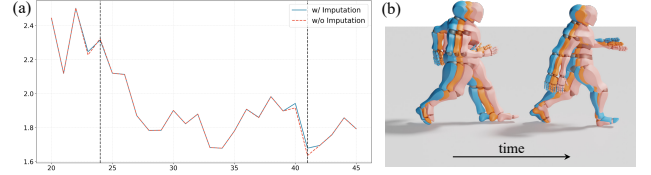


Figure 8. Smoothness analysis of motion transitions around keyframes, using the example of Fig. 1 (a) of the paper. (a) Speed profiles with and without imputation. No abrupt spikes or noticeable fluctuations are observed around the keyframes (dashed lines) when imputation is applied, indicating that imputation does not introduce perceptible jitter. (b) Visualization of natural motion transitions around keyframes. Orange frames denote keyframes, while blue and pink indicate the frames before and after each keyframe.

boxes are temporally aligned with the actions of “pick up” and “walk to the target”, indicating that keyframes influence semantically related motion segments. This suggests that the attention map effectively encodes semantic information.

Effect of refinement on motion realism. (i) Our experiments suggest that the additional foot sliding caused by our trajectory refinement module is negligible compared with baselines. Any misalignment between the refined trajectory and local body movements is progressively corrected by the diffusion model in subsequent denoising steps using its learned motion prior. In our ablation study, our trajectory refinement achieves zero keyframe root error with only a slight increase in skating ratio (from 4.21% to 4.28%, a 1.66% relative change). Despite this slight increase, our method still yields less foot skating than all baselines. (ii) We acknowledge that diffusion imputation may introduce slight jitter, but it is visually negligible (see Fig. 8). Moreover, even with imputation, our method achieves a lower jitter metric than all baseline methods (see Tab. 7).

Table 7. Comparison of motion jitter across different methods.

Methods	GT	OmniControl	CondMDI	MaskControl	Ours	Ours (w/o Imputation)
Jitter ($\times 10m/s^3$) ↓	6.885	61.715	67.203	18.037	<u>16.172</u>	14.109

8. Motion Representation

Our method represents the motion $\mathbf{x} \in \mathbb{R}^{N \times D}$ as a sequence of poses over N frames, where each pose $\mathbf{p} \in \mathbb{R}^D$ is represented by D features. We represent each pose as a combination of three components: (i) a global component containing the position of the root joint in world space $\mathbf{r}^p \in \mathbb{R}^3$ and global root rotation around the Y axis $\mathbf{r}^r \in \mathbb{R}^6$ represented in 6D rotations; (ii) a local pose component

containing joint positions relative to the root $\mathbf{j}^p \in \mathbb{R}^{3(J-1)}$, joint rotations relative to parent joints $\mathbf{j}^r \in \mathbb{R}^{6J}$ represented in 6D rotations, and foot-ground contact labels $\mathbf{c} \in \mathbb{R}^4$, where $J = 22$ is the number of joints; and (iii) a body shape component $\theta^b \in \mathbb{R}^{10}$ parameterized by the SMPL-X shape parameters [35]. Thus, the representation of each pose is defined as:

$$\mathbf{p} = \langle \mathbf{r}^p, \mathbf{r}^r, \mathbf{j}^p, \mathbf{j}^r, \mathbf{c}, \theta^b \rangle \in \mathbb{R}^{218}. \quad (7)$$

Our motion representation differs from previous formats, such as the relative-root-based H3D-Format [13], while being well-suited for controllable generation and practical animation workflows. In contrast to representations that operate in root-relative coordinates, we follow Cohan et al. [10] to explicitly model the root joint position \mathbf{r}^p and rotation \mathbf{r}^r in world space. This global representation allows us to directly constrain keyframe positions in world space, enabling precise spatio-temporal control over character trajectories—an essential requirement in interactive motion authoring. We exclude velocity-based representations because position and rotation are sufficient for animators to specify keyframe constraints, and explicitly specifying suitable velocities for each joint would be overly laborious for animators. Furthermore, incorporating the body-shape parameters θ^b allows our framework to generate motions for characters with diverse body shapes, enhancing the generalization of the resulting motions.

Following [27], we re-extract motions in SMPL-X format, resample them to 30 FPS, and keep sequences between 2 and 10 seconds (60–300 frames). Each sequence is represented with a 22-joint skeleton [13] and converted into our motion representation. Notably, joint rotations can be decomposed into twist and swing components [24]. The H3D-Format retargets motions to a unified skeleton and uses vanilla Inverse Kinematics (IK) [5] to recover rotations. However, this approach ignores twist rotation, potentially leading to incorrect end-effector orientations in the reconstructed motion. In contrast, by leveraging the SMPL-X representation directly, we do not require retargeting to a uniform skeleton. This allows us to preserve the original twist and swing components, resulting in more faithful end-effector orientations.

9. Text-conditioned Motion In-betweening

9.1. Network Architecture

Our motion diffusion backbone is implemented as a 1D convolutional U-Net with Adaptive Group Normalization (AdaGN) [32], following the architecture of Huang et al. [19]. The network uses a base channel width of 512 and channel multipliers of [1, 1, 1, 1].

For condition encoding, we use two Transformer Encoder modules: one for text and one for keyframe condi-

tions, both sharing the same architecture. Each Transformer consists of four encoder layers with a latent dimension of 512. The text branch processes pre-trained CLIP token embeddings [43], while the keyframe branch encodes the keyframe constraints. Following Chen et al. [8], we adopt 8 learnable latent tokens for each Transformer, yielding a final feature tensor of size 8×512 for both text and keyframe conditions.

9.2. Loss Function

Losses. Our training objective combines several complementary terms to ensure controllability and motion realism. Following the formulation of Tevet et al. [53], we predict the input motion, i.e., $\hat{\mathbf{x}}_0 = p_\theta(\mathbf{x}_t, t, c)$ with the objective:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0 | c), t \sim [1, T]} \left[\|\mathbf{x}_0 - p_\theta(\mathbf{x}_t, t, c)\|_2^2 \right], \quad (8)$$

where c represents the text and keyframe conditions in our context. In addition to the diffusion denoising loss, we apply a reconstruction loss between generated keyframes and input keyframe constraints:

$$\mathcal{L}_{\text{key}} = \|\mathbf{m}_{\mathbf{K}} \odot (\hat{\mathbf{x}}_0 - \mathbf{K})\|_2^2, \quad (9)$$

where \odot denotes the Hadamard product. To further enhance physical plausibility, we mitigate foot sliding by penalizing the velocity of feet in contact with the ground:

$$\mathcal{L}_{\text{foot}} = \frac{1}{N-1} \sum_{i=1}^{N-1} \left\| (\text{rec}(\hat{\mathbf{x}}_0^{i+1}) - \text{rec}(\hat{\mathbf{x}}_0^i)) \cdot \mathbf{f}_i \right\|_2^2, \quad (10)$$

where $\hat{\mathbf{x}}_0^i$ denotes the predicted motion features at i^{th} frame, $\text{rec}(\cdot)$ reconstructs the joint positions based on the predicted $\hat{\mathbf{x}}_0$, and \mathbf{f}_i is the binary foot-contact mask for i^{th} frame, following the formulation of Tevet et al. [53]. As a result, the training objective is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{diff}} + \lambda_{\text{key}} \mathcal{L}_{\text{key}} + \lambda_{\text{foot}} \mathcal{L}_{\text{foot}}, \quad (11)$$

where $\lambda_{\text{key}}, \lambda_{\text{foot}}$ are empirically set to 1 in our experiments.

9.3. Training Details

We train our framework with a batch size of 64 for 500K iterations. The diffusion model follows a DDPM formulation with $T = 1000$ denoising steps and a cosine noise schedule. We use the Adam optimizer with an initial learning rate of $1\text{e-}4$ and a weight decay of 0.01. To improve training stability, we adopt a learning rate decay of 1% every 5,000 steps, and apply gradient clipping with a maximum norm of 1. We further employ an exponential moving average (EMA) of the model parameters with $\beta = 0.9999$, and use the EMA-averaged model for inference to achieve improved generation quality.

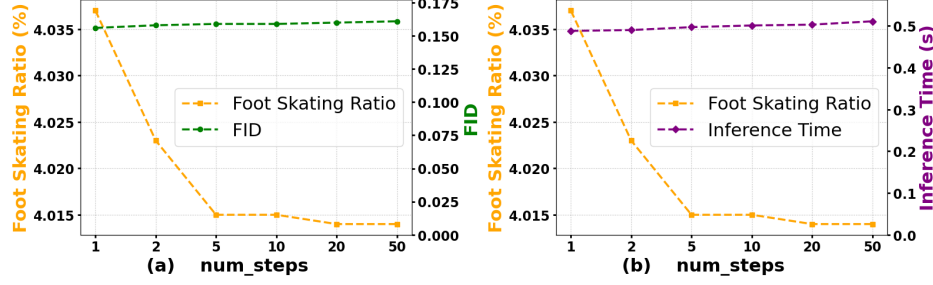


Figure 9. Impact of the number of refinement steps $S_{\text{threshold}}$ (applied during the final denoising steps) on: (a) Foot Skating Ratio and FID; and (b) Foot Skating Ratio and Inference Time. We select the first and last frames from the ground truth motions as keyframe constraints.

Following Cohan et al. [10], we assume that the keyframe signal and motion feature share the same dimensionality. Thus, for each constrained frame and joint, all corresponding features within the 218-dimensional vector must be provided. For instance, conditioning on the root joint requires providing both \mathbf{r}^p and \mathbf{r}^r for every frame. Conditioning on other joints also requires root joint information, since all joint positions are represented relative to the root. Similarly, foot contact information is accessible to the model only if the corresponding foot or ankle joints are observed.

To simulate diverse spatio-temporal constraints, we employ a stochastic masking mechanism during training. We first randomly sample k frame indices from the motion sequence. For each selected frame, we further sub-sample a subset of joints and set the corresponding entries of the observation mask \mathbf{m}_K to 1, while all unconstrained entries remain zero. The keyframe feature matrix \mathbf{K} is then constructed by retaining the ground-truth motion features at these masked positions and padding the rest with zeros. To improve robustness against missing control signals, both the text prompt and the keyframe constraint are independently replaced with a null state—an empty string \emptyset or an all-zero mask, respectively—with a probability of 10%.

9.4. Inference Details

We apply separate classifier-free guidance strategies for the text and keyframe conditions. Text provides high-level semantic guidance, mapping an abstract description (e.g. “a person walks sadly”) to a high-dimensional manifold of possible motions, which requires relatively strong guidance to ensure that the generated motion aligns with the textual semantics. In contrast, keyframes impose precise and unambiguous spatio-temporal constraints. We observe that applying classifier-free guidance to keyframes over-constrains the model, causing it to overfit the given poses and degrade natural motion dynamics. Therefore, we apply classifier-free guidance only for the text condition with a scale $\omega_t = 2.5$ following Huang et al. [19], while keeping the keyframe guidance unscaled. The sampling procedure, based on DDIM-50, is summarized in Algorithm 1,

and the trajectory refinement method is further detailed in Algorithm 2.

We explore the impact of the number of refinement steps $S_{\text{threshold}}$ applied during the final denoising steps. As illustrated in Fig. 9, increasing $S_{\text{threshold}}$ reduces the Foot Skating Ratio but results in higher FID and longer inference time. We hypothesize that initiating refinement at earlier denoising steps facilitates a gradual alignment of the generated motion with the keyframe constraints. This is achieved through an iterative cycle of refining the estimate, re-injecting noise, and subsequent denoising. This process effectively guides the intermediate latent states to progressively converge toward the constraints, thereby reducing the magnitude of root trajectory corrections required in the final steps and minimizing foot skating. However, forcing this hard refinement followed by re-noising at very early steps (high noise levels) can disrupt the natural diffusion trajectory, potentially degrading the overall motion quality. Based on this trade-off, we set $S_{\text{threshold}} = 5$.

Algorithm 1 DDIM Sampling with Inference Refinement

Require: Text prompt p , keyframe feature matrix \mathbf{K} , keyframe mask \mathbf{m}_K

Require: Time sequence $\tau = [\tau_1, \dots, \tau_S]$ with $\tau_S = T, \tau_0 = 0$

- 1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 2: **for** $i = S, \dots, 1$ **do**
- 3: $t \leftarrow \tau_i$
- 4: $s \leftarrow \tau_{i-1}$
- 5: $\hat{\mathbf{x}}_0^{\text{uncond}} = p_\theta(\mathbf{x}_t, t, \emptyset, \mathbf{K}, \mathbf{m}_K)$
- 6: $\hat{\mathbf{x}}_0^{\text{cond}} = p_\theta(\mathbf{x}_t, t, p, \mathbf{K}, \mathbf{m}_K)$
- 7: $\hat{\mathbf{x}}_0 = \hat{\mathbf{x}}_0^{\text{uncond}} + \omega_t(\hat{\mathbf{x}}_0^{\text{cond}} - \hat{\mathbf{x}}_0^{\text{uncond}})$
- 8: **if** $s \leq S_{\text{threshold}}$ **then**
- 9: $\tilde{\mathbf{x}}_0 = \text{TrajectoryRefinement}(\hat{\mathbf{x}}_0, \mathbf{K}, \mathbf{m}_K)$
- 10: $\hat{\mathbf{x}}_0 = \mathbf{K} \odot \mathbf{m}_K + \tilde{\mathbf{x}}_0 \odot (1 - \mathbf{m}_K)$
- 11: **end if**
- 12: $\hat{\mathbf{e}}_\theta = (\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \hat{\mathbf{x}}_0) / \sqrt{1 - \bar{\alpha}_t}$
- 13: $\mathbf{x}_s = \sqrt{\bar{\alpha}_s} \hat{\mathbf{x}}_0 + \sqrt{1 - \bar{\alpha}_s} \cdot \hat{\mathbf{e}}_\theta$
- 14: **end for**
- 15: **return** \mathbf{x}_0

Algorithm 2 Trajectory Refinement

Require: Predicted motion $\hat{\mathbf{x}}_0$, keyframe feature matrix \mathbf{K} , keyframe mask \mathbf{m}_K

Ensure: Refined motion $\tilde{\mathbf{x}}_0$

```
1:  $\tilde{\mathbf{x}}_0 \leftarrow \hat{\mathbf{x}}_0$ 
2: Extract keyframe indices  $\{k_0, k_1, \dots, k_M\}$  from  $\mathbf{m}_K$ 
3: if  $k_0 = 0$  then
4:    $\tilde{\mathbf{x}}_0^0 \leftarrow \text{ReplaceRoot}(\hat{\mathbf{x}}_0^0, \mathbf{R}(\mathbf{K}_0))$ 
5: end if
6: for  $i = 0, \dots, M$  do
7:   if  $i = 0$  then
8:      $n_{\text{start}} \leftarrow 0, n_{\text{end}} \leftarrow k_i$ 
9:   else
10:     $n_{\text{start}} \leftarrow k_{i-1}, n_{\text{end}} \leftarrow k_i$ 
11:   end if
12:    $\mathbf{K}_{\text{start}} \leftarrow \mathbf{K}_{n_{\text{start}}}, \mathbf{K}_{\text{end}} \leftarrow \mathbf{K}_{n_{\text{end}}}$ 
13:    $\Delta \mathbf{r} \leftarrow \mathbf{R}(\mathbf{K}_{\text{end}}) - \mathbf{R}(\hat{\mathbf{x}}_0^{n_{\text{end}}})$ 
14:   for  $n = n_{\text{start}}, \dots, n_{\text{end}} - 1$  do
15:      $\hat{\mathbf{v}}_n \leftarrow \mathbf{R}(\hat{\mathbf{x}}_0^{n+1}) - \mathbf{R}(\hat{\mathbf{x}}_0^n)$ 
16:   end for
17:   for  $d \in \{x, y, z\}$  do
18:      $W_d \leftarrow \sum_{s=n_{\text{start}}}^{n_{\text{end}}-1} |\hat{\mathbf{v}}_{s,d}|$ 
19:     for  $n = n_{\text{start}}, \dots, n_{\text{end}} - 1$  do
20:        $w_{n,d} \leftarrow |\hat{\mathbf{v}}_{n,d}|/W_d$ 
21:        $\hat{\mathbf{v}}_{n,d} \leftarrow \hat{\mathbf{v}}_{n,d} + w_{n,d} \cdot \Delta \mathbf{r}_d$ 
22:     end for
23:   end for
24:   for  $n = n_{\text{start}} + 1, \dots, n_{\text{end}}$  do
25:      $\hat{\mathbf{r}}_n \leftarrow \hat{\mathbf{r}}_{n-1} + \hat{\mathbf{v}}_{n-1}$ 
26:      $\tilde{\mathbf{x}}_0^n \leftarrow \text{ReplaceRoot}(\hat{\mathbf{x}}_0^n, \hat{\mathbf{r}}_n)$ 
27:   end for
28: end for
29: return  $\tilde{\mathbf{x}}_0$ 
```

10. Motion Editing

Our motion editing is achieved through an invert-and-sample process: The original motion is first inverted to a latent noise sequence \mathbf{x}_T using DDIM inversion [45]. The resulting latent noise serves as the starting point for the denoising process guided by the edited keyframe targets, producing motion that satisfies the constraints.

10.1. Diffusion Inversion

The diffusion inversion retrieves the corresponding noise map \mathbf{x}_T given an input \mathbf{x}_0 . The vanilla DDIM inversion [45] assumes that the ODE process of DDIM denoising is reversible in the limit of small steps, leading to the following inversion update:

$$\mathbf{x}_{t+1} = \sqrt{\bar{\alpha}_{t+1}} p_\theta(\mathbf{x}_t, t) + \sqrt{1 - \bar{\alpha}_{t+1}} \epsilon_\theta(\mathbf{x}_t, t), \quad (12)$$

where $\bar{\alpha}_{t+1}$ is a constant hyper-parameter, $p_\theta(\mathbf{x}_t, t)$ denotes the motion predicted by the model and $\epsilon_\theta(\mathbf{x}_t, t)$ denotes the noise predicted at time step t , derived from:

$$\epsilon_\theta(\mathbf{x}_t, t) = \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t} p_\theta(\mathbf{x}_t, t))}{\sqrt{1 - \bar{\alpha}_t}}, \quad (13)$$

In both Eq. 12 and Eq. 13, $p_\theta(\mathbf{x}_t, t)$ serves as an approximation of $p_\theta(\mathbf{x}_{t+1}, t+1)$, following the formulation in [45], and we omit condition c for clarity.

This vanilla DDIM inversion suffers from cumulative approximation errors, as the inversion process approximates $p_\theta(\mathbf{x}_{t+1}, t+1)$ with $p_\theta(\mathbf{x}_t, t)$, which introduces cumulative errors over inversion steps and leads to inaccurate reconstructions, causing the inversion trajectory to deviate from the denoising path, which in turn causes the edited motion to deviate from the original semantics. To address this, we employ the fixed-point iteration technique to preserve the semantics of the original sequence following [11, 34].

At each inversion step, we first calculate an initial estimate \mathbf{x}'_{t+1} using Eq. 12. As argued in [11], $p_\theta(\mathbf{x}'_{t+1}, t+1)$ provides a closer approximation to $p_\theta(\mathbf{x}_{t+1}, t+1)$ than $p_\theta(\mathbf{x}_t, t)$; therefore, we refine \mathbf{x}_{t+1} as:

$$\begin{aligned} \mathbf{x}_{t+1} = & \sqrt{\bar{\alpha}_{t+1}} p_\theta(\mathbf{x}'_{t+1}, t+1) \\ & + \sqrt{\frac{1 - \bar{\alpha}_{t+1}}{1 - \bar{\alpha}_t}} (\mathbf{x}_t - \sqrt{\bar{\alpha}_t} p_\theta(\mathbf{x}'_{t+1}, t+1)), \end{aligned} \quad (14)$$

This fixed-point refinement reduces the approximation error at each step, thus better retaining the original motion semantics in \mathbf{x}_T . We further explore the impact of the number of fixed-point iteration steps, as shown in Fig. 10.

10.2. Keyframe-guided Denoising

Given the inverted latent \mathbf{x}_T , we leverage target keyframes \mathbf{K} as a condition for the denoising process. The model then adaptively modifies motion dynamics to produce a plausible motion sequence that satisfies the keyframe constraints while preserving the original semantics. We further apply the inference refinement strategies in Sec. 3 in the main paper to ensure strict spatial adherence. Furthermore, our method enables strict keyframe adherence, which is crucial for precise local editing. For instance, to keep a motion segment unmodified, an artist can designate all frames within this segment as keyframes. Our approach then strictly enforces these constraints while plausibly re-synthesizing the modified motion parts.

10.3. Implementation Details

For the motion editing experiments in Sec. 5.2, we do not provide text conditions following DNO [22], ensuring a fair comparison (see Tab. 5 in the main paper). Our method employs 50 steps each for DDIM inversion and the subsequent keyframe-guided denoising. The inversion process does not

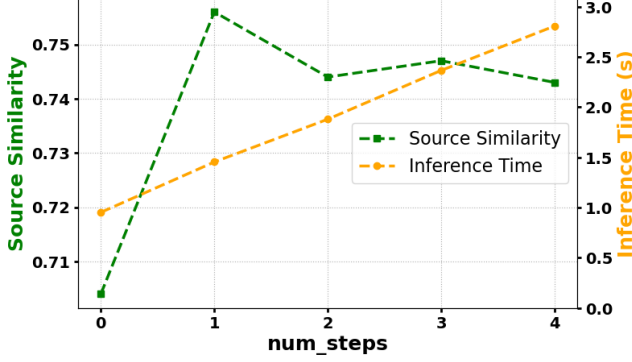


Figure 10. Impact of the number of fixed-point iteration steps on Source Similarity and Inference Time.

require keyframe conditions. We perform exactly one fixed-point iteration per timestep during inversion, as experiment results in Fig. 10 indicate this is sufficient: additional iterations yield no improvements while increasing inference time. For the DNO baseline, we retrained its MDM backbone on our dataset. We also set its DDIM inversion and denoising steps to 50, matching our method. Following their official configuration, we run the optimization for 300 steps per denoising step.

To construct our test set, we selected semantically similar motion pairs from the HumanML3D test set. Specifically, we leverage the MotionFix dataset [6], a language-based motion editing dataset that consists of semantically similar motion pairs. From this, we further selected 122 pairs and verified that all editing targets were contained within the HumanML3D test set. For pose editing scenarios, we randomly select 1–5 frames from the source motion and replace them with their corresponding poses from the target motion. For trajectory editing scenarios, we randomly select 1–5 frames from the source motion and then scale their (x, z) displacement relative to the first frame using random factors $s_x, s_z \sim \mathcal{U}[-1, 2]$. These factors are sampled once per sequence and applied consistently to all selected frames.

11. Experiment Settings

11.1. Baselines

Since we re-extract motions in the SMPL-X format and re-sample them to 30 FPS, we retrain all baseline models on this unified data format to ensure a fair comparison.

CondMDI. We utilize their global-root representation [10] augmented with body shape parameters $\theta^b \in \mathbb{R}^{10}$, yielding a 273-dimensional feature vector. During training, we adopt their *randomly sampled partial keyframes* training strategy, which aligns with our masking approach. For the diffusion process, we employ the DDPM framework with $T = 1000$ steps for both training and inference.

MaskControl. We utilize their relative-root representation [41] augmented with body shape parameters $\theta^b \in \mathbb{R}^{10}$, yielding a $(263 + 10)$ -dimensional feature vector. During training, we adapt their *Train on All Joints* strategy for keyframe conditioning by randomly selecting 22 joints as constraints, consistent with our approach. We adhere to the original training configurations for all other hyperparameters. For inference, we adopt the *Accurate* protocol defined in the original paper, using 600 iterations of *Logits Optimization* in the last step of the unmasking process and 100 iterations during steps 1 to 9 of the unmasking process.

We observe that the results generated by our adapted implementation of MaskControl are slightly worse than those shown in the original visualizations. We attribute this discrepancy to differences in task settings. In particular, MaskControl relies on the optimization for spatial constraints, which makes the problem more challenging in our setting. (i) Compared to partial joint constraints, keyframe constraints are spatially dense, requiring all joint constraints to be satisfied simultaneously, which makes optimization much harder and can lead to over-constrained motions and artifacts such as drifting (see the foot skating ratios in Tab. 1 and Tab. 3). (ii) Optimization inherently risks causing motion latents to deviate from the learned distribution, degrading motion quality. This is particularly evident with temporally sparse constraints, where insufficient guidance leads to under-constrained divergence and degraded naturalness.

OmniControl. We adopt the same data representation as MaskControl for OmniControl. Following the original method, the model is first pre-trained using the MDM [53] model and subsequently trained with the OmniControl configuration. Inference is performed using DDPM ($T = 1000$). Adhering to their inference protocol, we apply spatial guidance with 10 optimization steps per timestep for the first 990 denoising steps, and increase this to 500 optimization steps for the final 10 denoising steps.

SFControl. For the second stage of training, we adopt the same data representation used in MaskControl. Notably, during the first stage (trajectory generation), we adhere to the official codebase, which includes additional loss terms not documented in the original paper: position reconstruction $\mathcal{L}_{\text{recon}}$, velocity reconstruction \mathcal{L}_{vel} , and foot skating $\mathcal{L}_{\text{foot_skating}}$. We incorporate these losses using the weights specified in their implementation. Furthermore, following the open-source code, we construct the trajectory representation at each frame as a concatenation of three components: (i) global root orientation $r^r \in \mathbb{R}^1$, (ii) global root position $r^p \in \mathbb{R}^3$, and (iii) end-effector positions $e^p \in \mathbb{R}^{3J}$ (relative to the root), where $J = 5$ denotes the number of end-effector joints.

11.2. Evaluation Metrics

Motion representation for evaluation. Since baseline methods employ distinct motion representations, which often contain varying degrees of redundancy, a direct comparison of their raw model outputs is infeasible. Given that the quality of the reconstructed motion is what ultimately matters, we evaluate all methods based on reconstructed 3D joint positions, which provide a consistent and fair basis for quantitative comparison. To ensure our metrics capture the intrinsic quality and semantics of the motion, invariant to absolute global translation, we further transform this reconstructed data into a unified representation. Specifically, the final evaluation representation used for metric computation consists of: (i) velocity-based root position and rotation, and (ii) local joint position and velocity within the root coordinate system.

Evaluation model. Since the pre-trained encoders from prior work [13] are incompatible with our evaluation representation, we retrain a text-motion alignment model following Lu et al. [28]. This model employs a VAE-based architecture [37] consisting of a motion encoder, a text encoder, and a motion decoder. The training objective is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{E}}\mathcal{L}_{\text{E}}, \quad (15)$$

where the reconstruction loss term \mathcal{L}_{rec} measures the fidelity of the reconstructed motion given text or motion input. The Kullback-Leibler (KL) divergence loss \mathcal{L}_{KL} regularizes the encoded latent distributions— $\mathcal{N}(\mu_M, \Sigma_M)$ for motion and $\mathcal{N}(\mu_T, \Sigma_T)$ for text—to align with a standard normal distribution, $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Finally, a cross-modal embedding similarity loss \mathcal{L}_{E} enforces the latent codes for text (\mathbf{z}_T) and motion (\mathbf{z}_M) to be similar to each other. We set λ_{KL} to 10^{-5} and λ_{E} to 0.1 in our experiments. This retrained backbone serves as the foundation for computing FID and semantic alignment metrics.

Motion quality. To assess motion quality, we employ the Frechet Inception Distance (FID) metric, which compares the feature distributions of generated and real motions using our pre-trained motion encoder. Additionally, we incorporate the foot skating ratio metric proposed by [21] into our motion quality evaluation. Following their criteria, foot skating is detected when the foot’s displacement exceeds 2.5 cm per frame while maintaining ground contact (foot height < 5 cm). Instead of calculating the proportion of frames in which either foot skids, we compute the sliding ratio for each foot independently and report the average across both feet.

High-level semantic alignment. For high-level semantic alignment, we use motion-retrieval precision (R-precision) and Multimodal Distance (MM Dist) to measure how well the generated motions align with the input text prompts following Tevet et al. [53]. These metrics leverage the pre-

trained models to map motion and coarse-grained text into a shared semantic space, which captures high-level semantics such as action categories, but overlooks low-level motion semantics such as precise timing. For example, given the text prompt “A person initially running and decelerating to walking” with keyframes K_1 specifying a *run* pose and K_2 specifying a *walk* pose, these metrics can assess whether the generated motion contains the *walk* and *run* actions, but cannot evaluate whether the timing of each action and the transitions between them align with the spatio-temporal constraints implied by the keyframes.

Low-level semantic alignment. We aim to assess whether each generated *inter-keyframe transition* (i.e., the segment between two consecutive keyframes) adheres to the low-level semantics implied by the keyframe cues combined with the text prompt. However, explicit semantic labels (e.g., precise text descriptions) are typically unavailable for these intermediate segments. To address this, we adopt the corresponding ground-truth motion segment as the proxy for the intended semantics. We thus propose a novel Segment-level Semantic Similarity (SS Similarity) metric, which quantifies the semantic similarity between the generated inter-keyframe transition and its corresponding ground-truth motion segment. Since the ground truth motion inherently encapsulates the precise timing constraints between keyframes, a high similarity implies that the generated motion exhibits temporal dynamics consistent with the ground truth at the segment level, thereby suggesting better alignment with the timing specified by the keyframes.

Specifically, we leverage a pre-trained TMR model [39] to project both the generated and ground-truth segments into a shared semantic latent space, computing their cosine similarity. Our TMR model adopts the same architecture as our text-motion alignment model but is optimized with an additional contrastive objective:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{E}}\mathcal{L}_{\text{E}} + \lambda_{\text{NCE}}\mathcal{L}_{\text{NCE}}, \quad (16)$$

where \mathcal{L}_{NCE} denotes the InfoNCE loss [33]. Since InfoNCE optimizes the embedding space by maximizing the cosine similarity between positive pairs (and minimizing it for negative ones), it explicitly structures the latent manifold to reflect semantic similarity via the cosine metric. Accordingly, we set $\lambda_{\text{KL}} = \lambda_{\text{E}} = 10^{-5}$ and $\lambda_{\text{NCE}} = 0.1$ to prioritize the optimization of cosine-based semantic alignment.

Moreover, we validate that cosine similarity in this TMR motion space serves as a reliable proxy for semantic similarity. We observe a Pearson correlation coefficient of 0.826 between the cosine similarity of motion embedding pairs and the cosine similarity of their corresponding text embedding pairs in the learned TMR space. This strong correlation suggests that high cosine similarity to the ground-truth motion in the TMR space is a useful proxy for alignment with the underlying textual semantics.

References

- [1] Dhruv Agrawal, Martin Guay, Jakob Buhmann, Dominik Borer, and Robert W. Sumner. Pose and skeleton-aware neural IK for pose and motion editing. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10, 2023. 3
- [2] Dhruv Agrawal, Jakob Buhmann, Dominik Borer, Robert W Sumner, and Martin Guay. Skel-betweener: a neural motion rig for interactive motion authoring. *ACM Transactions on Graphics (TOG)*, 43(6):1–11, 2024. 3
- [3] Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. In *2019 International Conference on 3D Vision (3DV)*, pages 719–728. IEEE, 2019. 3
- [4] Elly Akhouni, Hung Yu Ling, Anup Deshmukh, and Judith Butepage. Silk: Smooth interpolation framework for motion in-betweening a simplified computational approach. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2891–2900. IEEE, 2025. 3
- [5] Andreas Aristidou and Joan Lasenby. Fabrik: A fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 73(5):243–260, 2011. 2
- [6] Nikos Athanasiou, Alpár Cseke, Markos Diomataris, Michael J Black, and Gül Varol. Motionfix: Text-driven 3d human motion editing. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 3, 5
- [7] Ling-Hao Chen, Shunlin Lu, Wenxun Dai, Zhiyang Dou, Xuan Ju, Jingbo Wang, Taku Komura, and Lei Zhang. Pay attention and move better: Harnessing attention for interactive motion generation and training-free editing. *arXiv preprint arXiv:2410.18977*, 2024. 3
- [8] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18000–18010, 2023. 3, 2
- [9] Loïc Ciccone, Cengiz Öztireli, and Robert W Sumner. Tangent-space optimization for interactive animation control. *ACM Transactions on Graphics (TOG)*, 38(4):1–10, 2019. 2
- [10] Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. Flexible motion in-betweening with diffusion models. In *ACM SIGGRAPH 2024 conference papers*, pages 1–9, 2024. 2, 3, 6, 7, 5
- [11] Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise: Real image inversion through iterative noising. In *European Conference on Computer Vision*, pages 395–413. Springer, 2024. 5, 4
- [12] Purvi Goel, Kuan-Chieh Wang, C Karen Liu, and Kayvon Fatahalian. Iterative motion editing with natural language. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–9, 2024. 3
- [13] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022. 6, 2
- [14] Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1900–1910, 2024. 3
- [15] Félix G Harvey and Christopher Pal. Recurrent transition networks for character locomotion. In *SIGGRAPH Asia 2018 Technical Briefs*, pages 1–4. 2018. 3
- [16] Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher J. Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4):60, 2020. 3
- [17] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016. 3
- [18] Seokhyeon Hong, Chaelin Kim, Serin Yoon, Junghyun Nam, Sihun Cha, and Junyong Noh. Salad: Skeleton-aware latent diffusion for text-driven motion generation and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7158–7168, 2025. 3
- [19] Yiheng Huang, Hui Yang, Chuanchen Luo, Yuxi Wang, Shibiao Xu, Zhaoxiang Zhang, Man Zhang, and Junran Peng. Stablemofusion: Towards robust and efficient diffusion-based motion generation framework. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 224–232, 2024. 3, 4, 2
- [20] Inwoo Hwang, Jinseok Bae, Donggeun Lim, and Young Min Kim. Motion synthesis with sparse and flexible keyjoint control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13203–13213, 2025. 3, 6, 7
- [21] Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2151–2162, 2023. 3, 6
- [22] Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1334–1345, 2024. 3, 8, 4
- [23] Chuqiao Li, Julian Chibane, Yannan He, Naama Pearl, Andreas Geiger, and Gerard Pons-Moll. Unimotion: Unifying 3d human motion synthesis and understanding. In *2025 International Conference on 3D Vision (3DV)*, pages 240–249. IEEE, 2025. 3
- [24] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3383–3393, 2021. 2
- [25] Zhengyuan Li, Kai Cheng, Anindita Ghosh, Uttaran Bhattacharya, Liangyan Gui, and Aniket Bera. Simmotionedit: Text-based human motion editing with motion similarity prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27827–27837, 2025. 3

- [26] Han Liang, Jiacheng Bao, Ruichi Zhang, Sihan Ren, Yuecheng Xu, Sibe Yang, Xin Chen, Jingyi Yu, and Lan Xu. Omg: Towards open-vocabulary motion generation via mixture of controllers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 482–493, 2024. 3
- [27] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 6, 2
- [28] Shunlin Lu, Ling-Hao Chen, Ailing Zeng, Jing Lin, Ruimao Zhang, Lei Zhang, and Heung-Yeung Shum. Humantomato: text-aligned whole-body motion generation. In *Proceedings of the 41st International Conference on Machine Learning*, pages 32939–32977, 2024. 6
- [29] Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11451–11461. IEEE, 2022. 5
- [30] Zichong Meng, Zeyu Han, Xiaogang Peng, Yiming Xie, and Huaizu Jiang. Absolute coordinates make motion generation easy, 2025. 3
- [31] Clinton A Mo, Kun Hu, Chengjiang Long, and Zhiyong Wang. Continuous intermediate token learning with implicit motion manifold for keyframe based motion interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13894–13903, 2023. 3
- [32] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2
- [33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 6
- [34] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–11, 2023. 5, 4
- [35] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10975–10985, 2019. 6, 2
- [36] Jiawen Peng, Zhuoran Liu, Jingzhong Lin, and Gaoqi He. Precise motion inbetweening via bidirectional autoregressive diffusion models. *Computer Animation and Virtual Worlds*, 36(3):e70040, 2025. 3
- [37] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021. 6
- [38] Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision*, pages 480–497. Springer, 2022. 3
- [39] Mathis Petrovich, Michael J Black, and Gül Varol. Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9488–9497, 2023. 6
- [40] Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. Mmm: Generative masked motion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1546–1555, 2024. 3
- [41] Ekkasit Pinyoanuntapong, Muhammad Saleem, Korrawe Karunratanakul, Pu Wang, Hongfei Xue, Chen Chen, Chuan Guo, Junli Cao, Jian Ren, and Sergey Tulyakov. Maskcontrol: Spatio-temporal control for masked motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9955–9965, 2025. 2, 3, 6, 7, 5
- [42] Jia Qin, Youyi Zheng, and Kun Zhou. Motion in-betweening via two-stage transformers. *ACM Transactions on Graphics (TOG)*, 41(6):184:1–184:16, 2022. 3
- [43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 3, 2
- [44] Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. Human motion diffusion as a generative prior. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 3
- [45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 5, 4
- [46] Paul Starke, Sebastian Starke, Taku Komura, and Frank Steinicke. Motion in-betweening with phase manifolds. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 6(3):1–17, 2023. 3
- [47] Justin Studer, Dhruv Agrawal, Dominik Borer, Seyed-morteza Sadat, Robert W Sumner, Martin Guay, and Jakob Buhmann. Factorized motion diffusion for precise and character-agnostic motion inbetweening. In *Proceedings of the 17th ACM SIGGRAPH conference on motion, interaction, and games*, pages 1–10, 2024. 3
- [48] Haowen Sun, Ruikun Zheng, Haibin Huang, Chongyang Ma, Hui Huang, and Ruizhen Hu. Lgtm: Local-to-global text-driven human motion diffusion model. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–9, 2024. 3
- [49] Xiangjun Tang, He Wang, Bo Hu, Xu Gong, Ruifan Yi, Qilong Kou, and Xiaogang Jin. Real-time controllable motion transition for characters. *ACM Transactions on Graphics (TOG)*, 41(4):1–10, 2022. 3
- [50] Xiangjun Tang, Linjun Wu, He Wang, Bo Hu, Xu Gong, Yuchen Liao, Songnan Li, Qilong Kou, and Xiaogang Jin.

Rsmr: Real-time stylized motion transition for characters. In *SIGGRAPH '23 Conference Proceedings*, pages 1–10, 2023.

[3](#)

- [51] Xiangjun Tang, Linjun Wu, He Wang, Yiqian Wu, Bo Hu, Songnan Li, Xu Gong, Yuchen Liao, Qilong Kou, and Xiaogang Jin. Decoupling contact for fine-grained motion style transfer. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. [5](#)
- [52] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *European Conference on Computer Vision*, pages 358–374. Springer, 2022. [3](#)
- [53] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. [3](#), [4](#), [6](#), [2](#), [5](#)
- [54] Weilin Wan, Zhiyang Dou, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. Tlcontrol: Trajectory and language control for human motion synthesis. In *European Conference on Computer Vision*, pages 37–54, 2024. [3](#)
- [55] Zhiming Wang, Ning Ge, and Jianhua Lu. Motion in-betweening with spatial and temporal transformers. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. [3](#)
- [56] Andrew P. Witkin and Michael Kass. Spacetime constraints. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1988, Atlanta, Georgia, USA, August 1-5, 1988*, pages 159–168. ACM, 1988. [3](#)
- [57] Linjun Wu, Xiangjun Tang, Jingyuan Cong, He Wang, Bo Hu, Xu Gong, Songnan Li, Yuchen Liao, Yiqian Wu, Chen Liu, et al. Semantically consistent text-to-motion with unsupervised styles. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–10, 2025. [3](#)
- [58] Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicontrol: Control any joint at any time for human motion generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [3](#), [4](#), [6](#), [7](#)
- [59] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14730–14740, 2023. [3](#)
- [60] Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. Finemogen: Fine-grained spatio-temporal motion generation and editing. *Advances in Neural Information Processing Systems*, 36:13981–13992, 2023. [3](#)
- [61] Bowen Zheng, Ke Chen, Yuxin Yao, Zijiao Zeng, Xinwei Jiang, He Wang, Joan Lasenby, and Xiaogang Jin. Autokeyframe: Autoregressive keyframe generation for human motion synthesis and editing. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–12, 2025. [3](#)