TAME: TOPOLOGY-AWARE TEXT-DRIVEN MOTION EDITING ACROSS HETEROGENEOUS HUMANOID SKELETONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-driven motion editing modifies an existing motion sequence according to a text instruction while preserving the content of the source motion. Existing methods are typically built for a single, fixed skeletal topology, which limits their use in animation pipelines where characters differ in joint count and skeletal hierarchy. We present TOPOLOGY-AWARE MOTION EDITOR (TAME), a flow-matching transformer that edits motions on humanoid skeletons of varying topology. TAME represents motion as per-joint, per-frame tokens and models interactions among joints, across frames, and with the text instruction through skeletal, temporal, and text cross-attention layers. To make the skeletal attention follow each character's hierarchy, TAME replaces full joint attention with Topology-Constrained Skeletal Propagation (TCSP), which restricts attention to one-hop kinematic neighbors in the skeleton's adjacency matrix. We further introduce Edit-Focused Representation Alignment (EFRA), a self-distilled representation alignment strategy that aligns student features with cleaner EMA-teacher features exclusively on edit-relevant joint-time tokens, making edits faithful to the instruction. To make this setting trainable and comparable, we construct TOPOMOTIONFIX, a multi-topology extension of MotionFix with seen- and unseen-topology evaluation protocols. TAME outperforms previous methods in edit alignment and source preservation on MotionFix and reliably edits motions on unseen skeletons in TOPOMOTIONFIX.

1 INTRODUCTION

Character animation (Tevet et al., 2023; Zhang et al., 2024; Guo et al., 2024) is rarely completed in a single generation pass: an initial motion may not match the creator's intent due to the ambiguity of high-level language instructions. Motion editing (Athanasiou et al., 2024; Delmas et al., 2023) therefore plays a crucial role in animation workflows: it refines an existing motion according to a text instruction, such as raising an arm higher, while keeping unedited motion unchanged.

In animation production, a motion editor is expected to accommodate the varying skeletal topologies of different characters and to edit a motion faithfully according to the text instruction. Despite recent progress in text-driven motion editing (Li et al., 2025; Chen et al., 2024; Yang et al., 2025), existing methods fall short in both respects. First, most existing methods overlook the broader need of diverse character animation: editing motions across different skeletal topologies. Instead, they are designed for a single fixed topology (Athanasiou et al., 2024). However, in real-world production environments (Gat et al., 2025; Lee et al., 2023; Yang et al., 2024), human motion data commonly involves varying skeletal configurations beyond the canonical topology assumed by these methods. This topology mismatch limits the practical applicability of current motion editing methods. Second, existing methods struggle to realize the requested edit, so the edited motion is often misaligned with the instruction. For instance, when the required change is subtle, as in "stand without stretching leg," current methods tend to ignore it, whereas when the instruction is complex or abstract, as in "act as if violin is being played," the generated motion deviates substantially from the text semantics. Applying the requested edit while preserving the unedited motion remains an unsolved challenge.

To edit motion across different skeletal topologies while keeping each edit faithful to the instruction, we propose TAME (Topology-Aware Motion Editor), a flow-matching (Lipman et al., 2023) trans-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

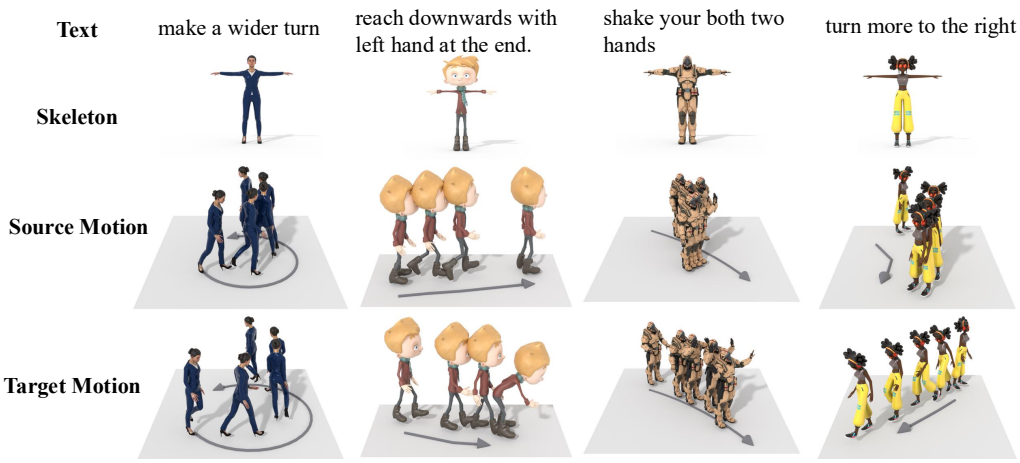

Figure 1: Text-driven motion editing across heterogeneous humanoid skeletons. TAME generates edited motions from a text instruction, a source motion, and the character's skeleton topology.

former for text-driven motion editing. Existing motion editing methods (Athanasiou et al., 2024; Li et al., 2025; Delmas et al., 2023; Chen et al., 2024) typically represent each pose as a single vector and focus mainly on temporal relationships while overlooking spatial interactions among skeletal joints. To address this limitation, TAME embeds motion as per-joint tokens at each frame, forming a skeleto-temporally structured representation that keeps the per-joint feature dimension fixed as the number of joints varies across skeletons. The network applies attention along the skeletal and temporal axes to model joint interactions and motion dynamics, while text cross-attention captures fine-grained interactions between the edit prompt and skeleto-temporal motion units. To handle motions on skeletons of varying topology, we introduce TOPOLOGY-CONSTRAINED SKELETAL PROPAGATION (TCSP) for skeletal interaction: given the input topology, TCSP permits one-step communication only between a joint and its physical neighbors, so that topology becomes a structural communication rule rather than a learnable bias in full attention. Long-range coordination remains possible but must emerge through multi-layer propagation along the character's kinematic chain.

To make edits faithful to the instruction, we propose Edit-Focused Representation Alignment (EFRA), which imposes representation guidance on the edited parts of a motion (Fig. 2). Our key insight is that motion editing is an asymmetric learning problem: the source motion provides a direct reference for the unedited parts, whereas the edited parts can only be inferred from its interaction with the text instruction. The edited parts therefore call for stronger guidance. Representation alignment lends itself to this purpose: recent work has shown that a diffusion transformer generates better when its intermediate features are aligned to a stronger representation, and that this representation can come not only from an external encoder (Yu et al., 2025) but also from the network itself, whose features grow more informative as the noise decreases and the depth increases (Jiang et al., 2025a). Accordingly, EFRA realizes this guidance by self-distillation and applies the alignment to the edited parts. It aligns an earlier-layer student feature under higher noise to a later-layer teacher feature under lower noise. The alignment is restricted to the edit-relevant joint-time tokens, identified during training by a DTW edit map derived from each source–target pair. The teacher is an EMA copy of the student and is the network deployed at inference, so EFRA adds no inference cost.

To train and evaluate TAME on heterogeneous humanoid skeletons, we construct TOPOMOTIONFIX, a multi-topology extension of MotionFix (Athanasiou et al., 2024) that retargets its source–target–instruction triplets onto a pool of humanoid rigs with different joint counts and hierarchies, and provides both seen- and unseen-topology evaluation protocols.

In summary, our contributions are fourfold. First, we propose TAME, to the best of our knowledge the first text-driven motion editor for heterogeneous humanoid skeletons, whose topology-constrained skeletal propagation (TCSP) turns the input topology into a structural communication rule of the network. Second, we introduce EFRA, a self-distillation strategy that strengthens guidance on the edited parts of a motion and thereby makes edits more faithful to the instruction. Third, we construct TOPOMOTIONFIX, a multi-topology benchmark built on MotionFix with seen- and unseen-topology

evaluation protocols. Fourth, TAME outperforms prior methods on MotionFix in both edit alignment and source preservation and generalizes robustly to unseen skeleton topologies in TOPOMOTIONFIX, with ablations validating the contributions of TCSP and EFRA.

2 RELATED WORK

2.1 TEXT-DRIVEN MOTION EDITING

Early 3D motion editing relied on handcrafted spatial-temporal constraints to manipulate trajectories (Witkin & Kass, 1988; Gleicher, 1997), where each edit demands manually specified rules and does not generalize to free-form text instructions. Motion editing methods including DNO (Karunratanakul et al., 2024), MoMo (Raab et al., 2024), MotionCLR (Chen et al., 2024), and SALAD (Hong et al., 2025) manipulate the attention or initial noise of pretrained motion diffusion models without further training; in the absence of paired editing supervision, however, these methods struggle to deliver fine-grained semantic edits while preserving the source motion. LLM-driven editors such as Iterative Motion Editing (Goel et al., 2024) and CoMo (Huang et al., 2024) translate text into symbolic operations applied to a frozen motion model; however, they rely on a textual description of the source motion as input rather than the motion data itself. MotionFix (Athanasiou et al., 2024) introduces the canonical triplet-based supervised setting (source motion, target motion, instruction) for text-driven 3D motion editing. Building on it, SimMotionEdit (Li et al., 2025) and PartMotionEdit (Yang et al., 2025) add motion similarity prediction as an auxiliary task at the frame and part level, respectively, while MotionReFit (Jiang et al., 2025b) blends source and target motions through the MotionCutMix augmentation; more recently, OmniME (Shi et al., 2026) adds negative supervision through a text–motion triplet loss, and Han & Kim (2026) fuse frame-wise and joint-wise motion features and predict joint-wise motion differences as an auxiliary task. These methods, however, are designed for a single fixed topology and struggle to realize the edit instruction faithfully. TAME addresses both limitations with a topology-aware editing architecture that generalizes across heterogeneous humanoid skeletons and a self-distilled representation alignment that provides more informative guidance on the edited regions.

2.2 SKELETON-AGNOSTIC AND ARBITRARY-TOPOLOGY MOTION MODELING

Existing methods modeling motion across diverse skeletal structures (Lee et al., 2023; Yang et al., 2024; Gat et al., 2025; Liu et al., 2025; Khani et al., 2025; Mo et al., 2025; Wang et al., 2026; Chen et al., 2025b; Xu et al., 2026) focus on generation, representation, or retargeting rather than editing. Early work such as SAME (Lee et al., 2023) learns a skeleton-agnostic motion embedding, and OmniMotionGPT (Yang et al., 2024) extends generation to animal skeletons. Recent generation works further diversify backbones and data scale, including AnyTop (Gat et al., 2025) with a Graphormer-style attention bias (Ying et al., 2021), UniMoGen (Khani et al., 2025), OmniSkel (Liu et al., 2025), PUMPS (Mo et al., 2025), X-MoGen (Wang et al., 2026), OmniZoo (Chen et al., 2025b), and NECromancer (Xu et al., 2026). A separate retargeting thread pioneered by skeleton-aware networks (Aberman et al., 2020) and Motion2Motion (Chen et al., 2025c) transfers a given motion across skeletons rather than editing it. Architecturally, prior methods such as AnyTop (Gat et al., 2025) inject topology through attention biases that allow long-range joint communication within a single layer. TAME instead enforces a hard one-hop adjacency mask (TCSP), so that distant joints interact only via multi-layer propagation along the kinematic chain.

2.3 REPRESENTATION ALIGNMENT IN DIFFUSION MODELS

Representation alignment has emerged as an effective auxiliary regulariser for diffusion transformer training. REPA (Yu et al., 2025) aligns the noisy hidden states of the denoiser with clean features from a pretrained visual encoder; follow-ups including MaskDiT (Zheng et al., 2024), SD-DiT (Zhu et al., 2024), TREAD (Krause et al., 2025), VA-VAE (Yao et al., 2025), and MAETok (Chen et al., 2025a) extend this to mask-prediction and tokenizer-level alignment with foundation models. SRA (Jiang et al., 2025a) removes the external dependence by self-distilling across noise levels, matching an earlier student feature under higher noise to a later EMA-teacher feature under lower noise. To the best of our knowledge, EFRA is the first to apply representation alignment to text-driven motion editing, through a token-wise dual-timestep schedule driven by a DTW-derived edit mask.

3 METHOD

3.1 OVERVIEW

Given a source motion $M_s \in \mathbb{R}^{T\times J\times D}$ defined on a skeleton structure $\mathcal{S}$ and a text editing instruction c, our goal is to generate an edited motion $M_t \in \mathbb{R}^{T\times J\times D}$ that semantically aligns with the instruction while preserving source motion content. The skeleton structure is represented as $\mathcal{S} = \{\mathcal{N}_\mathcal{S}, A_\mathcal{S}\}$, where $\mathcal{N}_\mathcal{S} = \{n_j\}_{j=1}^{J}$ is the set of joint names and $A_\mathcal{S} \in \{0,1\}^{J\times J}$ is the binary adjacency matrix, with $A_\mathcal{S}[i,j] = 1$ if and only if joints i and j are kinematic neighbors. As shown in Fig. 2, our method TAME has two components: a topology-aware flow-matching (Lipman et al., 2023; Liu et al., 2023; Hu et al., 2023) transformer with topology-constrained skeletal propagation (TCSP) that serves as the editing backbone (Section 3.2), and EFRA, an edit-focused representation alignment strategy (Section 3.3).

Motion representation. We represent a motion as a 3D tensor $M \in \mathbb{R}^{T\times J\times D}$, where T and J denote the frame and joint dimensions after padding for batching, and D is the number of motion features per joint. For each joint j, its feature vector $\phi_j \in \mathbb{R}^{12}$ is defined as

$$\phi_j = \begin{cases} \{p_j, r_j, u_j\} \in \mathbb{R}^{12}, & j \neq j_r, \\ \{\Delta p_r, r_r, h_r\} \in \mathbb{R}^{12}, & j = j_r, \end{cases}$$

where j_r denotes the root joint. For non-root joints, $p_j \in \mathbb{R}^3$ is the root-relative position, $r_j \in \mathbb{R}^6$ is the local joint rotation (Zhou et al., 2019), and $u_j \in \mathbb{R}^3$ is the velocity. For the root joint, $\Delta p_r \in \mathbb{R}^3$ is the pelvis translation, $r_r \in \mathbb{R}^6$ is the root orientation after removing the heading rotation, and $h_r = (\sin\delta\theta, \cos\delta\theta, z_{\text{root}}) \in \mathbb{R}^3$, where $\delta\theta$ is the heading change and z_{root} is the pelvis height.

3.2 TOPOLOGY-AWARE MOTION EDITOR BACKBONE

Given the inputs M_s, c, $\mathcal{S}$ defined in Section 3.1 and a noised target motion $\tilde{M}_t$, the flow-matching transformer of TAME predicts the rectified-flow velocity $\hat{v}_\theta \in \mathbb{R}^{T\times J\times D}$ that drives the training objective in Section 3.3. As illustrated in Fig. 2, the architecture has two parts: an input embedding stage, and a stack of L identical TAME *blocks*.

Input embedding. We embed each frame of the source motion and the noised target motion as per-joint features. Since the non-root feature group $\{p_j, r_j, u_j\}$ and the root feature group $\{\Delta p_r, r_r, h_r\}$ have different semantics, we use separate projections for root and non-root joints. Joint identity is provided by a T5 text encoder applied to the joint names; for each joint j, the resulting embedding $e_{\mathcal{S},j} \in \mathbb{R}^d$ is added to the joint token at every frame. To distinguish source from target along the time axis and mark their boundary, we add learnable source/target segment embeddings and insert a separator token $M_{\text{sep}} \in \mathbb{R}^{1\times J\times d}$, yielding

$$\bar{M} = [\bar{M}_s; M_{\text{sep}}; \bar{M}_t] \in \mathbb{R}^{(2T+1)\times J\times d}.$$

TAME block with topology-constrained skeletal propagation. Each TAME block contains three sequential attention sub-layers operating on the same joint-time tokens along three axes: joint, time, and text instruction. The joint-axis sub-layer applies our topology-constrained skeletal propagation (TCSP), in which self-attention is restricted to one-hop kinematic neighbors. We achieve this by adding a mask $H_\mathcal{S}$ to the pre-softmax attention scores, with 0 for neighbor pairs and $-\infty$ for non-neighbors, so that non-neighbor attention is suppressed to zero after softmax:

$$H_\mathcal{S}[i,j] = \begin{cases} 0, & \text{if } i = j \text{ or } A_\mathcal{S}[i,j] = 1, \\ -\infty, & \text{otherwise,} \end{cases}$$

so that for each attention head,

$$\text{TCSP}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_h}} + H_\mathcal{S}\right)V,$$

where d_h is the head dimension. Information between distant joints reaches them only through stacked layers along the kinematic chain.

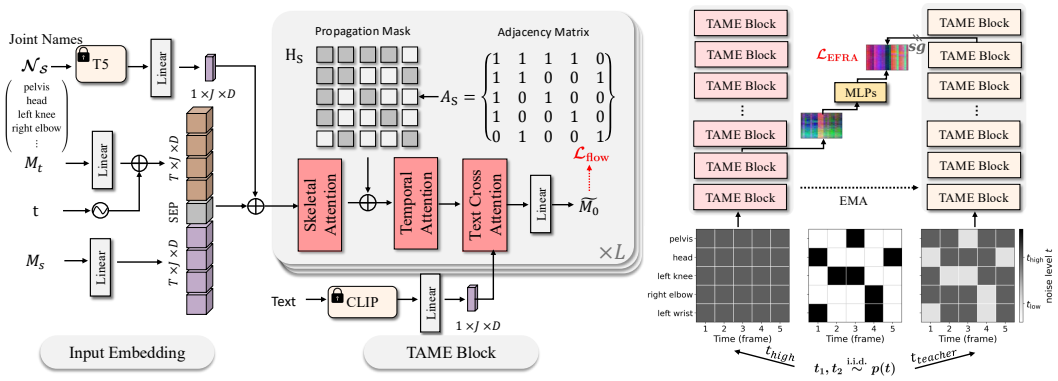

Figure 2: Overview of TAME. **Left:** the topology-aware motion editor backbone embeds the source motion M_s and noised target motion $\tilde{M}_t$ as joint-time tokens and applies L layers of topology-constrained skeletal propagation (TCSP), temporal self-attention, and text cross-attention to predict the velocity field $\hat{v}_\theta$. **Right:** the Edit-Focused Representation Alignment (EFRA) branch, used only in training. A DTW edit map E marks the edit-relevant tokens and drives edit-aware dual-timestep scheduling, which makes the EMA teacher cleaner than the student on those tokens. Student and teacher features are then aligned only on those tokens.

The time-axis sub-layer applies standard temporal self-attention, allowing each target joint to attend to its corresponding source-joint trajectory. The third sub-layer conditions motion tokens on the text instruction via cross-attention, with motion tokens as queries and CLIP-encoded text tokens as keys and values. After L blocks, we discard the source and separator segments, apply a final normalisation, and use separate output heads for root and non-root joints to produce $\hat{v}_\theta \in \mathbb{R}^{T\times J\times D}$.

3.3 Flow Matching with Edit-Focused Representation Alignment

Our design starts from a key observation: motion editing is an asymmetric learning problem. The unedited parts of the target closely follow the source motion, which serves as a reliable reference for them. The edited parts, in contrast, have to be inferred from the source motion and the text instruction, and therefore call for stronger guidance. We obtain this guidance from self-representation alignment (SRA) (Jiang et al., 2025a), which builds on the observation that the intermediate features of a diffusion transformer become more informative as the noise decreases and the depth increases, and therefore aligns the weaker features of the network to its better ones without any external representation component. Accordingly, EFRA aligns an earlier-layer student feature under higher noise to a later-layer teacher feature under lower noise, where the teacher is an EMA copy of the student (Fig. 2, upper right). Rather than applying the alignment to the whole target motion, EFRA restricts it to the edit-relevant joint-time tokens, identified by a DTW edit map derived from each source–target pair (lower centre). The noise levels of the student and the teacher are assigned by edit-aware dual-timestep scheduling: the student takes a globally higher noise (lower left), and the teacher takes the lower noise only on those tokens (lower right). Below we describe the three parts in turn: the DTW edit map, the edit-aware dual-timestep scheduling, and the alignment objective.

DTW edit map. Restricting the alignment to the edited parts requires supervision on where the edit takes place, which can be recovered by comparing the source motion with the target motion. Prior work (Li et al., 2025; Yang et al., 2025) derives such supervision only at the frame or body-part level, too coarse for joint-time tokens. To obtain a skeletal-temporally structured edit signal, we use *Dynamic Time Warping* (DTW) (Sakoe & Chiba, 1978) to align source and target frames, read out joint-level distances along the alignment, and threshold them after per-sample min-max normalisation into a binary edit mask $E \in \{0,1\}^{T\times J}$, where $E[i,j]=1$ marks the joint-time tokens to be edited. The mask is precomputed offline for each source–target pair. The full construction algorithm and per-step details are deferred to Appendix C.

Edit-aware dual-timestep scheduling. To confine the teacher's guidance to the edited parts, we restrict the information asymmetry between the teacher and the student to the edit tokens. We realize

this by decoupling the student's and teacher's timesteps token-wise and injecting the edit mask E into the teacher's schedule, so that the teacher receives a lower noise level than the student only on edit tokens. Given two i.i.d. timesteps $t_1, t_2 \sim p(t)$ on $[0, 1]$, we order them as $t_{\text{low}} = \min(t_1, t_2)$ and $t_{\text{high}} = \max(t_1, t_2)$, and define the token-wise teacher timestep

$$t_{\text{teacher}}[i, j] = \begin{cases} t_{\text{low}}, & \text{if } E[i, j] = 1, \\ t_{\text{high}}, & \text{otherwise.} \end{cases}$$

The student is paired with t_{high} uniformly across all target tokens (Fig. 2, lower left), while the teacher follows t_{teacher} (lower right). Under the rectified-flow noise schedule with a shared noise $\epsilon \sim \mathcal{N}(0, I)$, the noised inputs are

$$\tilde{M}_t = (1 - t_{\text{high}})M_t + t_{\text{high}}\epsilon, \quad \tilde{M}_t^{\text{teacher}} = (1 - t_{\text{teacher}})M_t + t_{\text{teacher}}\epsilon,$$

where t_{teacher} is broadcast element-wise across the feature axis.

Self-distilled feature alignment. Finally, the alignment objective follows SRA in pairing an earlier student layer l_s with a later teacher layer l_t, so that the teacher is ahead of the student in both noise level and processing depth. The student and the teacher receive the same source motion, text instruction, and skeleton, and differ only in their noised input and their parameters: the teacher parameters θ' are an EMA copy of the student parameters θ, updated after every optimiser step as $\theta' \leftarrow \alpha\theta' + (1 - \alpha)\theta$ with momentum α (Jiang et al., 2025a). For each target joint-time token (i, j), we take the output of the l_s-th block of the student, $H^{l_s}_{\theta,i,j} \in \mathbb{R}^d$, and the output of the l_t-th block of the teacher, $H^{l_t}_{\theta',i,j} \in \mathbb{R}^d$. The student feature is mapped by a two-layer MLP projection head P_ψ and aligned to the teacher feature, which is held fixed by the stop-gradient operator $\text{sg}(\cdot)$, with the cosine distance $d_{\cos}$ over the edit tokens only (Fig. 2, upper right):

$$\mathcal{L}_{\text{EFRA}} = \mathbb{E}\left[\frac{1}{|\Omega_E|}\sum_{(i,j)\in\Omega_E} d_{\cos}\left(P_\psi(H^{l_s}_{\theta,i,j}), \text{sg}(H^{l_t}_{\theta',i,j})\right)\right],$$

where $\Omega_E = \{(i, j) \in \Omega \mid E[i, j] = 1\}$ with Ω the set of valid target joint-time tokens, and $d_{\cos}(a, b) = 1 - \frac{a^\top b}{\|a\|_2\|b\|_2}$. Combined with the standard rectified-flow objective on the student velocity prediction $\hat{v}_\theta$ under the source, text, and skeleton conditions, with target $v^* = \epsilon - M_t$,

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\left[\frac{1}{|\Omega|}\sum_{(i,j)\in\Omega} \left\|\hat{v}_{\theta,i,j} - v^*_{i,j}\right\|_2^2\right],$$

we obtain the final training objective

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_{\text{EFRA}}\mathcal{L}_{\text{EFRA}},$$

with weighting hyperparameter λ_{EFRA}. The alignment is fully self-distilled: the only added learnable parameter is the projection head P_ψ, which is discarded at inference. At inference, we deploy the EMA-teacher copy of the backbone (rather than the student), so EFRA adds no inference cost.

4 Experiments

We start by introducing the multi-topology dataset TopoMotionFix (Section 4.1), together with our evaluation metrics and seen/unseen-topology protocols (Section 4.2). We then present the main quantitative results, comparing TAME with state-of-the-art motion-editing methods (Section 4.3). Next, we provide ablations on the key components of TAME (Section 4.4). Finally, we demonstrate qualitative results on both seen and unseen topologies (Section 4.5).

4.1 TopoMotionFix Dataset

TopoMotionFix comprises three components: textual edit instructions, source and target motion sequences, and the corresponding skeletal structures. Building on MotionFix (Athanasiou et al., 2024), we retarget its motion-skeleton pairs onto a pool of 83 humanoid rigs sourced from LAFAN1, ACCAD, TotalCapture, PFNN, and SFU motion-capture datasets, augmented with skeleton randomisation following (Lee et al., 2023; Liu et al., 2025); the full construction procedure is detailed in Appendix A. TopoMotionFix contains 25,794 distinct skeletal configurations, 40,304 unique text-motion-skeleton triplets, and 1,416 unique textual edit instructions, inheriting MotionFix's train/test partition.

Table 1: Quantitative comparison. ↑ indicates higher values are better, ↓ indicates lower values are better; bold indicates best; dash indicates unavailable.

Dataset	Protocol	Method	Generated-to-Target (Batch-32)				Generated-to-Source (Batch-32)			
			R@1↑	R@2↑	R@3↑	AvgR↓	R@1↑	R@2↑	R@3↑	AvgR↓
		GT	100.00	100.00	100.00	1.00	74.01	84.52	89.91	2.03
MotionFix	Seen	MDM (Tevet et al., 2023)	4.03	7.56	10.48	15.55	2.62	6.15	9.38	15.88
MotionFix	Seen	MDM-BP (Athanasiou et al., 2024)	39.10	50.09	54.84	6.46	61.28	69.55	73.99	4.21
MotionFix	Seen	TMED (Athanasiou et al., 2024)	64.11	76.51	82.56	2.72	73.19	85.08	91.03	1.92
MotionFix	Seen	MotionReFit (Jiang et al., 2025b)	66.33	80.05	84.98	2.64	83.47	90.42	92.84	1.73
MotionFix	Seen	SimMotionEdit (Li et al., 2025)	70.62	82.92	88.12	2.38	72.71	83.54	87.50	–
MotionFix	Seen	Han & Kim (2026)	74.38	88.54	92.08	1.92	78.96	91.04	93.33	–
MotionFix	Seen	OmniME (Shi et al., 2026)	77.29	88.54	91.88	1.79	–	–	–	–
MotionFix	Seen	TAME	**78.75**	**91.67**	**94.79**	**1.55**	**86.04**	**94.58**	**97.50**	**1.36**
TopoMotionFix	Unseen (SMPL+held-out rigs)	AnyTop (Gat et al., 2025)	68.85	83.87	89.72	2.01	84.17	92.54	95.26	1.42
TopoMotionFix	Unseen (SMPL+held-out rigs)	TAME w/o TCSP, w/o EFRA	67.54	81.55	87.10	2.13	82.06	91.73	95.36	1.51
TopoMotionFix	Unseen (SMPL+held-out rigs)	TAME w/o EFRA	71.67	85.48	91.23	1.97	87.00	93.75	96.88	1.39
TopoMotionFix	Unseen (SMPL+held-out rigs)	TAME	**72.78**	**86.69**	**91.43**	**1.95**	**90.02**	**95.46**	**97.68**	**1.29**

4.2 Evaluation Setup

Evaluation protocols. We define two evaluation protocols, both using the test pairs and the evaluator provided by MotionFix (Athanasiou et al., 2024). In the *seen-topology* protocol, we train on MotionFix's training set with only the input representation converted, enabling fair comparison with prior fixed-topology motion-editing methods. In the *unseen-topology* protocol, the test set uses 10 rigs, the SMPL skeleton and 9 held-out rigs, while the training set uses 83 rigs. The held-out rigs are excluded from training and from the skeleton randomisation of Algorithm 1, adapted from SAME (Lee et al., 2023), so their topologies are fixed and unseen. The construction of the held-out test sets and the evaluation of motions generated on non-SMPL skeletons are detailed in Appendix B.

Evaluation metrics. We evaluate the semantic alignment of edited motions with both the text instruction and the source motion. Following MotionFix (Athanasiou et al., 2024), alignment is measured via motion-to-motion retrieval using pretrained TMR as a feature extractor. We report R@1, R@2, R@3 and average rank (AvgR) on both generated-to-target ($g \to t$) and generated-to-source ($g \to s$) retrieval, with a fixed candidate batch of 32 test-set motions (fixed across runs to eliminate randomness), and under the test-set setting (deferred to Appendix D).

Baselines. For the seen-topology protocol, we compare TAME with current motion-editing methods, including MDM (Tevet et al., 2023), MDM-BP and TMED (Athanasiou et al., 2024), MotionReFit (Jiang et al., 2025b), SimMotionEdit (Li et al., 2025), OmniME (Shi et al., 2026), and the cross-axis fusion model of Han & Kim (2026). For the unseen-topology protocol, no prior text-driven motion editing method operates on arbitrary skeletons; to validate the effectiveness of our method, we adapt AnyTop (Gat et al., 2025), an arbitrary-topology motion generation framework, by adapting its topological conditioning scheme into our TAME editing pipeline.

Implementation details. We adopt rectified flow matching (Liu et al., 2023; Lipman et al., 2023) with 50 sampling steps at inference on a single NVIDIA RTX A6000 (48 GB) GPU, 9 TAME blocks, and a latent dimension of 512. Each model is trained on 2 NVIDIA RTX Pro 6000 (96 GB) GPUs for approximately one day. More implementation details can be found in Appendix E.

4.3 Main Results

Quantitative comparisons. We report quantitative comparisons in Table 1; test-set retrieval results are reported in Table 3 (Appendix D). On the seen-topology setting, TAME outperforms all fixed-topology baselines on both edit alignment and source preservation. We attribute this to edit-focused representation alignment (EFRA), which sharpens supervision on edit-relevant joint-time tokens via a dual-timestep self-distillation gap, and the per-joint motion representation, which preserves source content more faithfully than canonical pose vectors used by prior baselines. On the unseen-topology setting, TAME surpasses the AnyTop-adapted baseline by $+3.93$ batch R@1, demonstrating robustness to topologies unseen during training.

Table 2: Ablations on MotionFix. The first two rows ablate TCSP (TAME w/o TCSP, w/o EFRA vs. TAME w/o EFRA); blocks (a)-(c) keep TCSP and EFRA enabled and vary one hyperparameter: PH (projection-head dimension), Block (student/teacher layer pair (l_s, l_t)), Edit Mask (✓/×: with/without the DTW edit mask). Bold marks our default. ↑/↓ indicates higher/lower is better.

				Generated-to-Target (Batch-32)				Generated-to-Source (Batch-32)			
	PH	Block	Edit Mask	R@1↑	R@2↑	R@3↑	AvgR↓	R@1↑	R@2↑	R@3↑	AvgR↓
TAME w/o TCSP, w/o EFRA	—	—	—	74.80	88.00	92.64	1.80	84.38	92.64	95.16	1.52
TAME w/o EFRA	—	—	—	74.70	87.50	90.62	1.85	84.17	92.34	95.06	1.51
(a) Projection-head dimension	256	4/7	✓	76.11	89.11	93.45	1.66	84.27	92.84	94.86	1.53
	512	4/7	✓	77.32	88.31	92.74	1.74	84.48	92.44	94.76	1.59
	1024	**4/7**	✓	**78.75**	**91.67**	**94.79**	**1.55**	**86.04**	**94.58**	**97.50**	**1.36**
(b) Alignment block layers (l_s, l_t)	1024	2/6	✓	76.01	89.01	93.15	1.70	83.47	92.24	94.66	1.61
	1024	3/7	✓	77.22	89.72	93.45	1.60	82.86	91.43	94.46	1.58
	1024	3/8	✓	76.21	88.21	92.94	1.65	81.35	91.63	94.35	1.60
	1024	**4/7**	✓	**78.75**	**91.67**	**94.79**	**1.55**	**86.04**	**94.58**	**97.50**	**1.36**
(c) DTW edit mask	**1024**	**4/7**	✓	**78.75**	**91.67**	**94.79**	**1.55**	**86.04**	**94.58**	**97.50**	**1.36**
	1024	4/7	×	75.71	87.70	92.54	1.74	82.06	91.33	94.56	1.64

4.4 Ablation Studies

TCSP's effectiveness for heterogeneous topologies is established by Table 1. Here, in Table 2, we further validate EFRA and ablate its hyperparameters on seen-topology MotionFix.

Effects of TCSP and EFRA. We ablate the components of TAME in Table 2. The first two rows isolate TCSP: on this single-topology setting, the gap between TAME w/o EFRA and TAME w/o TCSP, w/o EFRA is only -0.10 edit R@1, indicating that the adjacency prior is dispensable on a fixed skeleton. In contrast, in the unseen-topology setting TAME surpasses AnyTop's content-dependent bias by $+3.93$ edit R@1 (Table 1), indicating that the hard topology constraint matters once the input topology varies. TAME further surpasses TAME w/o EFRA by $+4.05$ edit R@1, together with consistent gains on source preservation, validating EFRA as the key design choice behind our edit alignment.

EFRA hyperparameters. For the projection-head dimension, d_p=256 already lifts edit R@1 by $+1.41$ over TAME w/o EFRA, confirming that even a narrow alignment head injects useful self-distillation signal, and R@1 continues to climb through d_p=1024. For the student/teacher layer pair (l_s, l_t), we sweep four configurations covering shallower SRA ratios and our deeper default; the mid-stack student to near-final teacher pair $(4, 7)$ is best, while pulling the student too early (s=2) or the teacher to the last layer (t=8) drops edit R@1 by $\sim$2.5–2.7, matching SRA's pattern. Removing the DTW edit mask reduces edit R@1 by -3.04 and source R@1 by -3.98: aligning over trivially preserved frames dilutes the learning signal on the tokens that actually carry the editing intent.

4.5 Qualitative Results

TAME avoids the editing failures that TMED, SimMotionEdit and OmniME exhibit in Fig. 3: ignoring explicit prohibitions (column 1: TMED raises the arm in keyframes 4–8 and OmniME in keyframes 4–6, while SimMotionEdit obeys the instruction but loses the source's slow forward bend), distorting the source action when applying speed edits (column 2: TMED only copies the source endpoints, SimMotionEdit advances the onset to keyframe 2 but misreads the single-arm raise as a bilateral motion, and OmniME advances the onset only to keyframe 4 and lets the hand drift to the ear instead of the chest), and deviating from the text semantics under complex or abstract instructions (column 3: SimMotionEdit ignores the violin instruction, TMED merely repeats the source's first half, and OmniME raises both arms overhead instead of forming the violin pose).

The unseen-topology examples in Fig. 4 further demonstrate that TAME does not simply memorize skeleton layouts from training. When evaluated on Mixamo skeletons unseen during training, our model still produces plausible local edits and preserves the overall source motion trajectory. In contrast, baselines more frequently introduce topology-dependent artifacts, including unnatural limb trajectories, over-smoothed edits, or unintended changes to unedited joints. These examples

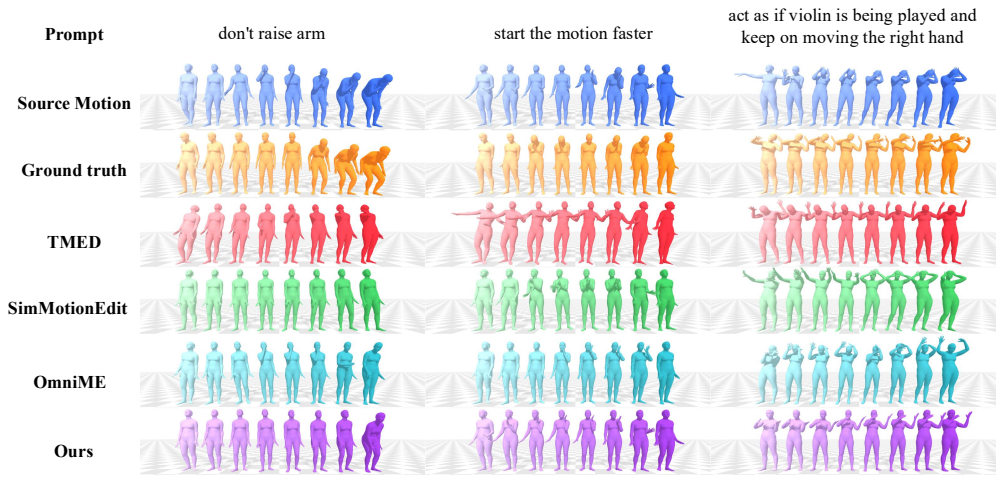

Figure 3: Qualitative comparison on the seen-topology setting. Columns are edit prompts; rows are source, ground truth, TMED (Athanasiou et al., 2024), SimMotionEdit (Li et al., 2025), OmniME (Shi et al., 2026), and our TAME. Each cell shows 8 keyframes (light to dark in time).

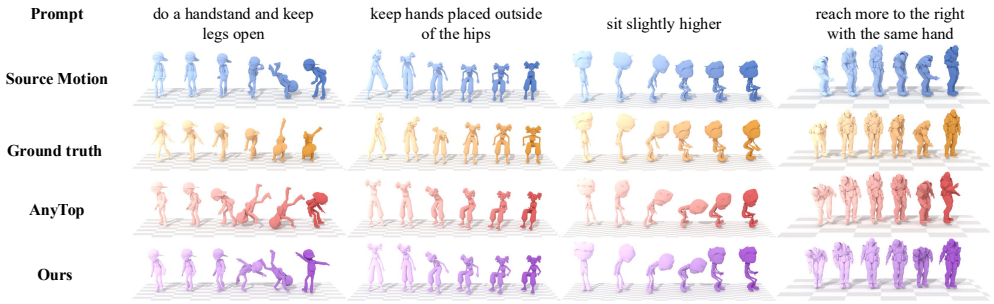

Figure 4: Qualitative comparison on the unseen-topology setting, evaluating TopoMotionFix-trained models on Mixamo skeletons. Columns are edit prompts; rows are source, ground truth, AnyTop (Gat et al., 2025), and our TAME. Each cell shows 6 keyframes (light to dark in time).

complement the quantitative results by showing that the gains in retrieval correspond to visually coherent, instruction-faithful edited motions.

5 Conclusion

We studied text-driven motion editing across heterogeneous humanoid skeletons, a setting that better reflects the diverse character data encountered in real animation pipelines. To make the setting trainable and evaluable, we constructed TopoMotionFix, a multi-topology extension of MotionFix with seen- and unseen-topology evaluation protocols. We proposed TAME, a topology-aware flow-matching transformer that represents motion as joint-time tokens and uses topology-constrained skeletal propagation (TCSP) to confine joint attention to one-hop kinematic neighbors, so that the model operates natively on skeletons of varying topology. Since the edited parts of a motion have to be inferred from the source motion and the instruction and therefore call for stronger guidance, we further introduced EFRA, a self-distilled representation alignment strategy that aligns the student to a cleaner EMA teacher only on the edit tokens identified by a DTW edit map. On MotionFix, TAME outperforms fixed-topology editing methods on both edit alignment and source preservation. On TopoMotionFix, it generalizes robustly to skeleton topologies unseen during training, bringing text-driven motion editing to the varied characters of animation production.

AI USE STATEMENT

In this work, we did not use generative AI tools for tasks requiring mandatory disclosure. We did not use generative AI tools to develop theoretical models, formulate mathematical claims, design experiments, implement methods, or interpret experimental results, and the remaining mandatory-disclosure tasks are not applicable to this work. We used generative AI tools for code review, debugging, and minor code editing, as well as to proofread the manuscript for grammar, clarity, and readability. All AI-assisted work was reviewed by the authors. AI-assisted code modifications were tested for correctness, and suggested manuscript revisions were verified to preserve the technical meaning and accuracy of the paper. The authors take full responsibility for the final content of this work, including all text and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our perceptual study (Appendix F) involved voluntary adult participants who rated anonymised motion clips; no personal data was collected. TOPOMOTIONFIX is derived from a publicly available motion editing dataset and publicly released character rigs, and will be released under the same terms as its sources. The method edits skeletal motion data only and does not involve identifiable human appearance.

REPRODUCIBILITY STATEMENT

Implementation details of TAME, including the architecture, training configuration, and all EFRA hyperparameters, are given in Appendix E. The TOPOMOTIONFIX construction procedure and the DTW edit-map construction are specified with pseudocode in Appendices A and C. The evaluation protocols and metrics follow the public MotionFix test set and evaluator, as described in Section 4.2 and Appendix D. The unseen-topology protocol, including the construction of the held-out test set and the evaluation of motions generated on non-SMPL skeletons, is detailed in Appendix B.

REFERENCES

Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (ToG)*, 39(4):62–1, 2020.

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*, pp. 1–11, 2024.

Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. In *Forty-second International Conference on Machine Learning*, 2025a.

Keyi Chen, Mingze Sun, Zhenyu Liu, Zhangquan Chen, and Ruqi Huang. Topology-agnostic animal motion generation from text prompt. *arXiv preprint arXiv:2512.10352*, 2025b.

Ling-Hao Chen, Wenxun Dai, Xuan Ju, Shunlin Lu, and Lei Zhang. Motionclr: Motion generation and training-free editing via understanding attention mechanisms. *arxiv:2410.18977*, 2024.

Ling-Hao Chen, Yuhong Zhang, Zixin Yin, Zhiyang Dou, Xin Chen, Jingbo Wang, Taku Komura, and Lei Zhang. Motion2motion: Cross-topology motion transfer with sparse correspondence. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025c.

Ginger Delmas, Philippe Weinzaepfel, Francesc Moreno-Noguer, and Grégory Rogez. Posefix: Correcting 3d human poses with natural language. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15018–15028, 2023.

Inbar Gat, Sigal Raab, Guy Tevet, Yuval Reshef, Amit Haim Bermano, and Daniel Cohen-Or. Anytop: Character animation diffusion with any topology. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–10, 2025.

Michael Gleicher. Motion editing with spacetime constraints. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pp. 139–ff, 1997.

Purvi Goel, Kuan-Chieh Wang, C Karen Liu, and Kayvon Fatahalian. Iterative motion editing with natural language. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–9, 2024.

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*, pp. 1900–1910, 2024.

Gyojin Han and Junmo Kim. Cross-axis feature fusion with joint-wise motion difference prediction for text-based 3d human motion editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30618–30628, June 2026.

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 Computer Vision and Pattern Recognition Conference*, pp. 7158–7168, 2025.

Vincent Tao Hu, Wenzhe Yin, Pingchuan Ma, Yunlu Chen, Basura Fernando, Yuki M Asano, Efstratios Gavves, Pascal Mettes, Bjorn Ommer, and Cees GM Snoek. Motion flow matching for human motion synthesis and editing. *arXiv preprint arXiv:2312.08895*, 2023.

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 *European Conference on Computer Vision*, pp. 180–196. Springer, 2024.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. *arXiv preprint arXiv:2505.02831*, 2025a.

Nan Jiang, Hongjie Li, Ziye Yuan, Zimo He, Yixin Chen, Tengyu Liu, Yixin Zhu, and Siyuan Huang. Dynamic motion blending for versatile motion editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22735–22745, 2025b.

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*, pp. 1334–1345, 2024.

Aliasghar Khani, Arianna Rampini, Evan Atherton, and Bruno Roy. Unimogen: Universal motion generation. *arXiv preprint arXiv:2505.21837*, 2025.

Felix Krause, Timy Phan, Ming Gui, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Tread: Token routing for efficient architecture-agnostic diffusion training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15703–15713, 2025.

Sunmin Lee, Taeho Kang, Jungnam Park, Jehee Lee, and Jungdam Won. Same: Skeleton-agnostic motion embedding for character animation. In *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–11, 2023.

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*, pp. 27827–27837, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Qingyuan Liu, Ke Lv, Kun Dong, Jian Xue, Zehai Niu, and Jinbao Wang. Text-to-any-skeleton motion generation without retargeting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12926–12936, 2025.

Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Clinton Ansun Mo, Kun Hu, Chengjiang Long, Dong Yuan, Wan-Chi Siu, and Zhiyong Wang. Pumps: Skeleton-agnostic point-based universal motion pre-training for synthesis in human motion tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14496–14506, 2025.

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 (ICCV)*, pp. 9488–9497, October 2023.

Sigal Raab, Inbar Gat, Nathan Sala, Guy Tevet, Rotem Shalev-Arkushin, Ohad Fried, Amit Haim Bermano, and Daniel Cohen-Or. Monkey see, monkey do: Harnessing self-attention in motion diffusion for zero-shot motion transfer. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–13, 2024.

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*, pp. 8748–8763. PmLR, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

Zhenwu Shi, Jingyu Gong, Peiwei Wang, Xingzan Wang, Tianwen Qian, Wenxi Li, Yuan Fang, Jiao Xie, Lizhuang Ma, and Shaohui Lin. Omni-supervised motion editing: Balancing change and invariance through positive-negative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30595–30606, June 2026.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-Or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023.

Xuan Wang, Kai Ruan, Liyang Qian, Guo Zhi Zhi, Chang Su, and Gaoang Wang. X-mogen: Unified motion generation across humans and animals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 10234–10242, 2026.

Andrew Witkin and Michael Kass. Spacetime constraints. *ACM Siggraph Computer Graphics*, 22(4): 159–168, 1988.

Mingxi Xu, Qi Wang, Zhengyu Wen, Phong Dao Thien, Zhengyu Li, Ning Zhang, Xiaoyu He, Wei Zhao, Kehong Gong, and Mingyuan Zhang. Necromancer: Breathing life into skeletons via bvh animation. *arXiv preprint arXiv:2602.06548*, 2026.

Yujie Yang, Zhichao Zhang, Jiazhou Chen, and Zichao Wu. Partmotionedit: Fine-grained text-driven 3d human motion editing via part-level modulation. *arXiv preprint arXiv:2512.24200*, 2025.

Zhangsihao Yang, Mingyuan Zhou, Mengyi Shan, Bingbing Wen, Ziwei Xuan, Mitch Hill, Junjie Bai, Guo-Jun Qi, and Yalin Wang. Omnimotiongpt: Animal motion generation with limited data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1249–1259, 2024.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15703–15712, 2025.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

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. In *The Thirteenth International Conference on Learning Representations*, 2025.

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.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *Transactions on Machine Learning Research*, 2024.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5745–5753, 2019.

Rui Zhu, Yingwei Pan, Yehao Li, Ting Yao, Zhenglong Sun, Tao Mei, and Chang Wen Chen. Sd-dit: Unleashing the power of self-supervised discrimination in diffusion transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8435–8445, 2024.

Algorithm 1: Skeleton augmentation (adapted from (Lee et al., 2023))

Input: Character skeleton pool $\mathcal{C}$; bone-scale range $[s_{\min}, s_{\max}]$; height-scale range $[h_{\min}, h_{\max}]$
Output: Augmented skeleton $\mathcal{S}'$

1. Sample base skeleton.
$\mathcal{S} \sim \mathrm{Uniform}(\mathcal{C})$; $\mathcal{S}' \leftarrow \mathrm{copy}(\mathcal{S})$;
2. Joint-level perturbations.
for *joint* $j \in \mathcal{S}'.\mathrm{joints}$ **do**
  if $j \in \{\mathrm{Hip}, \mathrm{LowerBack}, \mathrm{Shoulder}\}$ **then**
    randomly insert dummy joints adjacent to j;
  end
  if $j \in \{\mathrm{Spine}, \mathrm{Neck}\}$ **then**
    randomly add or remove sub-joints along j's chain;
  end
  if $j \in \{\mathrm{EndEffectors}\}$ **then**
    $j.\mathrm{offset} \leftarrow \mathbf{0}$ with random probability;
  end
  sample $s_x, s_y, s_z \overset{\mathrm{iid}}{\sim} \mathcal{U}(s_{\min}, s_{\max})$; $j.\mathrm{offset} \leftarrow j.\mathrm{offset} \odot (s_x, s_y, s_z)$;
end
3. Global scaling.
apply left-right symmetric bone-offset scaling;
sample $h \sim \mathcal{U}(h_{\min}, h_{\max})$; $\mathcal{S}'.\mathrm{height} \leftarrow \mathcal{S}'.\mathrm{height} \cdot h$;
return $\mathcal{S}'$;

A TopoMotionFix Construction Details

This appendix details the TopoMotionFix construction summarised in Section 4.1. We retarget each MotionFix editing pair onto a set of sampled skeletons while keeping the original text instruction unchanged. The retargeting pipeline largely reuses the protocol of SAME (Lee et al., 2023), which is also adopted by SkeleMotion-3D (Liu et al., 2025).

The full pipeline is split into two stages. Algorithm 1 defines the per-skeleton augmentation routine: starting from a base skeleton sampled from a character pool, we randomly add or remove joints around the spine and neck, optionally insert dummy joints at the hip, lower back, and shoulder, zero out random end-effector offsets, scale per-axis bone offsets uniformly with $[s_{\min}, s_{\max}] = [0.7, 1.3]$, apply left-right symmetric scaling, and randomly sample a global height. Algorithm 2 drives the overall dataset construction: for each MotionFix editing pair, we first append the original SMPL pair, then generate augmented skeletons via Algorithm 1, retarget both source and target motions onto each augmented skeleton (the editing pair must share the same target skeleton), and convert each retargeted clip into our per-joint motion representation.

The resulting dataset packages the 6,730 source-target editing pairs into 40,304 editing instances (6,730 on the original SMPL skeleton plus 33,574 on randomised humanoid skeletons), spanning 25,794 unique skeletal configurations. Each instance carries one source motion and one target motion on a shared skeleton, giving 80,608 motion-skeleton clips at the per-motion level. The 6,730 textual editing instructions are inherited from MotionFix and contain 1,416 unique sentences across 56,728 total words, averaging 8.4 words per instruction.

B Unseen-Topology Protocol and Evaluation on Non-SMPL Skeletons

This appendix details the unseen-topology protocol summarised in Section 4.2: how the held-out test sets are built and how motions generated on non-SMPL skeletons are scored with the MotionFix evaluator.

Held-out rigs and test sets. TopoMotionFix contains 92 humanoid rigs in total, of which 9 are assigned to the test set and 83 to the training set. Together with the SMPL skeleton of MotionFix, the 9 held-out rigs make up the 10 test topologies. Each held-out rig is used only in the test set,

Algorithm 2: TOPOMOTIONFIX construction

Input: MotionFix editing pairs $\mathcal{P} = \{(M_s^n, M_t^n, c^n)\}_{n=1}^N$ on the original SMPL skeleton $\mathcal{S}_0^n$
character skeleton pool $\mathcal{C}$
augmentations per pair K
Output: Multi-topology editing dataset $\mathcal{D}_{\text{topo}}$

1. Initialise.
$\mathcal{D}_{\text{topo}} \leftarrow \emptyset$;
2. Iterate over editing pairs.
for *each pair* $(M_s^n, M_t^n, c^n) \in \mathcal{P}$ **do**
  2.1 Original SMPL pass:
  $\mathcal{D}_{\text{topo}} \leftarrow \mathcal{D}_{\text{topo}} \cup \{(M_s^n, M_t^n, c^n, \mathcal{S}_0^n)\}$;
  2.2 Augmented topology pass:
  for $k = 1$ *to* K **do**
    $\mathcal{S}^{n,k} \leftarrow$ SKELETONAUGMENT($\mathcal{C}$); // Algorithm 1
    for $r \in \{\text{src}, \text{tgt}\}$ **do**
      $M_r^{n,k} \leftarrow$ RETARGET($M_r^n, \mathcal{S}^{n,k}$);
    end
    $\mathcal{D}_{\text{topo}} \leftarrow \mathcal{D}_{\text{topo}} \cup \{(M_s^{n,k}, M_t^{n,k}, c^n, \mathcal{S}^{n,k})\}$;
  end
end
return $\mathcal{D}_{\text{topo}}$;

with its original, fixed topology and bone lengths. The held-out rigs are removed both from the training samples and from the character pool $\mathcal{C}$ from which Algorithm 1 samples base skeletons, so no randomised variant of a held-out rig appears in training. The samples of the unseen-topology test set come from two sources. The first is the original MotionFix test split of $1{,}013$ source–target pairs on the SMPL skeleton. The second is built by retargeting each of these $1{,}013$ pairs, with the retargeting pipeline of SAME (Lee et al., 2023), onto 4 rigs drawn at random from the 9 non-SMPL held-out rigs. The unseen-topology test set therefore contains $5{,}065$ source–target–instruction triplets: $1{,}013$ on the SMPL skeleton and $4{,}052$ on the 9 held-out rigs.

Evaluation on non-SMPL skeletons. The evaluator provided by MotionFix is a TMR model (Petrovich et al., 2023) trained on the 135-dimensional SMPL representation (pelvis translation, 6D pelvis orientation, and 6D rotations of the 21 SMPL body joints). We retain this evaluator rather than training a topology-specific one, since it underlies every published result we compare against and thus keeps the unseen-topology scores directly comparable with their seen-topology counterparts. Motions generated on non-SMPL rigs are therefore retargeted onto the SMPL skeleton before scoring. Each edited motion generated on a non-SMPL skeleton is retargeted with the same SAME pipeline, converted to the 135-dimensional representation, and scored with the retrieval metrics. For scoring, the $4{,}052$ retargeted triplets are divided into four subsets of $1{,}013$, each containing every test pair exactly once. Each subset of $1{,}013$ samples is then evaluated exactly as in the single-topology setting, under both the batch-32 and the test-set gallery settings. The unseen-topology scores in Tables 1 and 3 are the average over the five subsets, namely the SMPL subset and the four retargeted subsets. Since the retargeting from a rig back to SMPL is itself lossy, we additionally report a ground-truth ceiling in Table 3 (GT (retargeted)), obtained by retargeting the MotionFix test targets onto the held-out rigs and back to SMPL through the same pipeline. The round trip changes every metric by at most 0.02 points relative to the original ground truth, so the retargeting loss is acceptable.

C DTW EDIT MAP CONSTRUCTION

This appendix details the DTW edit map construction summarised in Section 3.3. The mask $E \in \{0, 1\}^{T \times J}$ is constructed for each source–target pair and stored with the dataset. We write $p_{i,j} \in \mathbb{R}^3$ for the position and $r_{i,j} \in \mathbb{R}^6$ for the 6D rotation of joint j at frame i, with superscripts s and t marking the source and edited target.

The full procedure is given in Algorithm 3: we first compute a joint-level frame-pair distance from the position and rotation components with equal weights, then assemble a frame cost matrix where

Algorithm 3: DTW edit map construction

Input: Source motion M_s and edited target motion M_t, both in $\mathbb{R}^{T\times J\times D}$; threshold τ_g; window width b
Output: Binary edit mask $E \in \{0,1\}^{T\times J}$

1. Joint-level frame-pair distance.
$d_{i,i',j} \leftarrow \frac{1}{2}\|p^s_{i,j} - p^t_{i',j}\|_1 + \frac{1}{2}\|r^s_{i,j} - r^t_{i',j}\|_1$;
2. Windowed frame cost matrix.
$C_{i,i'} \leftarrow \begin{cases} \frac{1}{J}\sum_{j=1}^{J} d_{i,i',j}, & |i-i'| \le b \\ +\infty, & \text{otherwise} \end{cases}$;
3. DTW alignment.
$\Pi \leftarrow \mathrm{DTW}(C) \subseteq [T]\times[T]$; `// optimal alignment path`
$\pi(i) \leftarrow \arg\min_{i':(i,i')\in\Pi} C_{i,i'}$; `// best-aligned target frame`
$\bar{d}_{i,j} \leftarrow d_{i,\pi(i),j}$; `// per-joint distance along path`
4. Per-sample min-max normalisation.
$G_{i,j} \leftarrow (\bar{d}_{i,j} - \min_{i,j}\bar{d}_{i,j})/(\max_{i,j}\bar{d}_{i,j} - \min_{i,j}\bar{d}_{i,j})$;
5. Threshold to binary mask.
$E_{i,j} \leftarrow \mathbf{1}[G_{i,j} > \tau_g]$;
return E;

Table 3: Quantitative comparison under the test-set retrieval setting of the official SimMotionEdit evaluation implementation, with the same metrics as Table 1. Baseline numbers are taken from the respective papers; dash indicates unavailable; bold indicates best within each protocol.

Dataset	Protocol	Method	Generated-to-Target (Test set)				Generated-to-Source (Test set)			
			R@1↑	R@2↑	R@3↑	AvgR↓	R@1↑	R@2↑	R@3↑	AvgR↓
		GT	64.36	88.75	95.56	1.74	20.83	33.66	40.47	33.13
		GT (retargeted)	64.34	88.73	95.55	1.73	20.82	33.65	40.47	33.12
MotionFix	Seen	MDM (Tevet et al., 2023)	0.00	0.30	0.49	476.65	0.10	0.10	0.10	486.26
MotionFix	Seen	MDM-BP (Athanasiou et al., 2024)	8.69	14.71	18.36	180.99	30.21	36.82	40.47	106.05
MotionFix	Seen	TMED (Athanasiou et al., 2024)	14.51	21.72	28.73	56.63	22.41	34.45	40.57	31.42
MotionFix	Seen	MotionReFit (Jiang et al., 2025b)	14.13	23.52	30.53	54.06	43.77	56.72	64.13	24.09
MotionFix	Seen	SimMotionEdit (Li et al., 2025)	25.49	39.33	49.21	23.49	–	–	–	–
MotionFix	Seen	Han & Kim (2026)	29.45	45.26	54.55	16.42	–	–	–	–
MotionFix	Seen	OmniME (Shi et al., 2026)	32.02	50.20	59.88	13.06	–	–	–	–
MotionFix	Seen	TAME	**34.98**	**50.40**	**60.08**	**10.33**	**54.35**	**66.60**	**73.72**	**7.58**
TopoMotionFix	Unseen (SMPL+held-out rigs)	AnyTop (Gat et al., 2025)	24.70	37.75	45.26	17.49	46.25	60.47	67.59	8.85
TopoMotionFix	Unseen (SMPL+held-out rigs)	TAME w/o TCSP, w/o EFRA	23.72	37.15	45.85	20.14	44.07	57.51	65.81	7.82
TopoMotionFix	Unseen (SMPL+held-out rigs)	TAME w/o EFRA	26.68	**42.09**	50.79	17.11	54.74	66.80	72.13	8.80
TopoMotionFix	Unseen (SMPL+held-out rigs)	TAME	**28.46**	**42.09**	**53.16**	**16.53**	**63.83**	**79.05**	**83.60**	**6.96**

pairs beyond a window of width b on the diagonal are masked to $+\infty$, and run classical DTW on this matrix to obtain the optimal source-to-target alignment path. The path identifies, for each source frame, its best-aligned target frame; we extract the per-joint distance at this target frame, apply per-sample min-max normalisation into $[0,1]$, and threshold with τ_g to obtain the binary mask.

This window absorbs small temporal misalignments between source and target while preventing pathological matches between distant parts of the sequence; per-sample min-max normalisation makes the threshold τ_g comparable across samples regardless of their overall edit intensity. Throughout, we use $b = 10$ frames and $\tau_g = 0.2$.

D Test-Set Retrieval Results

Following the official evaluation implementation of SimMotionEdit (Li et al., 2025), which is also adopted by Han & Kim (2026) and OmniME (Shi et al., 2026), we additionally report the retrieval metrics of Section 4.2 under its test-set setting, in which each generated motion is ranked among the test motions rather than within a batch of 32. Table 3 lists the results; baseline numbers are taken from the respective papers, and a dash indicates an unavailable number. Under both protocols, TAME obtains the best score on every metric, for edit alignment and source preservation alike.

E Implementation Details

We adopt rectified flow matching (Liu et al., 2023; Lipman et al., 2023) with the linear interpolation of Section 3.3. The training timestep $t \in [0, 1]$ is drawn uniformly at random from a grid on $[0, 1]$ with spacing $1/1{,}000$, and inference uses 50 Euler sampling steps. Following Athanasiou et al. (2024), we apply two-way classifier-free guidance with a guidance scale of 2.5 for the text condition and 2.0 for the source motion condition, with dropout 0.05 applied independently on each of the text, source-motion, and joint conditions during training. We encode the natural-language edit instruction with a frozen CLIP-ViT-L/14 text encoder (Radford et al., 2021) (injected through cross-attention), and the joint-name strings of the target skeleton with the pretrained T5-base text encoder (Raffel et al., 2020) (added to the per-joint token embeddings).

The flow-matching transformer consists of $L = 9$ TAME blocks with 8 attention heads, a latent dimension of 512, and an FFN dimension of 1024. Our edit-focused representation alignment (EFRA) uses an EMA teacher with momentum $\alpha = 0.9999$, a projection head with hidden dimension 1024, and an alignment loss connecting student layer 4 to teacher layer 7 via cosine distance restricted to the DTW-derived edit mask (threshold $\tau_g = 0.2$), with auxiliary loss weight $\lambda_{\mathrm{EFRA}} = 0.5$ relative to the rectified-flow objective.

We train using the AdamW optimiser (Loshchilov & Hutter, 2019) with an initial learning rate of $1{\times}10^{-4}$ decayed to $1{\times}10^{-6}$ along a cosine schedule with 300 warmup steps, weight decay 0.01, for $1{,}000$ epochs at an effective batch size of 128, taking approximately one day on two NVIDIA RTX Pro 6000 (96 GB) GPUs. We implement TAME in PyTorch Lightning and train every ablation from scratch with this configuration unless otherwise noted.

F Perceptual Study

To complement the retrieval-based metrics in the main paper, we conduct a perceptual study with human participants to assess how well TAME's edits align with text instructions and how plausible the resulting motions appear, compared with SimMotionEdit under the seen-topology protocol and AnyTop under the unseen-topology protocol.

F.1 Setup

We run two parallel studies, one for the seen-topology protocol and one for the unseen-topology protocol of Table 1. Each participant in a study observes 10 sample sets, randomly drawn from the MotionFix test split. Within each sample set, the participant is shown the source motion, the text edit instruction, and three candidate edited motions:

- **Seen-topology study**: ground truth (GT), TAME, and SimMotionEdit (Li et al., 2025).
- **Unseen-topology study**: ground truth (GT), TAME, and AnyTop (Gat et al., 2025).

The three candidates are presented in random order within each sample set to avoid positional bias. Participants rate each candidate on two metrics, both on a 3-point Likert scale (higher is better):

- *Alignment with text*: how well the edited motion follows the instruction, irrespective of motion quality. 1 = the edited motion is completely different from the instruction; 2 = the edited motion follows only some parts of the instruction; 3 = the edited motion fully follows the instruction.
- *Plausibility*: the quality of the edited motion, irrespective of instruction-following. 1 = severe issues (self-intersections, unrealistic poses or movements); 2 = minor issues (foot sliding, plausible but slightly non-humanlike movements); 3 = no issues.

We follow the protocol of SimMotionEdit (Li et al., 2025) for the rating scales to allow direct comparison with their reported numbers under the seen-topology protocol.

F.2 RESULTS

We recruited $N = 25$ participants per study, consisting of graduate students and staff from a university campus, yielding 250 ratings per candidate per metric in each study. Table 4 reports the mean Likert score and standard deviation aggregated across all participants and sample sets, for the seen-topology and unseen-topology protocols.

Table 4: Perceptual evaluation. Mean $\pm$ standard deviation on the 3-point Likert scale across 25 participants and 10 sample sets per participant (250 ratings per cell). Higher is better. Each block compares ground truth (GT), our TAME, and one baseline: SimMotionEdit for the seen-topology protocol and AnyTop for the unseen-topology protocol.

Protocol	Method	Alignment ↑	Plausibility ↑
Seen	GT	2.53 ± 0.66	2.61 ± 0.69
	SimMotionEdit (Li et al., 2025)	2.08 ± 0.81	2.28 ± 0.76
	TAME (ours)	$\mathbf{2.31 \pm 0.74}$	$\mathbf{2.46 \pm 0.71}$
Unseen	GT	2.53 ± 0.66	2.61 ± 0.69
	AnyTop (Gat et al., 2025)	1.73 ± 0.84	1.98 ± 0.81
	TAME (ours)	$\mathbf{1.96 \pm 0.79}$	$\mathbf{2.14 \pm 0.77}$

Across both protocols, participants prefer TAME over the compared baseline by approximately 0.2 Likert points on both metrics, mirroring the retrieval-based gaps in the main paper. The Alignment gap is slightly larger than the Plausibility gap, indicating that human observers recognise TAME's edits as more faithful to the instruction while the baseline motions can still appear visually clean. Comparing TAME across protocols, scores drop by 0.32–0.35 points moving from seen to unseen topology, reflecting the additional difficulty of editing motions on skeletons that the model has not seen during training; even so, TAME remains substantially ahead of AnyTop, which is not designed for text-based editing.