SimMotionEdit: Text-Based Human Motion Editing with Motion Similarity Prediction

Zhengyuan Li[1]   Kai Cheng[1]   Anindita Ghosh[2]   Uttaran Bhattacharya[3]   Liangyan Gui[4]   Aniket Bera[1]

[1]Purdue University   [2]DFKI, MPI-INF, Saarland Informatics Campus
[3]Adobe Inc.   [4]University of Illinois Urbana-Champaign

Abstract

Text-based 3D human motion editing is a critical yet challenging task in computer vision and graphics. While training-free approaches have been explored, the recent release of the MotionFix dataset, which includes source-text-motion triplets, has opened new avenues for training, yielding promising results. However, existing methods struggle with precise control, often leading to misalignment between motion semantics and language instructions. In this paper, we introduce a related task — motion similarity prediction — and propose a multi-task training paradigm, where we train the model jointly on motion editing and motion similarity prediction to foster the learning of semantically meaningful representations. To complement this task, we design an advanced Diffusion-Transformer-based architecture that separately handles motion similarity prediction and motion editing. Extensive experiments demonstrate the state-of-the-art performance of our approach in both editing alignment and fidelity. Project URL: https://github.com/lzhyu/SimMotionEdit.

1. Introduction

Human motion synthesis plays a central role in diverse applications, including character animation [5, 15, 49], human-robot collaborations [3, 28], and autonomous driving [36]. Current approaches primarily focus on initial control signals, such as manipulating joint positions [56, 59] or applying fine-grained language instructions [26, 65]. With advancements in text-based motion generation [6, 14, 21, 22, 53], the ability to control and edit human motions has emerged as the next frontier, which can enable animators to refine motion sequences in animation workflows to fit specific requirements. In this work, we aim to address this need by focusing on editing existing human motions rather than generating them solely from other conditioning inputs.

Recent works [8, 41] have leveraged pre-trained motion diffusion models with attention manipulation to edit motion
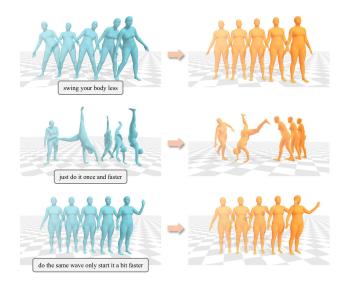

Figure 1. **Text-Based Motion Editing.** Our method SimMotionEdit generates edited human motion sequences from text instructions and source motion sequences.

features, achieving effects such as style transfer and word emphasis. However, these methods are limited in scope due to the lack of dedicated model training and supervision. With the release of the text-based human motion editing dataset MotionFix [4], we can now incorporate robust supervision to develop SimMotionEdit, a system that enables free-form text-based human motion editing.

The primary challenge in this task lies in aligning the generated motion with both the source motion and the text instructions, as they jointly define the desired outcome. Enhancing the interaction between the source motion and the text instructions is thus crucial for improved performance. To this end, the success of multi-task learning [67] has demonstrated that models can achieve better generalization by leveraging knowledge from other tasks. Auxiliary loss, a key concept in multi-task learning, has proven effective across various domains, including image classification [51],

scene parsing [68], reinforcement learning [29], and code generation [13]. The essential criterion in designing auxiliary tasks is that they should support the learning of representations that benefit the main task. Consider the example of an animator instructed to edit a human motion sequence based on an editing description and a source motion. The animator does not begin editing immediately — a common first step is to identify the keyframes to modify. Similarly, for our model, identifying keyframes to edit is not only essential but also beneficial to performance. In many cases, the keyframes to edit are predictable. For example, given the instruction "don't turn back at the end; instead, keep walking", it is reasonable to anticipate frame changes at the point where the character starts turning back. As a result, the frames where the edit happens are dissimilar to the corresponding frames in the source motion. Following this intuition, we apply techniques such as sliding windows, value normalization, and weight balancing across different similarity measures to construct a predictable time series of motion similarity. We formulate the task as *motion similarity prediction*, where the model predicts the similarity between the source and the edited motions given the edit instructions.

Given the multitask nature of learning motion editing via motion similarity prediction, we opt for the diffusion framework due to its flexibility. While various network architectures [53, 65, 66] have been proposed for human motion synthesis, they are not directly applicable to human motion editing, as most of them are designed to operate solely with text as input. Further, we separate our neural network into two parts: a condition transformer for motion similarity prediction and a diffusion transformer [37] for motion editing. This design choice offers two key advantages. First, the conditional transformer helps enhance the text and source motion features to embed information on motion similarity. The diffusion transformer effectively leverages those embeddings to generate the edited motion. Second, the condition transformer handles motion similarity prediction *decoupled* from the diffusion process, which allows each part of the neural network to specialize in its individual task and expedites as well as stabilizes training convergence.

We summarize our key contributions as follows:

- *SimMotionEdit*, a text-based human motion editing framework with an auxiliary task of motion similarity prediction to efficiently inter-relate source motion features and instruction text features.
- A Motion Diffusion Transformer architecture, trained on motion similarity prediction and edited motion generation to effectively learn text-based motion editing.
- Demonstrating the state-of-the-art performance of our architecture and our auxiliary task through extensive quantitative and qualitative experiments on the open-source dataset of MotionFix [4].

2. Related Works

We briefly discuss existing literature on the related areas of 3D human motion editing, diffusion models for motion generation, and auxiliary tasks used to train neural networks.

3D Human Motion Editing. Earlier works on 3D human motion editing have explored editing motions with space or time constraints [17, 18, 30]. However, generating realistic and coordinated motion editing typically requires users to manually provide such specific constraints. To this end, deep learning approaches have explored automating the process of human motion editing with simple inputs by incorporating different sub-tasks, such as motion stylization [1, 61], pose editing [35], and in-betweening [40, 47, 53, 54]. While techniques like motion stylization can perform global edits on motions, they do not enable precise control. Further, these approaches cannot be applied to the free-form text-based setting. Kim et al. [27] edit motions through diffusion inpainting, which requires a mask as input. Raab et al. [41] perform motion following by replacing the queries in the self-attention. MotionCLR [8] leverages attention manipulation to provide a set of motion editing methods. However, it does not directly take in texts as edit conditions. Goel et al. [19] heuristically select keyframes to edit certain types of motions. For text-based human pose editing, PoseFix [12] enables free-form text as input. Zhang et al. [65] relies on large language models to rewrite the script and generate edited motions and only considers the original motion at the text level. However, it cannot be applied without providing a description of the source motion. Huang et al. [24] prompts GPT to edit latent codes and generate edited human motions. MotionFix [4] provides a dataset and proposes a diffusion-based method to perform motion editing. Our task follows MotionFix [4], and we propose a novel architecture and an auxiliary task to improve generative performance.

Motion Generation via Diffusion Models. Diffusion models [23] have been widely applied to human motion generation from text [9–11, 31, 32, 45, 53, 60, 63, 64] and other modalities [16, 44, 50, 54, 58].The state-of-the-art motion diffusion models collapse text input into a single token. Shafir et al. [47] modify MDM's [53] motion inpainting process to control end-effector trajectories but require dense multi-modal inputs like entire joint trajectories instead of the free-form text-based instructions. With the introduction of recent text-motion 3D human datasets [20, 33, 39], there has been increased interest in conditioning the generation on free-form language inputs, in which [8, 47, 53, 55, 59, 64] have produced impressive results in free-from-text-based 3D human motion generation using diffusion models.

(a) Motion Diffusion Transformer
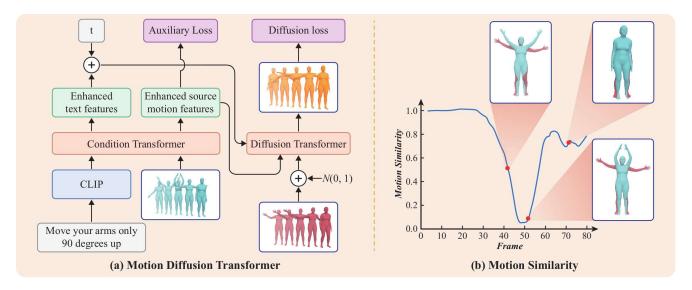
(b) Motion Similarity

Figure 2. **Overview of SimMotionEdit.** (a) The architecture consists of two modules: the condition transformer and the diffusion transformer. The condition transformer performs the auxiliary task of motion similarity prediction and enables the source motion features and the text features to mix. The diffusion transformer takes in the enhanced text features, the embedded diffusion step t as the condition, the noisy edited motion, and the enriched source motion features, and predicts the denoised edited motions. (b) The auxiliary task motion similarity prediction is inspired by the fact that, given the text instructions, the similarity between source and edited motions is predictable. We use blue for the source motion, red for the edited motion, and orange for the generated motion.

Auxiliary Tasks. Auxiliary tasks have played a significant role in enhancing the training of neural networks. Earlier works such as GoogLeNet (Inception v1) [51] added an auxiliary loss at intermediate layers to provide additional gradient signals to mitigate vanishing gradients. PSPNet[68] incorporated an auxiliary loss at intermediate layers for the scene parsing task. Auxiliary losses have also been leveraged in techniques including deep reinforcement learning [25, 29] and code generation [13] to enhance learning stability and feature extraction. Recently, REPA [62] finds that aligning the features of the denoiser and pre-trained discriminative models substantially boosts the training of diffusion models. Other works [46, 52] have also shown that auxiliary tasks can significantly improve the model's generalization by providing complementary signals to align the model's internal representations. However, to the best of our knowledge, auxiliary losses have not been explored previously for text-based human motion editing.

3. Method

Given a text instruction L and a source motion sequence $X = [x^0, x^1, ..., x^F]$ with F frames, our objective is to generate an edited sequence $M = [m^0, m^1, ..., m^{F'}]$ with F' frames, where each $m^i \in \mathbb{R}^D$ represents a human pose, D being the feature dimension. Following MotionFix [4], each m^i consists of global velocity, global orientation, joint rotations, and local joint positions, totaling $D = 207$. We illustrate our network architecture, the Motion Diffusion

Transformer, in Fig. 2. It consists of a condition transformer and a diffusion transformer. The condition transformer processes the source motion X and the text instructions L, and provides enhanced condition features via motion similarity prediction (Sec. 3.2). The diffusion transformer takes in the concatenated sequence of enhanced source motion features and noised edited motion, and uses the enhanced text features as conditions, to generate edited motions (Sec. 3.3). We train the network jointly on motion similarity prediction and diffusion-based motion editing (Sec. 3.1).

3.1. Diffusion-Based Motion Editing

Denoising diffusion models [23, 48] are generative models that use a structured sequence of transformations based on a Markov chain. The diffusion process comprises a forward step, $q(M_t \mid M_{t-1})$, where we iteratively add Gaussian noise to a clean edited motion sample M_0 over T timesteps until it resembles the distribution of $p(M_T)$:

$$M_t = \sqrt{\bar{\alpha}_t}M_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \tag{1}$$

where $\bar{\alpha}_t \in (0, 1)$ controls the rate of diffusion and $t \in [0, T]$. When $\bar{\alpha}_t$ is small enough, we can approximate $M_T \approx \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{I}$ being the unit matrix. In the subsequent reverse step, the diffusion process samples motions from a multivariate Gaussian distribution $M_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$,

and we obtain the conditional probability of M_0 as

$$p_\theta \left(M_0 \mid c\right) = \int_{M_1}^{M_T} p\left(m_T\right) \prod_{u=1}^{T} p_\theta\left(m_{u-1} \mid m_u, c\right) \mathrm{d}\mathbf{m}. \tag{2}$$

For text-conditioned motion editing, we approximate $p_\theta((M_{t-1} \mid M_t)$ with a learnable function $\mathcal{E}\left(M_t, t, L, X\right)$, which conditions the edited motion on both the text instruction L and the source motion X. Following recent approaches [4, 19, 43, 53], we predict the original motion signal $M_0 = \mathcal{E}\left(M_t, t, L, X\right)$ via the editing loss function,

$$\mathcal{L}_e = \mathbb{E}_{M_0 \sim q(M_0 \mid L, X), t \sim [1,T]} \left[\left\|M_0 - \mathcal{E}\left(M_t, t, L, X\right)\right\|_2^2\right]. \tag{3}$$

3.2. Motion Similarity Prediction

Our auxiliary task of motion similarity prediction follows the intuition that to edit a motion, the model must first identify the parts that need editing. Put differently, given the source and the edited motions, it should quantify the similarities between them. To design this auxiliary task, we consider the following two aspects:

- Constructing a *predictable similarity curve* (over time) between the source and the edited motions, and
- Designing an *auxiliary loss function* that synergizes with motion editing.

3.2.1. Predictable Motion Similarity Curve

In our work, we consider the motion similarity $\mathbf{S}^R$ between two 3D motion sequences in the space of joint rotations to leverage its scale-agnostic nature. In the rotation space, we define the raw similarity S_i^{Rr} of one frame i in X as the minimum distance between this frame and any frame within a corresponding sliding window of the edited motion, as

$$S_i^{Rr} = -\min_{|i-j| \leq W} d_r\left(x^i, m^j\right), \tag{4}$$

where W is the size of the sliding window, and d_r represents a distance metric (*e.g.*, Euclidean) between the source and the edited poses. The sliding window approach allows each source frame to *find* its best match in the edited motion within an acceptable range of misalignment. Note that if, instead, we calculate the difference between corresponding frames with identical indices, *i.e.*, x^i and m^i, the raw similarity S^r would be low for motions that are nearly identical but shifted by a few frames. Symmetrically, we define S_i^{Rl} in the joint location space. We balance the effects of S^{Rl} and S^{Rr} by empirically weighting them, as

$$S_i^R = w_1 S_i^{Rr} + w_2 S_i^{Rl}. \tag{5}$$

However, S_i^R is not entirely predictable due to two issues. First, different editing instructions can result in varying scales of change. For instance, both (a) and (b) in Fig. 3
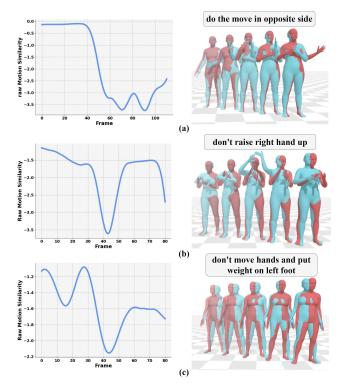

Figure 3. **Raw Motion Similarity**. We translate the global positions of sampled poses of the source motion and the edited motion for a clear view.

represent simple edits. Yet, S_i^R starts near 0 in (a) and near -1 in (b). Such variations are challenging for the model to learn. Second, when the edit is fine-grained, the subtle changes make the curve appear random. For example, in Fig. 3 (c), the hand movements corresponding to the text instructions "don't move hands and put weight on left foot" are subtle in the source motion, so the differences in the edited motion are minimal. As a result, the corresponding raw motion similarity curve lacks any identifiable patterns.

To address the first issue, we normalize all values in S^R to $[0, 1]$ with min-max normalization and make the curve scale-agnostic, as

$$S_i^N = \frac{S_i^r - \min_j S_j^R}{\max_j S_j^R - \min_j S_j^R}. \tag{6}$$

Our sliding window (Eq. (4)) and normalization (Eq. (6)) approaches also address the second issue. In addition, we filter out data samples that exhibit a sufficiently large semantic gap between the text instructions and the motion similarity from our model training, as these introduce unnecessary noise in the similarity curve. To this end, we define a motion signal-to-noise ratio (MotionSNR) based on the similarities of the top-κ and bottom-κ frames in a se-

quence, calculated as follows:

$$\mathbf{T}^R = \{S_k^R \mid S_k^R \geq S_j^R \quad \forall S_j^R \in \mathbf{S}^R - \mathbf{T}^R$$
$$\text{and } |\mathbf{T}^R| = \kappa\}, \quad (7)$$
$$\mathbf{B}^R = \{S_k^R \mid S_k^R \leq S_j^R \quad \forall S_j^R \in \mathbf{S}^R - \mathbf{B}^R$$
$$\text{and } |\mathbf{B}^R| = \kappa\}, \quad (8)$$
$$\text{MotionSNR} = \frac{\sum_{x \in \mathbf{T}^R} x}{\sum_{x \in \mathbf{B}^R} x}. \quad (9)$$

For our model training, we choose $\kappa = 5$.

Ideally, the motion for non-edited frames should remain the same as the source motion, while for edited frames, it should differ. Therefore, MotionSNR should be infinite for an ideal pair of source and edited motions. In practice, motion pairs with high MotionSNR should be easier for training, and the similarities should be easier to predict. Therefore, we filter out sequences with low MotionSNR using an empirically determined threshold. We verify the impact of this filtering approach in Sec. 4.2.

3.2.2. Auxiliary Loss Function

An underlying assumption of text-based motion editing is that the source motion and the text instructions should collectively predict the edited motion. By extension, we assume that the source motion and the text instructions should also collectively predict the motion similarity. Additionally, we note that motion similarity should contain sufficient information for motion editing. For instance, if the text instruction is "use both hands to catch the ball", motion similarity should identify which frames involve using only one hand to catch the ball and, therefore, need editing. We capture this notion of motion similarity through our auxiliary loss function. To implement this loss, instead of directly regressing the motion similarity curve, we quantize the values in S^N into K classes. Given motion similarity S_i^N, we use a quantization function $\mathcal{Q}(\cdot)$ as

$$\mathfrak{s}_i = \mathcal{Q}\left(S_i^N\right) := \begin{cases} 0, & S_i^N < \tau_0, \\ 1, & \tau_0 \leq S_i^N < \tau_1, \\ \vdots & \vdots \\ K-1, & S_i^N \geq \tau_{K-2}, \end{cases} \quad (10)$$

where $\mathfrak{s}_i \in \{0, \ldots, K-1\}$ provides the class label corresponding to each similarity level. We split the space $[0, 1]$ into K equal-length intervals for the similarity levels.

For each frame i, assuming the network outputs logits $\mathbf{z}_i = [z_{i,0}, \ldots, z_{i,K-1}]$, we convert logits to probabilities using the softmax function

$$p_{i,k} = \frac{\exp(z_{i,k})}{\sum_{l=0}^{K-1} \exp(z_{i,l})} \quad \forall k \in \{0, \ldots, K-1\}. \quad (11)$$

We can then write the cross-entropy loss for the classification task, which gives our auxiliary loss function, as

$$\mathcal{L}_{\text{aux}} = -\frac{1}{F} \sum_{i=0}^{F-1} \log p_{i,\mathfrak{s}_i} \quad (12)$$

Quantizing the similarity values provides the following benefit. For a given pairing of source motion, text instructions, and edited motion, the edited motion is only one of all feasible edited motions given just the source motion and the text instructions. Therefore, regression-based learning confines the model's ability to see only the provided edited motions as feasible. Quantization, by contrast, allows the model to focus on the coarser similarity levels rather than perfect frame-level similarity, striking a balance between optimizing the auxiliary and the editing losses and also enabling the model to generalize to a wider range of text instructions and motion variations than what is available from the dataset. We further validate this design choice experimentally in Tab. 3.

Lastly, combining the editing and the auxiliary losses (Eqs. (3) and (12)), we write the total training loss as

$$\mathcal{L} = \mathcal{L}_{\text{aux}} + \mathcal{L}_{\text{e}}. \quad (13)$$

3.3. Motion Diffusion Transformer

We combine diffusion-based motion editing and the auxiliary task of motion similarity prediction into our Motion Diffusion Transformer model Fig. 2. Our condition transformer follows the traditional transformer encoder architecture. It takes in the source motions and the CLIP features [42] of the text instructions, and outputs enhanced text features and motion features that contain sufficient information for the desired edited motions. This enhancement of the text and motion features via their intermixing is a direct consequence of using the auxiliary loss function (Eq. (12)). The diffusion transformer takes in the enhanced motion features concatenated with the noised edited motions. We also inject the enhanced text features into the diffusion transformer through an AdaLN-Zero layer, alongside the diffusion timestep t. We use the editing loss (Eq. (3)) for the model to learn the edited motions at the output of the diffusion transformer, and train it jointly with the auxiliary task. We adopt the standard DDPM diffusion paradigm [23] for both training and inference.

Our design ensures that the inputs to the auxiliary task are decoupled from the motion editing task. This is desirable since any changes in the noised edited motion (part of the diffusion transformer's inputs) should not affect the model's learning of the auxiliary task. Conversely, our design also ensures that the motion editing task fully leverages the enhanced motion and text features learned from the auxiliary task by directly using them as inputs for the denoising diffusion process.

4. Experiments and Results

We detail our experimental setup, including dataset, metrics, baselines, and implementation details, and elaborate on the results.

4.1. Setup

Data. The MotionFix dataset [4] is a language-based 3D motion editing dataset. Each data sample is a triplet consisting of a source motion, a target motion, and a text instruction. To ensure quality, semantically similar motion pairs are selected from motion capture (MoCap) data using the TMR motion embedding space [38]. Annotators then manually described the differences in a concise manner. MotionFix provides a total of 6,730 annotated triplets split into train, validation, and test sets.

Evaluation Metrics. For text-based human motion editing, we need to evaluate both the *semantic alignment* of the edited motions with the text instructions and the source motions, and the *fidelity* of the edited motions. Following the approach of MotionFix [4], we evaluate *alignment* based on motion-to-motion retrieval. We utilize pre-trained TMR [38] as a feature extractor and compute the retrieval accuracy of the generated motion in a fixed-size batch of motions. We report the top-1, top-2, and top-3 accuracies (R@1, R@2, and R@3) both in batch sizes of 32 and in the full test set. We also report the average rank for comparison. To evaluate *fidelity*, we utilize the MotionCritic score (M-score) [57]. M-score has been shown to align with human perceptual preferences, thus serving as a reasonable metric for motion realism. To compute M-scores, we use the authors' pre-trained model and transform our generated motions to match the pre-trained model's input.

Baselines. We compare our method with three baselines, MDM [53], MDM-BP [4], and TMED [4].
- MDM utilizes the pre-trained human motion diffusion model. The edit instruction is taken as text input.
- MDM-BP enhances MDM by introducing body-part detection using a GPT-based annotation system [2]. The method identifies body parts not mentioned in the text instructions, allowing the model to mask these parts and retain their source motion features.
- TMED is a conditional diffusion model specifically trained on the MotionFix dataset using motion-motion-text triplets. Unlike the baselines trained only on text-motion pairs, TMED leverages both the source motions and the text instruction as conditions to generate the desired edited motions.

Each of our baselines serves a specific role in evaluations. MDM represents the editing power of the diffusion model without the knowledge of source motions. MDM-BP knows both the source motions and the text instructions but is not trained on text-based motion editing datasets. TMED takes both source motions and text as input and is trained on the dataset, which is comparable to our approach, albeit with a different model architecture.

Implementation Details. We fix the DDPM [23] diffusion step to 300 and use the cosine noise scheduler. Following [4] and [7], we adapt two-way conditioning with a guidance scale of 2 for both the text and the source motion. We utilize pre-trained CLIP [42] from ViT-B/32 to encode the text. Our condition and diffusion transformers consist of 4 and 8 transformer encoder layers, respectively, 8 attention heads, and latent dimensions of 512. We train using a batch size of 128, and the AdamW optimizer [34] with a learning rate of 10^{-4}. We train our model for 1,500 epochs, which takes around a day on a single A100 GPU.

4.2. Quantitative Comparisons

We report quantitative comparisons in Tab. 1. Our model outperforms all baselines significantly in both motion-editing alignment and fidelity. MDM shows the weakest performance, aligning with the intuition that text instructions by themselves do not provide sufficient information to generate source-aligned edited motions. With GPT assistance in identifying edited body parts, MDM-BP shows substantial improvements over MDM. TEMD considers the influence of text instructions on the source motion and further outperforms MDM-BP. Our Motion Diffusion Transformer, together with the auxiliary loss for the learning of motion similarities, results in state-of-the-art performance. We attribute the progressive performance improvements to the knowledge of the source motion, effective dataset utilization, a specially designed network architecture, and an appropriate auxiliary loss function. We also evaluate our variant without filtering out noisy motion similarity curves for the auxiliary loss (w/o filtering – Tab. 1, last row). The performance downgrade implies that noisy data for motion similarity prediction negatively affects the training.

4.3. Qualitative Results

We show visual results compared with TMED [4] in Fig. 4. Our method improves on various generative artifacts of TMED, such as self-penetrations (row 1), failure to follow instruction nuances (row 2 – not generating motions with the right leg lifted, and row 3 – not generating the sitting down motion to be slower), and failure to maintain the effects of specific instructions (row 4 – lowering the hands instead of continuing to wave, row 5 – unexpectedly stepping forward, and row 6 – moving the arms beyond 90 degrees). These examples also illustrate the realism of our generated edited motions.

start earlier with arms already up

lift your right leg

sit down slower and don't get up

keep waving instead of lowering your hands

raise the hands sooner and don't step as far forward

move your arms only 90 degrees up

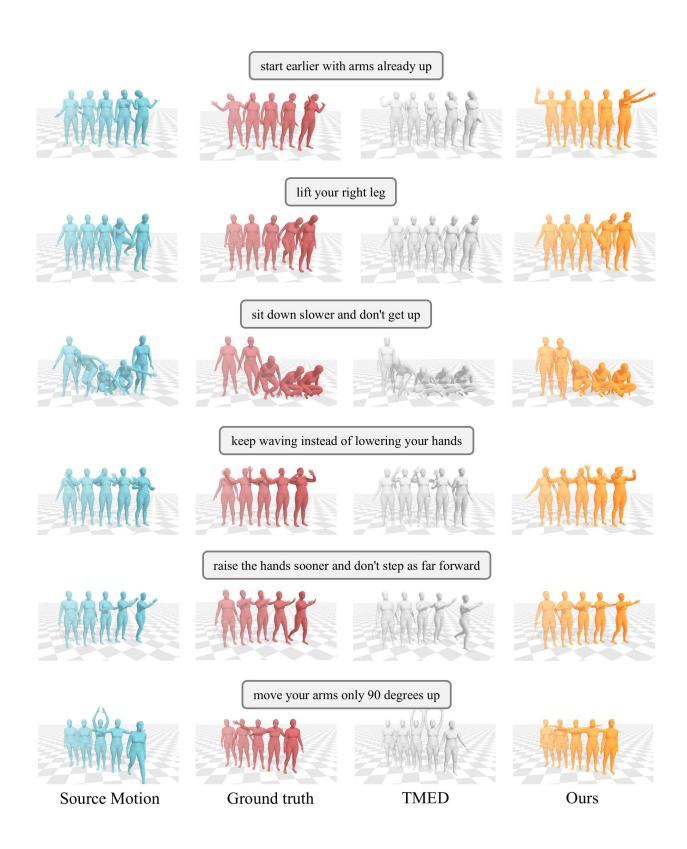Source Motion   Ground truth   TMED   Ours

Figure 4. **Qualitative Results.** We compare our method with TMED [4]. Our method outperforms TMED in terms of both fidelity and alignment with source motion and text instructions.

Table 1. **Comparison of Text-Based Motion Editing on the MotionFix [4] Dataset.** Our method outperforms all other baselines on generated-to-target retrieval and M-score. We report R@1, R@2 and R@3 as percentages. ↑ indicates higher values are better, ↓ indicates lower values are better, bold indicates **best**, and underline indicates <u>second best</u>.

Methods	Generated-to-Target (Batch)				Generated-to-Target (Test Set)				M-score↑
	R@1↑	R@2↑	R@3↑	AvgR↓	R@1↑	R@2↑	R@3↑	AvgR↓	
Ground Truth	100.0	100.0	100.0	1.00	64.36	88.75	95.56	1.74	−3.175
MDM [53]	4.03	7.56	10.48	15.55	0.10	0.10	0.10	–	–
MDM-BP [4]	39.10	50.09	54.84	6.46	8.69	14.71	18.36	180.99	–
TMED [4]	62.90	76.51	83.06	2.71	14.51	21.72	28.73	56.63	−3.512
Ours	**70.62**	**82.92**	**88.12**	**2.38**	**25.49**	**39.33**	**49.21**	**23.49**	**−3.210**
Ours (w/o aux loss)	<u>67.21</u>	78.75	<u>86.33</u>	2.49	23.32	<u>36.96</u>	<u>46.44</u>	26.02	−3.422
Ours (w/o filtering)	66.88	<u>80.42</u>	85.83	<u>2.46</u>	<u>23.73</u>	35.57	45.36	<u>24.94</u>	−3.418

Table 2. **Performances of SimMotionEdit with Different Combinations of Condition Features.** In the "no text feature" setting, the input to the condition transformer is the raw text. Using enhanced features for both text and motion leads to the best overall performance for motion realism.

Text Feat.	Motion Feat.	Generated-to-Target (Batch)			M-Score↑
		R@1↑	R@2↑	R@3↑	
Raw	Enhanced	70.42	81.25	86.67	−3.471
Raw	Raw	67.21	78.75	86.33	−3.422
Enhanced	Enhanced	<u>70.62</u>	<u>82.92</u>	<u>88.12</u>	**−3.210**
Enhanced	Raw	69.08	79.75	86.59	−3.450
Enhanced	No	66.46	81.25	84.38	−3.611
No	Enhanced	**71.04**	**83.96**	**89.58**	<u>−3.318</u>

Table 3. **Performances of SimMotionEdit with Different Auxiliary Losses.** Model performance improves when using a classification loss over a regression loss for the auxiliary task, indicating the better learning capacity enabled by classification. However, performance also decreases when increasing the number of classes beyond three, which brings the classes too close to each other.

Loss	#classes	Generated-to-Target (Batch)			M-Score↑
		R@1↑	R@2↑	R@3↑	
Reg.	∞	68.96	82.71	87.08	−3.750
Cls.	3	**70.62**	**82.92**	**88.12**	<u>−3.210</u>
Cls.	5	67.08	80.21	86.46	**−3.186**
Cls.	9	66.88	82.29	86.88	−3.722

4.4. Ablation Study

We perform ablations on both the Motion Diffusion Transformer and motion similarity prediction.

4.4.1. Motion Diffusion Transformer

We report comparisons with variations of our architecture in Tab. 2. The variant with both enhanced text and enhanced source motion features outperforms most of the other variants. It falls marginally behind on motion alignment and somewhat ahead on motion fidelity compared to using enhanced motion features with the raw text, indicating the strong influence of the source motion on aligning the edited motion but not having much influence on its fidelity. This outcome also suggests that the joint enhancement of both text and motion features provides a better balance between alignment and realism (fidelity). With these findings, we validate the effectiveness of enhanced features produced by the condition transformer and motion similarity prediction.

4.4.2. Motion Similarity Prediction

We report the results of ablating the auxiliary loss in Tab. 3. We find that quantizing the motion similarity values into

three classes yields the best results, implying that the auxiliary task should be simple enough to not overpower the main editing task. Further, a regression-based auxiliary loss reduces the motion fidelity given the overfitting tendencies when attempting to precisely predict the motion similarity curves.

5. Conclusion

We have presented SimMotionEdit, a state-of-the-art solution for text-based human motion editing. Our method utilizes the Motion Diffusion Transformer architecture to inject the source motion and the text instructions into a diffusion model for predicting the edited motions. We also leverage the novel auxiliary task of motion similarity prediction to further enhance the interplay of text and source motion features in the context of motion editing. Extensive experiments demonstrate that SimMotionEdit outperforms prior works in both alignment and realism of the edited motions. Through detailed ablation studies, we verify the effects of the core components of our method. We hope our work can push the limits of text-based human motion editing and inspire future research.

Acknowledgements

References

[1] Kfir Aberman, Yijia Weng, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Unpaired motion style transfer from video to animation. *ACM Transactions on Graphics (TOG)*, 39(4):64, 2020. 2

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6

[3] Sean Andrist, Bilge Mutlu, and Adriana Tapus. Look like me: matching robot personality via gaze to increase motivation. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 3603–3612, 2015. 1

[4] Nikos Athanasiou, Alpár Ceske, Markos Diomataris, Michael J. Black, and Gül Varol. MotionFix: Text-driven 3d human motion editing. In *ACM SIGGRAPH Asia*, 2024. 1, 2, 3, 4, 6, 7, 8, 12, 13

[5] Uttaran Bhattacharya, Elizabeth Childs, Nicholas Rewkowski, and Dinesh Manocha. Speech2affectivegestures: Synthesizing co-speech gestures with generative adversarial affective expression learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, page 2027–2036, New York, NY, USA, 2021. Association for Computing Machinery. 1

[6] Uttaran Bhattacharya, Nicholas Rewkowski, Abhishek Banerjee, Pooja Guhan, Aniket Bera, and Dinesh Manocha. Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents. In *VR*, pages 1–10, 2021. 1

[7] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, pages 18392–18402, 2023. 6

[8] Ling-Hao Chen, Wenxun Dai, Xuan Ju, Shunlin Lu, and Lei Zhang. Motionclr: Motion generation and training-free editing via understanding attention mechanisms. *arXiv preprint arXiv:2410.18977*, 2024. 1, 2

[9] 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 *CVPR*, pages 18000–18010, 2023. 2

[10] Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. Mofusion: A framework for denoising-diffusion-based motion synthesis. In *CVPR*, pages 9760–9770, 2023.

[11] Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. *ECCV*, 2024. 2

[12] Ginger Delmas, Philippe Weinzaepfel, Francesc Moreno-Noguer, and Grégory Rogez. Posefix: Correcting 3d human poses with natural language. In *ICCV*, pages 15018–15028, 2023. 2

[13] Yifeng Ding, Hantian Ding, Shiqi Wang, Qing Sun, Varun Kumar, and Zijian Wang. Horizon-length prediction: Advancing fill-in-the-middle capabilities for code generation with lookahead planning. *arXiv preprint arXiv:2410.03103*, 2024. 2, 3

[14] Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. Text-based motion synthesis with a hierarchical two-stream rnn. In *ACM SIGGRAPH 2021 Posters*, New York, NY, USA, 2021. Association for Computing Machinery. 1

[15] Anindita Ghosh, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek. Imos: Intent-driven full-body motion synthesis for human-object interactions. *Computer Graphics Forum*, 42(2):1–12, 2023. 1

[16] Anindita Ghosh, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek. Remos: 3d motion-conditioned reaction synthesis for two-person interactions. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXXVI*, page 418–437, Berlin, Heidelberg, 2024. Springer-Verlag. 2

[17] Michael Gleicher. Motion editing with spacetime constraints. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, page 139–ff., New York, NY, USA, 1997. Association for Computing Machinery. 2

[18] Michael Gleicher. Motion path editing. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, page 195–202, New York, NY, USA, 2001. Association for Computing Machinery. 2

[19] Purvi Goel, Kuan-Chieh Wang, C Karen Liu, and Kayvon Fatahalian. Iterative motion editing with natural language. In *ACM SIGGRAPH*, pages 1–9, 2024. 2, 4

[20] 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 *CVPR*, pages 5152–5161, 2022. 2

[21] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *ECCV*, pages 580–597, 2022. 1

[22] Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *CVPR*, pages 1900–1910, 2024. 1

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 2, 3, 5, 6

[24] Yiming Huang, Weilin Wan, Yue Yang, Chris Callison-Burch, Mark Yatskar, and Lingjie Liu. Como: Controllable motion generation through language guided pose code editing. In *Computer Vision – ECCV 2024*, pages 180–196, Cham, 2025. Springer Nature Switzerland. 2

[25] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016. 3

[26] Peng Jin, Yang Wu, Yanbo Fan, Zhongqian Sun, Wei Yang, and Li Yuan. Act as you wish: Fine-grained control of motion diffusion model with hierarchical semantic graphs. *NeurIPS*, 36, 2024. 1

[27] Jihoon Kim, Jiseob Kim, and Sungjoon Choi. Flame: Free-form language-based motion synthesis & editing. In *AAAI*, pages 8255–8263, 2023. 2

[28] Hema Swetha Koppula and Ashutosh Saxena. Anticipating human activities for reactive robotic response. In *IROS*, page 2071. Tokyo, 2013. 1

[29] Guillaume Lample and Devendra Singh Chaplot. Playing fps games with deep reinforcement learning. In *AAAI*, 2017. 2, 3

[30] Jehee Lee and Sung Yong Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, page 39–48, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 2

[31] Han Liang, Jiacheng Bao, Ruichi Zhang, Sihan Ren, Yuecheng Xu, Sibei Yang, Xin Chen, Jingyi Yu, and Lan Xu. Omg: Towards open-vocabulary motion generation via mixture of controllers. In *CVPR*, pages 482–493, 2024. 2

[32] Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. Intergen: Diffusion-based multi-human motion generation under complex interactions. *IJCV*, pages 1–21, 2024. 2

[33] 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. *NeurIPS*, 2023. 2

[34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[35] Boris N. Oreshkin, Florent Bocquelet, Félix G. Harvey, Bay Raitt, and Dominic Laflamme. Protores: Proto-residual network for pose authoring via learned inverse kinematics. In *ICLR*, 2022. 2

[36] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016. 1

[37] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 2

[38] Mathis Petrovich, Michael J Black, and Gül Varol. Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. In *ICCV*, pages 9488–9497, 2023. 6

[39] Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. Babel: Bodies, action and behavior with english labels. In *CVPR*, pages 722–731, 2021. 2

[40] Jia Qin, Youyi Zheng, and Kun Zhou. Motion in-betweening via two-stage transformers. *ACM Trans. Graph.*, 41(6), 2022. 2

[41] Sigal Raab, Inbar Gat, Nathan Sala, Guy Tevet, Rotem Shalev-Arkushin, Ohad Fried, Amit H Bermano, and Daniel Cohen-Or. Monkey see, monkey do: Harnessing self-attention in motion diffusion for zero-shot motion transfer. *ACM SIGGRAPH Asia*, 2024. 1, 2

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 5, 6

[43] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 4

[44] Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *CVPR*, 2023. 2

[45] Jiawei Ren, Mingyuan Zhang, Cunjun Yu, Xiao Ma, Liang Pan, and Ziwei Liu. Insactor: Instruction-driven physics-based characters. *NeurIPS*, 2023. 2

[46] S Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 3

[47] Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. Human motion diffusion as a generative prior. In *ICLR*, 2024. 2

[48] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265, 2015. 3

[49] Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics (TOG)*, 39(4):54–1, 2020. 1

[50] Kewei Sui, Anindita Ghosh, Inwoo Hwang, Jian Wang, and Chuan Guo. A survey on human interaction motion generation. *arXiv preprint arXiv:2503.12763*, 2025. 2

[51] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 1, 3

[52] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NeurIPS*, 30, 2017. 3

[53] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *ICLR*, 2023. 1, 2, 4, 6, 8

[54] Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *CVPR*, pages 448–458, 2023. 2

[55] Weilin Wan, Yiming Huang, Shutong Wu, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. Diffusionphase: Motion diffusion in frequency domain. *arXiv preprint arXiv:2312.04036*, 2023. 2

[56] Weilin Wan, Zhiyang Dou, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. Tlcontrol: Trajectory and language control for human motion synthesis. *ECCV*, 2024. 1

[57] Haoru Wang, Wentao Zhu, Luyi Miao, Yishu Xu, Feng Gao, Qi Tian, and Yizhou Wang. Aligning motion generation with human perceptions. In *ICLR*, 2025. 6

[58] Dong Wei, Xiaoning Sun, Huaijiang Sun, Shengxiang Hu, Bin Li, Weiqing Li, and Jianfeng Lu. Enhanced fine-grained motion diffusion for text-driven human motion synthesis. In *AAAI*, pages 5876–5884, 2024. 2

[59] Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicontrol: Control any joint at any time for human motion generation. In *ICLR*, 2024. 1, 2

[60] Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *ICCV*, pages 14928–14940, 2023. 2

[61] Wenjie Yin, Hang Yin, Kim Baraka, Danica Kragic, and Mårten Björkman. Dance style transfer with cross-modal transformer. In *WACV*, pages 5047–5056, 2023. 2

[62] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024. 3

[63] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *ICCV*, 2023. 2

[64] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6):4115–4128, 2024. 2

[65] Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. Finemogen: Fine-grained spatio-temporal motion generation and editing. *NeurIPS*, 36, 2024. 1, 2

[66] Mingyuan Zhang, Daisheng Jin, Chenyang Gu, Fangzhou Hong, Zhongang Cai, Jingfang Huang, Chongzhi Zhang, Xinying Guo, Lei Yang, Ying He, et al. Large motion model for unified multi-modal motion generation. In *ECCV*, pages 397–421. Springer, 2025. 2

[67] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34 (12):5586–5609, 2021. 1

[68] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 2, 3

SimMotionEdit: Text-Based Human Motion Editing with Motion Similarity Prediction

Supplementary Material

A. Perceptual Study

In addition to our quantitative and qualitative comparisons, we have also conducted a perceptual study with human participants to understand how human observers view the quality of motions edited by SimMotionEdit, particularly in comparison to the best available baseline TMED [4] and the corresponding ground truth edited motions.

A.1. Setup

Each participant in our study observed 10 sample sets corresponding to 10 randomly selected pairings of source motions and textual edit instructions. Within each sample set, a participant observed the source motion, the text instructions, and three versions of the edited motion — one each for the ground truth, SimMotionEdit, and TMED. We put these three edited motions in a random order within each sample set to avoid any positional biases in the study. We asked the participants to respond on two metrics:

- *Alignment of edited motions with text: How well does the edited motion follow the edit instructions, irrespective of the motion quality?* On this metric, the participants responded in a 3-point Likert Scale with the following descriptions: "Edited motion is completely different from the edit instructions" (1), "Edited motion only follows some parts of the edit instructions" (2), and "Edited motion fully follows the edit instructions" (3),

- *Plausibility of edited motions: How is the quality of the edited motion, irrespective of whether it follows the edit instructions?* On this metric, the participants responded in a 3-point Likert Scale with the following descriptions: "Edited motion has severe issues, *e.g.*, self-intersections, unrealistic poses or movements, *etc.* (1), "Edited motion has minor issues, *e.g.*, foot sliding, plausible but non-humanlike poses or movements, *etc.*" (2), and "Edited motion has no issues" (3).

We show an example layout with one source motion, one edit instruction, and one edited motion in Fig. A.1a and the scoring instructions and scoring area for the participants in Fig. A.1b.

A.2. Results

A total of 15 participants, consisting of students and staff from a University campus, responded to our perceptual study, leading to a total of 150 responses across the 10 sample sets. We show the distribution of participant scores aggregated over all the responses in Fig. A.2, and report the mean statistics in Tab. A.1. We observe that SimMotionEdit

Table A.1. **Perceptual Evaluation Mean Statistics.** We report the mean scores achieved by all three candidates in the perceptual study, averaging the aggregated responses across all the participants and sample sets. SimMotionEdit achieves scores that are 0.3 to 0.5 points higher than TMED on a 3-point Likert Scale.

Metric	Candidate	Mean Score ↑
Alignment	Ground Truth	2.53 ± 0.65
	SimMotionEdit (ours)	$\mathbf{2.17 \pm 0.78}$
	TMED [4]	1.70 ± 0.81
Plausibility	Ground Truth	2.79 ± 0.51
	SimMotionEdit (ours)	$\mathbf{2.37 \pm 0.70}$
	TMED [4]	1.99 ± 0.81

Table B.1. **Additional Performance and Efficiency Evaluations.** Compared to TMED, we report comparable generated-to-source accuracy measures and lower L@ distance and FID measures.

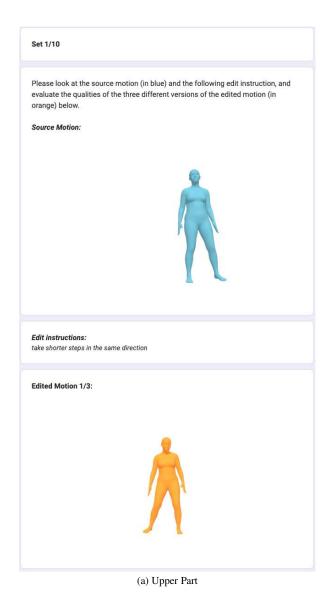| Method	Generated-to-Source (Batch)			L2 Dist.	FID
	R@1↑	R@2↑	R@3↑	(m)↓	↓
GT	74.01	84.52	89.91	–	–
TMED	71.77	**84.07**	**89.52**	0.278	0.167
Ours	**72.71**	83.54	87.50	**0.253**	**0.110**

is consistently scored higher than TMED [4], outperforming it by 40% to 50% on score distributions and by an absolute 0.3 to 0.5 points on the mean score on a 3-point Likert Scale across the two metrics.

B. Additional Quantitative Evaluations

For the preservation of the source motion, we follow TMED and use generated-to-source retrieval to measure motion preservation. Tab. B.1 shows the comparable performance of our method to TMED. For other fidelity-related metrics, since TMED does not provide the implementation of FID and L2 distance, we implement our own for a fair comparison. We see that our method outperforms TMED.

C. Additional Qualitative Results

As shown in Fig. C.1, our method can edit dance and crawling motions. It also successfully follows complex edit instructions.
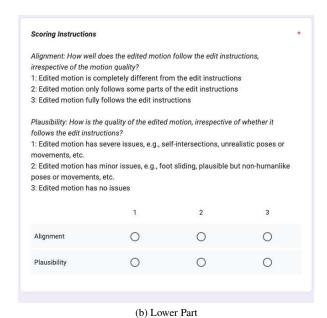
(a) Upper Part   (b) Lower Part

Figure A.1. **Perceptual Study Layout.** *(Upper Part)* We show an example of our study layout with one source motion, one edit instruction, and one edited motion. *(Lower Part)* We show the scoring instructions and scoring area for all the samples in the perceptual study.
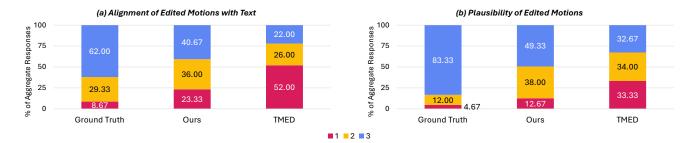

Figure A.2. **Perceptual Evaluation Score Distributions.** We show the distributions of aggregate responses from participants on the three versions of edited motions — Ground Truth, SimMotionEdit, and TMED [4] — on the two metrics of *Alignment* and *Plausibility*. We observe that participants have marked 3 for SimMotionEdit about 40% to 50% more times than TMED across the two metrics.
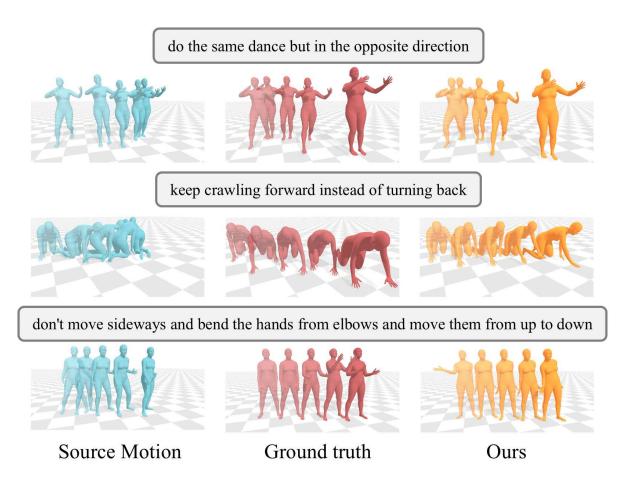
do the same dance but in the opposite direction

keep crawling forward instead of turning back

don't move sideways and bend the hands from elbows and move them from up to down

Source Motion   Ground truth   Ours

Figure C.1. **More Qualitative Results.**