

UniMotion: Motion Editing and Retargeting via Conditional Flow

ANONYMOUS AUTHOR(S)

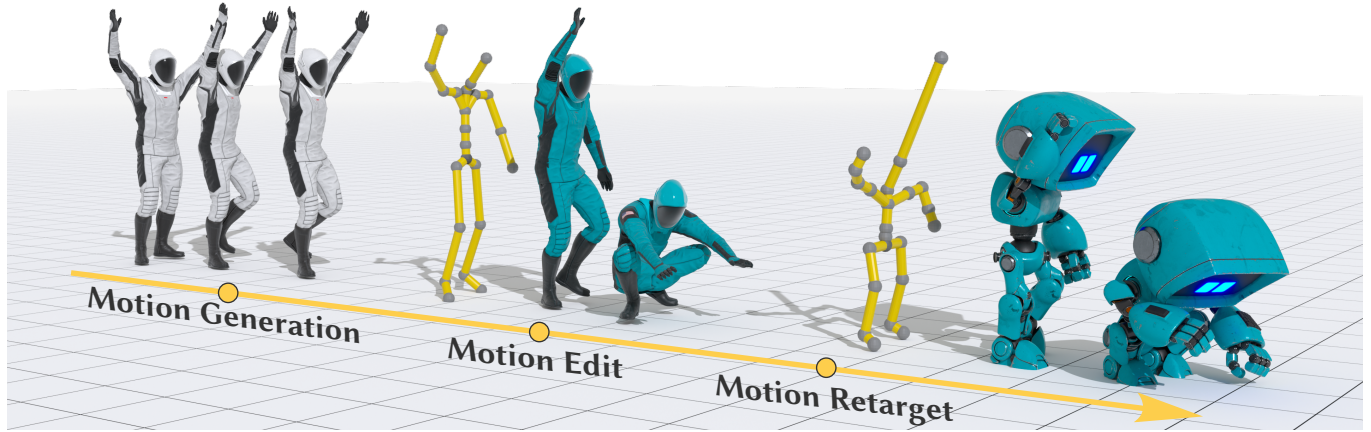


Fig. 1. **UniMotion unifies motion generation, editing, and retargeting.** A single rectified-flow model conditioned on text and skeleton enables (left) generation, (middle) zero-shot editing by changing only the text condition, and (right) zero-shot retargeting by changing only the skeleton condition.

Text-driven motion editing and skeletal retargeting are traditionally handled by fragmented pipelines with incompatible inputs and representations: editing relies on specialized generative steering, while retargeting is deferred to geometric post-processing. We present a unifying perspective where both tasks are cast as instances of *conditional transport* within a single generative framework. By leveraging recent advances in flow matching, we demonstrate that editing and retargeting are fundamentally the same generative task, distinguished only by which conditioning signal—semantic or structural—is modulated during inference. We implement this vision via a rectified-flow motion model jointly conditioned on text prompts and target skeletal structures. Our architecture extends a DiT-style transformer with per-joint tokenization and explicit joint self-attention to strictly enforce kinematic dependencies, while a multi-condition classifier-free guidance strategy balances text adherence with skeletal conformity. Experiments on SnapMoGen and a multi-character Mixamo subset show that a single trained model supports text-to-motion generation, zero-shot editing, and zero-shot retargeting. This unified approach simplifies deployment and improves structural consistency compared to task-specific baselines.

CCS Concepts: • **Computing methodologies** → **Animation**; *Machine learning*.

Additional Key Words and Phrases: motion editing, motion retargeting, text conditioning, T-pose conditioning, flow matching

ACM Reference Format:

Anonymous Author(s). 2026. UniMotion: Motion Editing and Retargeting via Conditional Flow. 1, 1 (March 2026), 10 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM XXXX-XXXX/2026/3-ART

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Character animation relies heavily on the ability to reuse and modify existing motion data. Traditionally, this is achieved through *motion retargeting*, which adapts motion across characters with different rest poses and proportions. Recently, text-to-motion generative models enable *prompt-driven motion editing*, allowing users to specify high-level semantic changes in natural language [Chen et al. 2023; Guo et al. 2025, 2024; Tevet et al. 2023; Zhang et al. 2022]. However, these two paradigms currently exist in isolation: editing is typically handled by generative models using scarce edit-specific supervision [Athanasios et al. 2024; Li et al. 2025] or heuristic steering [Hong et al. 2025], while retargeting remains a distinct post-process relying on inverse kinematics or learned mapping networks [Aberman et al. 2020; Gleicher 1998; Zhang et al. 2023a].

This fragmentation stems from the fact that editing and retargeting are usually formulated with incompatible inputs, objectives, and representations. First, the inputs differ: editing conditions on a *source motion* and a *text prompt*, whereas retargeting conditions on a *source motion* and a *target skeleton*. Second, the objectives diverge: text-driven editing seeks to preserve the source motion’s structure while altering its semantics, whereas retargeting seeks to preserve the semantic intent while adapting the motion to a new skeletal morphology. Third, and most critically, the underlying representations are mismatched. Most text-to-motion generators operate in a canonical output space (e.g., a fixed kinematic tree with standardized bone lengths) to simplify training [Chen et al. 2023; Tevet et al. 2023]. Consequently, structural adaptation is deferred to downstream tools rather than handled by the generative model itself. This separation complicates deployment—requiring multiple models and inconsistent representations—and hinders the composition of operations, such as simultaneous editing and retargeting, within a single interactive system.

We propose a unifying perspective where text-based editing and skeletal retargeting are fundamentally the same generative task, distinguished only by the conditioning signal being modulated. In our framework, both operations are cast as instances of *conditional transport*: **editing** corresponds to modifying the semantic condition (text) while fixing the structure, whereas **retargeting** corresponds to modifying the structural condition (skeleton) while fixing the semantics. Recent flow-based editing methods facilitate this unification by demonstrating that ordinary differential equations (ODEs) can map between source- and target-conditioned distributions without explicit inversion [Kulikov et al. 2025]. This suggests a simple yet powerful principle: if a motion generator is conditioned on *both* language and skeletal structure, changing either condition at inference time yields the desired manipulation “for free,” purely through the dynamics of the generative flow.

Guided by this principle, we introduce a single rectified-flow motion model jointly conditioned on a text prompt and a target character T-pose (serving as a proxy for bone lengths and rest pose). Our model is trained on motion clips paired with text and skeleton information. During inference, it is manipulated via a unified flow-based update rule: (i) **Text-based editing** changes the prompt while holding the target skeleton fixed; (ii) **Retargeting** swaps the skeleton condition while leaving the text fixed. To ensure effective skeletal conditioning, we adopt a DiT-style transformer backbone [Peebles and Xie 2022] operating on per-joint tokens and introduce a dedicated joint attention mechanism to explicitly model intra-frame kinematic dependencies. We also employ a multi-condition variant of classifier-free guidance to strictly control text adherence and skeletal conformity during both sampling and editing.

We evaluate our system on SnapMoGen [Guo et al. 2025] and a multi-character Mixamo subset. Across experiments, the same trained model and inference pipeline support text-to-motion generation, zero-shot text edits, and zero-shot intra-structure retargeting. This eliminates the need for task-specific toolchains. Beyond quantitative improvements in fidelity and consistency, qualitative results demonstrate that our dual-conditioning strategy better preserves motion structure during edits and produces target-conforming outputs even under significant morphological changes.

To summarize, the main contributions of this study include:

- We formulate text-driven motion editing and intra-structure motion retargeting under a unified conditional transport framework, treating them as equivalent generative tasks distinguished only by the modulated condition.
- We propose a dual-conditioning architecture—integrating text and target skeleton vectors—featuring a joint-token transformer with explicit joint attention to handle both semantic edits and cross-character transfer.
- We provide a comprehensive evaluation spanning generation, editing, and retargeting, demonstrating that a single model can serve as a versatile motion manipulation engine without task-specific fine-tuning.

2 Related Work

2.1 Text-to-Human Motion Generation

Human motion generation from text has seen rapid progress recently, fueled by gradual scale-growing datasets [Fan et al. 2025; Guo et al. 2025, 2022] and advances in generative modeling. The progress is largely driven by new *auto-regressive* methods and *diffusion* formulations. Token-based approaches first compress motion into discrete codes, typically via VQ-VAE style quantization [van den Oord et al. 2018], and then model the token sequence autoregressively, e.g. T2M-GPT [Zhang et al. 2023b]. MoMask further adopts masked modeling with hierarchical quantization for high-fidelity generation [Guo et al. 2024], and Guo et al. [2025] improves this method by using a multi-scale RVQ.

In parallel, diffusion-based generators have become a dominant framework, mitigating discretization and compression information loss in VQ-based methods. Early efforts include MDM [Tevet et al. 2023] and MotionDiffuse [Zhang et al. 2022]. Subsequent work advances this direction through iterative generation in StableMoFusion [Huang et al. 2024a] and latent-diffusion formulations such as MARDM [Meng et al. 2024]. Building on these diffusion formulations, SALAD introduces skeleton-aware latent diffusion to strengthen the generation and editing [Hong et al. 2025].

Skeleton priors during generation. Despite architectural differences, most text-to-motion generators are trained and sampled on a fixed canonical kinematic tree and do not explicitly incorporate a *skeleton prior* during the generative process. Instead, structural cues are typically encoded implicitly through the motion parameterization or through the learned encoder and token codebooks in VQ-based pipelines [Guo et al. 2024; van den Oord et al. 2018]. Consequently, adapting generated motions to different skeletons is usually deferred to a separate retargeting stage, see Sec. 2.3. Only a recent attempt [Cao et al. 2025] has been made by using skeleton-motion data and human specified key joint correspondence to guide the generation and retarget based on Gat et al. [2025].

2.2 Motion Editing

Motion editing aims to modify an existing motion while preserving non-edited content. Text-based motion editing has recently been studied under supervised and zero-shot settings. Supervised approaches rely on triplets {source motion, edit text, target motion}; MotionFix introduces a semi-automatically collected benchmark of such triplets and trains a conditional diffusion model for text-driven edits [Athanasios et al. 2024]. Building on this formulation, SimMotionEdit augments diffusion-based editing with an auxiliary motion similarity prediction objective [Li et al. 2025]. Zero-shot motion editing often uses mask-based inpainting to localize edits within a sequence [Tevet et al. 2023]. Related work also explores editing by steering a pretrained generator via attention or conditioning control [Chen et al. 2025; Hong et al. 2025], or by applying structured latent or code edits as in CoMo [Huang et al. 2024b].

Connections to image and video editing. Beyond motion-specific methods, our approach draws inspirations on text-driven image and video editing. Diffusion-based editing commonly realizes edits via attention control or inversion, as in Prompt-to-Prompt [Hertz et al. 2022] and FateZero [Qi et al. 2023]. FlowEdit demonstrates

inversion-free text-based editing using pretrained flow models, offering an alternative to inversion-based diffusion editing and directly motivating our design choices [Kulikov et al. 2025].

Edit prompts and data scarcity. Many text-based motion editing setups require an *edit instruction*, which is qualitatively different from standard text-to-motion captions and substantially less available at scale; hence, edit-text datasets are typically much smaller than text-motion pair counterparts and often require semi-automatic construction [Athanasios et al. 2024].

2.3 Motion Retargeting

Motion retargeting transfers motion from a source character to a target character with different morphology, such as bone lengths, skeleton structures, and mesh shape, while preserving motion semantics and physical plausibility. In recent years, learning-based approaches have become a dominant direction. Many methods formulate retargeting as unpaired translation between skeleton domains, combining cycle consistency with adversarial losses [Aberman et al. 2020; Lim et al. 2019; Villegas et al. 2018]. SAME adopts supervised training to learn a skeleton-agnostic motion latent and uses a single model to handle arbitrary skeleton retargeting [Lee et al. 2023].

Beyond skeleton-only formulations, mesh-aware retargeting highlights that avoiding interpenetrations and maintaining plausible surface interactions requires geometry-aware reasoning. Recent works therefore incorporate mesh-derived distance cues to enforce body-part relationship constraints: R2ET uses voxel-based distance-field losses [Zhang et al. 2023a], MeshRet relies on dense surface sensors [Ye et al. 2024], and SMRNet enforces relationships via surface edges sampling and convex-hull constraints [Zhang et al. 2025].

Notably, in contrast to the rapid adoption of diffusion/flow frameworks for text-to-motion generation and editing, as discussed in Sec. 2.2, motion retargeting has rarely been studied under this generative modeling; most methods instead learn from traditional adversarial based models or directly paired mappings.

3 METHOD

We propose treating motion retargeting as a conditional *generative* problem rather than a geometric optimization task. Under this view, the target skeleton (defined by bone lengths and rest pose) serves as a conditioning signal analogous to a “style” or “domain” code, while the motion semantics serve as the “content.” This parallels the logic of text-driven editing: just as changing a text prompt modifies the semantic trajectory of a motion, changing a skeleton condition modifies its morphological trajectory. By leveraging the FlowEdit framework [Kulikov et al. 2025] on top of a unified generative backbone, we enable both text-driven semantic edits and structure-driven retargeting within a single model, without requiring task-specific modules or training.

To achieve this vision, our approach relies on three integrated components: a dual-conditioned transformer backbone, a conditional rectified flow training strategy, and a unified FlowEdit inference scheme, as summarized in Figure 2 and Figure 3. In the subsequent sections, we first formalize the problem definition and data representation, followed by detailed descriptions of the model architecture and the unified inference procedure.

3.1 Problem Formulation

Problem Definition. We represent a human motion clip as a sequence of feature vectors $\mathbf{x} \in \mathbb{R}^{T \times D}$, where T is the sequence length and D is the feature dimension. Each motion $\mathbf{x}$ is associated with two conditioning signals: (i) a **semantic condition** $\mathbf{c}_{\text{txt}}$, derived from a natural language prompt, and (ii) a **structural condition** $\mathbf{c}_{\text{skel}}$, representing the character’s skeletal morphology (e.g., bone lengths and rest pose).

Our objective is to learn a *single* conditional distribution $p(\mathbf{x} | \mathbf{c}_{\text{txt}}, \mathbf{c}_{\text{skel}})$ that unifies generation and manipulation under a shared conditional transport framework. Specifically, the model must support three operations:

- (1) **Text-to-Motion Generation:** Sampling $\mathbf{x} \sim p(\mathbf{x} | \mathbf{c}_{\text{txt}}, \mathbf{c}_{\text{skel}})$ for any target skeleton $\mathbf{c}_{\text{skel}}$ within the supported topology.
- (2) **Zero-Shot Text-Based Editing:** Given a source motion $\mathbf{x}_{\text{src}}$ and its conditions, generating a modified motion $\mathbf{x}_{\text{tgt}}$ that adheres to a new prompt $\mathbf{c}_{\text{txt}}^{\text{tgt}}$ while preserving the non-edited part of the $\mathbf{c}_{\text{txt}}^{\text{src}}$ and the original motion.
- (3) **Zero-Shot Intra-Structure Retargeting:** Given a source motion $\mathbf{x}_{\text{src}}$ generated with skeleton $\mathbf{c}_{\text{skel}}^{\text{src}}$, producing a new motion $\mathbf{x}_{\text{tgt}}$ that preserves the source semantics but conforms strictly to a new target skeleton $\mathbf{c}_{\text{skel}}^{\text{tgt}}$ (where the skeletons share a graph topology but differ in bone lengths).

Data Representation. To simultaneously satisfy the requirements of standard evaluation metrics and precise geometric control, we operate in a concatenated feature space:

$$\mathbf{x}_t = [\mathbf{x}_t^{\text{gen}}; \mathbf{x}_t^{\text{ret}}], \quad \mathbf{x}_t^{\text{gen}} \in \mathbb{R}^{D_{\text{gen}}}, \mathbf{x}_t^{\text{ret}} \in \mathbb{R}^{D_{\text{ret}}}. \quad (1)$$

Generative Features ($\mathbf{x}^{\text{gen}}$). The first block follows the standard SnapMoGen layout ($D_{\text{gen}} = 8 + 12J$), consisting of root velocities (yaw/planar) and heights, local joint rotations, positions and velocities relative to the root, and foot contacts. While $\mathbf{x}^{\text{gen}}$ is essential for compatibility with existing text-to-motion metrics (e.g., Fréchet Inception Distance (FID)), it is ill-suited for retargeting. It uses root angular velocities instead of absolute angles, and temporally smooths rotations, discarding the absolute spatial frame and accumulating position error, observed as “root drift” when retargeting with $\mathbf{x}^{\text{gen}}$ alone.

Retargeting Features ($\mathbf{x}^{\text{ret}}$). To address the geometric limitations of $\mathbf{x}^{\text{gen}}$, we append AnyTop-style [Gat et al. 2025] features ($D_{\text{ret}} = 12J$) for each joint $j \in \{1, \dots, J\}$:

$$\mathbf{x}_{t,j}^{\text{ret}} = [\mathbf{p}_{t,j}; \mathbf{r}_{t,j}^{6D}; \mathbf{v}_{t,j}] \in \mathbb{R}^{12}. \quad (2)$$

Here, $\mathbf{p}_{t,j}$ and $\mathbf{v}_{t,j}$ denote canonicalized root-relative joint positions and velocities. Unlike using the root angular velocities to reconstruct the root transformations in $\mathbf{x}^{\text{gen}}$, our root rotation is directly obtained from $\mathbf{r}_{t,j}^{6D}$. To accurately predict $\mathbf{p}_{t,j}$ conditioned on a skeleton vector $\mathbf{c}_{\text{skel}}$, the model must learn the underlying kinematic constraints (i.e., bone lengths). This *implicit structure modeling* is the foundation of our retargeting capability: it allows the FlowEdit inference process to continuously “warp” the skeleton geometry from source to target.

Condition Inputs. We parameterize the semantic condition $\mathbf{c}_{\text{txt}}$ using text tokens $\mathbf{P} = \{\mathbf{w}_i\}_{i=1}^L$ from a frozen T5 encoder. The structural condition $\mathbf{c}_{\text{skel}}$ is parameterized as a vector $\mathbf{S} \in \mathbb{R}^{D_{\text{ret}}}$ following

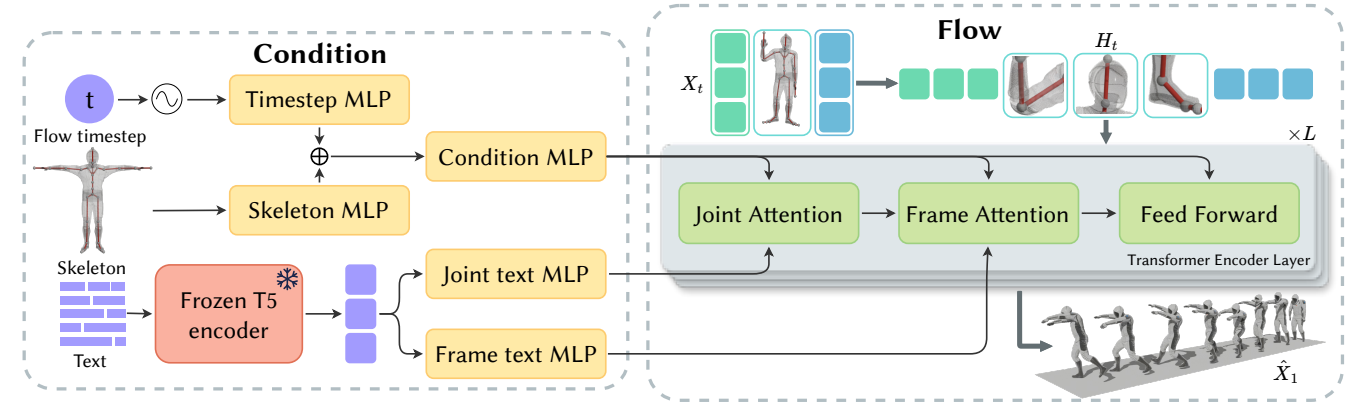


Fig. 2. Model Architecture. Input frame tokens are reshaped into per-joint tokens for processing. Time and skeleton conditions are injected via AdaLN, while text embeddings are integrated through cross-attention.

the layout of Eq. (2). In practice, we set the position channels of S to the target character's T-pose offsets (defining bone lengths and rest pose) and set the remaining rotation/velocity channels to identity/zero.

3.2 Model Architecture

We adopt a DiT-style transformer backbone that is jointly conditioned on text and skeleton. Let $\mathbf{X} \in \mathbb{R}^{T \times D}$ denote an input state such as $\mathbf{x}_\tau$, and let J be the joint count. Unlike prior motion transformers that treat each frame feature as a single token, we encourage body-part-level generation by operating on *per-joint tokens*. Concretely, instead of projecting $\mathbf{X}_t$ to an arbitrary hidden vector, we project it to J joint chunks and reshape:

$$\mathbf{H}_t = \text{reshape}(\mathbf{W}_{\text{in}}\mathbf{X}_t + \mathbf{b}_{\text{in}}) \in \mathbb{R}^{J \times d}, \quad d = \frac{D_h}{J}, \quad (3)$$

where D_h is the model total hidden size and $\mathbf{H}_{t,j}$ is the token for joint j at frame t . The input projection learns how to split the whole feature dimensions into joint parts.

Joint attention. Standard transformer backbones capture temporal structure via attention between frame tokens, while spatial inter-joint structure is typically handled *indirectly* by the per-token feed-forward network that mixes all joint parts inside the hidden frame vector. Without a strong skeletal inductive bias, this mixing must learn inter-joint relations through a generic MLP, which is parameter-heavy and often slow to learn. We keep the existing temporal attention and feed-forward updates, and additionally introduce a dedicated **joint self-attention** module that attends over joints within each frame by forcing the hidden dimension to be split into J joint chunks. This module is conceptually similar to attention module in graph transformers, but implemented as dense attention over the J joint tokens at each frame. Within each layer, we apply joint self-attention independently at each time step:

$$\mathbf{H}_{t,\cdot} \leftarrow \mathbf{H}_{t,\cdot} + g_{\text{jnt}}(\mathbf{c}) \cdot \text{Attn}_{\text{jnt}}(\text{AdaLN}(\mathbf{H}_{t,\cdot}, \mathbf{c})), \quad (4)$$

where $\mathbf{c}$ is the conditioning vector defined below, Attn_{jnt} attends over joints $j = 1 \dots J$ using joint rotary embeddings, and $g_{\text{jnt}}(\mathbf{c})$ is the AdaLN-Zero residual gate. This directly models intra-frame

joint dependencies, which is critical when generating on characters with different bone lengths.

Frame attention and feed-forward. After updating joint tokens, we form frame tokens back by concatenating all joints per frame, $\mathbf{F}_t = [\mathbf{H}_{t,1}; \dots; \mathbf{H}_{t,J}] \in \mathbb{R}^{D_h}$, and run temporal attention over the motion sequence with frame rotary embeddings. Finally, a SwiGLU feed-forward network updates frame tokens and the outputs are reshaped back to joints.

Skeleton and flow-time condition injection. We encode flow time τ with a sinusoidal embedding and map it to $\mathbf{c}_\tau \in \mathbb{R}^{D_h}$ with an MLP. We encode the target T-pose vector with a skeleton MLP, yielding $\mathbf{c}_s \in \mathbb{R}^{D_h}$. The final conditioning vector is

$$\mathbf{c} = \phi(\mathbf{c}_\tau + \mathbf{c}_s) \in \mathbb{R}^{D_h}, \quad (5)$$

where ϕ is a small condition-merging MLP. We inject $\mathbf{c}$ into every transformer block using AdaLN-Zero modulation with learned shift, scale, and a residual gate. This conditioning controls both joint-level and frame-level updates; for example, it drives the gate $g_{\text{jnt}}(\mathbf{c})$ in Eq. (4).

Text injection at joint and frame resolutions. We encode text prompt tokens $\{w_i\}_{i=1}^L$ with a frozen T5 encoder, producing token embeddings $\mathbf{e}_i \in \mathbb{R}^{d_{\text{ext}}}$. We then form: (i) *frame-level* text tokens $\mathbf{E}^{\text{frm}} \in \mathbb{R}^{L \times D_h}$ and (ii) *joint-level* text tokens $\mathbf{E}^{\text{jnt}} \in \mathbb{R}^{L \times d}$ via two separate linear projections. After joint attention, joint tokens query the text via cross-attention using $\mathbf{E}^{\text{jnt}}$ as keys/values, while $\mathbf{E}^{\text{frm}}$ participates in the subsequent frame-text cross attention after the frame attention. This design injects language at both the spatial resolution over joints and the temporal resolution over frames.

Output head. The final joint tokens are normalized and reshaped back to D_h , then projected to the motion feature dimension:

$$\hat{\mathbf{x}}_1 = \mathbf{W}_{\text{out}} \text{reshape}(\text{Norm}(\mathbf{H})) + \mathbf{b}_{\text{out}}. \quad (6)$$

We initialize $\mathbf{W}_{\text{out}}, \mathbf{b}_{\text{out}}$ to zero to stabilize early training.

3.3 Rectified Flow Matching

We model motions with a rectified flow (RF) [Liu et al. 2022] defined on the continuous-time path $\mathbf{x}_\tau \in \mathbb{R}^{T \times D}$, $\tau \in [0, 1]$. Let $\mathbf{x}_1$ be a data

sample and $\mathbf{x}_0 \sim \mathcal{N}(0, I)$ be base noise. To match FlowEdit’s assumption that the noisy state is a *linear interpolant*, we use rectified flow matching with the linear path

$$\mathbf{x}_\tau = (1 - \tau)\mathbf{x}_0 + \tau\mathbf{x}_1, \quad (7)$$

whose ground-truth velocity is

$$\mathbf{u}^* = \frac{d\mathbf{x}_\tau}{d\tau} = \mathbf{x}_1 - \mathbf{x}_0. \quad (8)$$

Predicting the clean target. Instead of predicting velocity, we predict the clean target sample $\hat{\mathbf{x}}_1 = f_\theta(\mathbf{x}_\tau, \tau; P, S)$ following the previous work [Cuba et al. 2025; Tevet et al. 2023], which empirically yields smoother outputs. To support inferencing and edit, we then convert the target prediction to a velocity field via

$$\mathbf{v}_\theta(\mathbf{x}_\tau, \tau; P, S) = \frac{\hat{\mathbf{x}}_1 - \mathbf{x}_\tau}{1 - \tau}, \quad (9)$$

where we clamp τ away from 1 for numerical stability.

Flow matching loss. Training minimizes a mean-squared error on the predicted target directly. Because $\mathbf{x}_\tau$ concatenates $\mathbf{x}_\tau^{\text{gen}}$ and $\mathbf{x}_\tau^{\text{ret}}$ blocks, we enforce the objective *per feature block*:

$$\begin{aligned} \mathcal{L}_{\text{simple}} = & \lambda_{\text{gen}} \mathbb{E}_{\mathbf{x}_1, \mathbf{x}_0, \tau} \left[\left\| \mathbf{f}_\theta^{\text{gen}}(\mathbf{x}_\tau, \tau; P, S) - \mathbf{x}_1^{\text{gen}} \right\|^2 \right] \\ & + \lambda_{\text{ret}} \mathbb{E}_{\mathbf{x}_1, \mathbf{x}_0, \tau} \left[\left\| \mathbf{f}_\theta^{\text{ret}}(\mathbf{x}_\tau, \tau; P, S) - \mathbf{x}_1^{\text{ret}} \right\|^2 \right], \end{aligned} \quad (10)$$

where $\mathbf{f}_\theta^{\text{gen}}$ and $\mathbf{f}_\theta^{\text{ret}}$ are the corresponding slices of the concatenated prediction. This prevents the higher-dimensional block from implicitly dominating the loss signal. We set $\lambda_{\text{gen}} = \lambda_{\text{ret}} = 1$ unless otherwise stated.

Multi-Condition Classifier-Free Guidance. To support flexible conditioning and strong text adherence, we use a multi-condition extension of classifier-free guidance (CFG). At sampling time, we evaluate the model under an unconditioned input and several conditioned variants: **uncond** ($P = \emptyset, S = \emptyset$), **text** ($P = \text{text}, S = \emptyset$), **skeleton** ($P = \emptyset, S = \text{skeleton}$), **text+skeleton** ($P = \text{text}, S = \text{skeleton}$). We fill the text tokens with padding tokens when $P = \emptyset$, and set all skeleton vector channels to zero when $S = \emptyset$. Let $\mathbf{v}_u$ denote the unconditioned velocity, and let $\mathbf{v}_k$ denote a conditioned velocity. We combine them as

$$\mathbf{v}_{\text{CFG}} = \mathbf{v}_u + \sum_k w_k (\mathbf{v}_k - \mathbf{v}_u), \quad (11)$$

where w_k are user-controlled guidance weights (e.g., w_{text} , w_{skeleton} , $w_{\text{text+skeleton}}$). During training we randomly drop conditions to learn the branches required by Eq. (11). Specifically, we drop *both* text and skeleton condition with probability p_{both} , and additionally drop *only* the text condition with probability p_{text} . This reflects our downstream use cases: we never need to generate motions with text-only conditioning, and without a skeleton condition the position channels can induce a random skeleton that does not match standard downstream formats (T-pose + joint rotations).

3.4 FlowEdit for Zero-Shot Editing and Retargeting

Unlike SDEdit and ODEdit, which edit by adding noise to the input and re-running the generative process, FlowEdit [Kulikov et al. 2025] constructs a direct transport ODE between the source- and target-conditioned distributions, without an explicit inversion or round-trip through pure noise. This shorter shared-noise path empirically

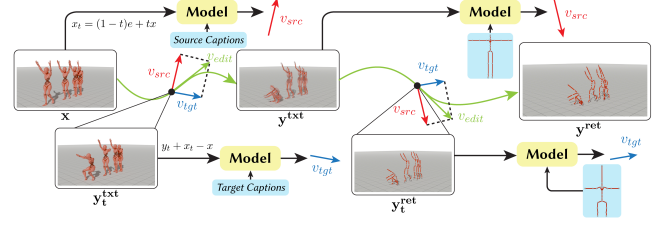


Fig. 3. Applications of our unified inference scheme. Left: text-based editing by changing the text condition. Right: retargeting by changing only the skeleton condition. Both use the same pre-trained model and an inversion-free update rule, where the edit velocity is obtained by combining velocity predictions under different conditions.

yields lower transport cost and better structure preservation, while remaining optimization-free and model-agnostic. FlowEdit treats editing as solving for an updated motion $\mathbf{y}$ by integrating a *difference-of-velocities* field under two conditions along a shared noise path.

The original FlowEdit formulation parameterizes time as $t \in [0, 1]$ with $t = 0$ representing data and $t = 1$ representing noise. In this paper we instead use $\tau \in [0, 1]$ with $\tau = 0$ representing noise and $\tau = 1$ representing data, consistent with our rectified-flow interpolation in Eq. (7). Let the source motion be $\mathbf{x}$ with source condition c_s and a target condition c_t . We define a noisy source trajectory

$$\tilde{\mathbf{x}}_\tau = (1 - \tau)\epsilon_\tau + \tau\mathbf{x}, \quad \epsilon_\tau \sim \mathcal{N}(0, I). \quad (12)$$

Update rule. Starting from $\mathbf{y}_{\tau_{\min}} = \mathbf{x}$, we integrate:

$$\frac{d\mathbf{y}_\tau}{d\tau} = \mathbf{v}_{\text{CFG}}(\mathbf{y}_\tau + \tilde{\mathbf{x}}_\tau - \mathbf{x}, \tau; c_t) - \mathbf{v}_{\text{CFG}}(\tilde{\mathbf{x}}_\tau, \tau; c_s), \quad (13)$$

where $\tau \in [\tau_{\min}, 1]$. The two velocity evaluations share the same $\tilde{\mathbf{x}}_\tau$ and thus the same noise. This is shown in Fig. 3.

Edit strength. We initialize at $\tau_{\min} \geq 0$ so the first target evaluation is anchored to the same partially noised state as the source: $\mathbf{y}_{\tau_{\min}} + \tilde{\mathbf{x}}_{\tau_{\min}} - \mathbf{x} = \tilde{\mathbf{x}}_{\tau_{\min}}$. Smaller $\tau_{\min}$ injects more noise and yields stronger edits, while larger $\tau_{\min}$ produces weaker edits that better preserve the source motion. We additionally use separate classifier-free guidance strengths for the source and target branches. We mirror this by allowing different guidance weights in Eq. (11) for the two velocity evaluations: $\{w_k^{(s)}\}$ for $\mathbf{v}_{\text{CFG}}(\cdot; c_s)$ and $\{w_k^{(t)}\}$ for $\mathbf{v}_{\text{CFG}}(\cdot; c_t)$. Intuitively, a smaller source guidance helps preserve the input motion, while a larger target guidance enforces the desired prompt or skeleton.

Task instantiations. (i) **Zero-shot text-based editing:** keep the skeleton fixed and change only text, $c_s = (P, S)$, $c_t = (P', S)$. (ii) **Zero-shot intra-structure retargeting:** keep text empty and change only skeleton, $c_s = (\emptyset, S)$, $c_t = (\emptyset, S')$. Both tasks use the same trained model and the same update rule, without extra training or fine-tuning.

4 Experiments

4.1 Dataset

SnapMoGen. We use the SnapMoGen dataset [Guo et al. 2025], which is designed for text-to-human-motion generation. It contains 20,450

Table 1. Text-to-motion generation quantitative evaluation on the SnapMoGen test split. Baseline numbers are taken from SnapMoGen (mean $\pm$ 95% CI). Bold and underlined numbers indicate the best and second-best results, respectively.

Methods	R Precision $\uparrow$			FID $\downarrow$	CLIP Score $\uparrow$	MModality $\uparrow$
	Top 1	Top 2	Top 3			
MDM	0.503 $\pm$ 0.002	0.653 $\pm$ 0.002	0.727 $\pm$ 0.002	57.783 $\pm$ 0.092	0.481 $\pm$ 0.001	13.412$\pm$0.231
T2M-GPT	0.618 $\pm$ 0.002	0.773 $\pm$ 0.002	0.812 $\pm$ 0.002	32.629 $\pm$ 0.087	0.573 $\pm$ 0.001	9.172 $\pm$ 0.181
StableMoFusion	0.679 $\pm$ 0.002	0.823 $\pm$ 0.002	0.888 $\pm$ 0.002	27.801 $\pm$ 0.063	0.605 $\pm$ 0.001	9.064 $\pm$ 0.138
MARDM	0.659 $\pm$ 0.002	0.812 $\pm$ 0.002	0.860 $\pm$ 0.002	26.878 $\pm$ 0.131	0.602 $\pm$ 0.001	9.812 $\pm$ 0.287
MoMask	0.777 $\pm$ 0.002	0.888 $\pm$ 0.002	0.927 $\pm$ 0.002	17.404 $\pm$ 0.051	<u>0.664$\pm$0.001</u>	8.183 $\pm$ 0.184
MoMask++	<u>0.802$\pm$0.001</u>	<u>0.905$\pm$0.002</u>	<u>0.938$\pm$0.001</u>	15.060$\pm$0.065	0.685$\pm$0.001	7.259 $\pm$ 0.180
Ours	0.917$\pm$0.001	0.973$\pm$0.001	0.987$\pm$0.001	<u>16.567$\pm$0.045</u>	0.663 $\pm$ 0.001	<u>11.259$\pm$0.293</u>

motion clips that total 43.7 hours, with 122,565 text descriptions whose average length is 48 words. Each clip lasts 4 to 12 seconds at 30 FPS. SnapMoGen is provided in a canonical BVH skeleton representation. We treat it as a single-skeleton dataset and use the provided skeleton for all text-to-motion evaluations.

Mixamo retargeting subset. Since SnapMoGen contains only a single canonical skeleton, we constructed a supplementary multi-character dataset from Mixamo [Adobe Systems Inc. 2018] to enable structural retargeting training. We curated a diverse subset of motions by filtering for common locomotion and action keywords (e.g., run, walk, sit) and subsampling 30% of the matches to minimize redundancy. To model cross-morphology transfer, we paired each selected motion with three distinct characters, creating varied source-skeleton combinations. All Mixamo characters are selected in the dataset, making it multiple times larger than the previous methods. These assets were standardized to the SnapMoGen topology by normalizing character heights, pruning incompatible joints (e.g., fingers), and reordering the kinematic chain to match the canonical depth-first traversal.

To integrate this data with the text-driven SnapMoGen pipeline, we generated synthetic captions using Qwen3-VL [Bai et al. 2025]. For each motion, we produced six caption variants using the original Mixamo filename as a prompt prior, ensuring the style and length distribution aligned with SnapMoGen’s ground-truth labels. Finally, we merged this Mixamo subset into the SnapMoGen dataset, maintaining an 80/20 train-test split across both data sources.

4.2 Implementation details

Model Configuration. We employ a 12-layer transformer backbone with 12 attention heads and a feed-forward expansion ratio of 3. To align with our per-joint tokenization strategy, we set the total hidden dimension to $D_h = 432$. Given a standard character topology of $J = 24$ joints, this allocates $d = 18$ hidden channels per joint token, ensuring sufficient capacity for the joint self-attention mechanism. We apply a dropout rate of 0.1 throughout the network. For text conditioning, we utilize a frozen T5-Large encoder [Ni et al. 2022] to extract semantic embeddings.

Training Protocol. We train the model with fixed-length motion windows of 320 frames. Optimization is performed using AdamW [Loshchilov and Hutter 2019] with a learning rate of 5×10^{-5} and a weight decay of 5×10^{-3} . We use a global batch size of 512 and

Table 2. Ablation study of the generation task on the SnapMoGen test split.

Methods	FID $\downarrow$	CLIP $\uparrow$	MM $\uparrow$
Ours w/o jnt attn	17.205 $\pm$ 0.061	0.664 $\pm$ 0.001	10.941 $\pm$ 0.321
Ours w/o jnt attn, larger D_h	17.085 $\pm$ 0.035	0.668$\pm$0.001	10.376 $\pm$ 0.345
Ours	16.567$\pm$0.045	0.663 $\pm$ 0.001	11.259$\pm$0.293

train 500 epochs. The entire training process takes approximately 8 hours on a single node equipped with 8 NVIDIA H100 GPUs.

4.3 Text-to-motion generation

We first evaluate the model’s core generative capability on the SnapMoGen test split. Note that for this standard benchmark, we use the single canonical SnapMoGen skeleton to ensure fair comparison with baselines.

Metrics. Following previous works [Guo et al. 2025; Tevet et al. 2023], we report FID, R-Precision (Top 3), CLIP score, and Multimodality. All metrics are computed in the feature space of a pre-trained TMR model [Petrovich et al. 2023].

We sample motions using the fourth-order Runge–Kutta (RK4) integrator with 100 steps [Hairer et al. 1993]. For classifier-free guidance (Eq. 11), we empirically set the guidance scales to $w_{\text{txt}} = 0.5$, $w_{\text{skel}} = 0$, and $w_{\text{both}} = 1.0$. This configuration prioritizes the joint text-skeleton condition while providing a moderate boost to semantic adherence via the text-only branch.

Figure 5 visualizes samples generated from long, complex prompts. Table 1 compares our method against state-of-the-art baselines. Our model achieves the highest R-Precision scores across all k-values, indicating superior semantic alignment with complex prompts. On FID and Multimodality, we consistently rank second-best, performing on par with top-tier methods like MoMask++ [Guo et al. 2025]. Crucially, prior methods often trade alignment for diversity. In contrast, UniMotion maintains a robust balance: it delivers state-of-the-art alignment and competitive realism without collapsing diversity. Most importantly, these results are achieved using a *unified* model trained to support arbitrary skeletons. This demonstrates that our dual-conditioning architecture incurs no performance penalty on the standard single-skeleton task.

Ablation Study. We investigate the impact of our architectural choices in Table 2. Removing the joint self-attention causes a sharp

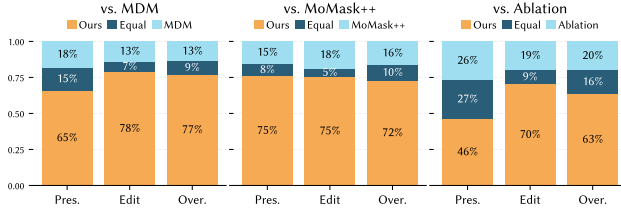


Fig. 4. **Perceptual user study.** Pairwise preferences (Ours/Equal/Baseline) for Source Preservation (Pres.), Edit Accuracy (Edit), and Overall Quality (Over.). Ours is preferred across all three criteria vs. MDM [Tevet et al. 2023] (65/78/77%), MoMask++ [Guo et al. 2025] (75/75/72%), and the mask-based ablation (46/70/63%) (Pres./Edit/Over.).

degradation in FID, indicating its vital role in modeling realistic kinematic chains. Moreover, despite adding only $\sim 0.4M$ parameters, the joint-attention model outperforms a wider baseline (hidden dim 576; 90M parameters) that lacks this mechanism. This confirms that the performance gain stems from the *inductive bias* of explicit spatial modeling rather than raw capacity.

4.4 Motion Editing

We evaluate model’s zero-shot editing capabilities, focusing on semantic modification while maintaining the structural integrity of the source motion.

Experimental Setup. We evaluate global and local editing, including motion addition, deletion, and replacement on 20 motions randomly sampled from the test split. We compare against MDM [Tevet et al. 2023], MoMask++ [Guo et al. 2025], and an ablation integrating MDM’s editing protocol into our framework.

Inference Configuration. We use 100 integration steps and control edit strength via starting time $\tau_{\min} \approx 0.1$. The source text guidance is set to 1.5, while target text guidance is increased to ≈ 3.5 to drive the edit. Skeleton guidance is fixed at 1.0 for both branches.

Perceptual Evaluation. Figure 7 provides a visual comparison of editing results. Given the subjective nature of motion editing, we conducted an A/B user study ($N = 76$ valid participants). Participants evaluated 20 motion pairs across 6 survey sets using 3 criteria: **Source Preservation**, measuring consistency of unedited parts; **Edit Accuracy**, measuring alignment with the target text; and **Overall Quality**, measuring realism and smoothness.

As summarized in Figure 4, users preferred our model in **70% of comparisons** on average. These results demonstrate a robust preference for our flow-based transport over mask-based alternatives. Unlike MDM, our method is fully automatic and does not require manual masking, offering a significant advantage in usability.

4.5 Motion Retargeting

We further evaluate the model’s ability to transfer motion between characters with identical topological graphs but significantly different bone lengths and proportions.

Inference Configuration. While structural retargeting is robust across a range of parameters, we found that maintaining skeletal guidance scales at $w_{\text{skel}} = 1.0$ for both source and target branches yields the most consistent structural fidelity. We use a 100-step

Table 3. Retargeting evaluation on the Mixamo test split.

Method	Error ($\times 10^3$)↓
Copy	15.89
SAN [Aberman et al. 2020]	17.52
SAME [Lee et al. 2023]	13.47
R2ET [Zhang et al. 2023a]	8.13
Ours w/o jnt attn, direct prediction	5.73
Ours, direct prediction	5.24
Ours, FK reconstruction	4.91

integrator and sweep the start step between 5 and 40 (stride 5) to select the optimal trade-off between source motion preservation and target skeletal adaptation for each pair.

Baselines & Metrics. We compare against three retargeting methods: *SAN* [Aberman et al. 2020] (without the adversarial loss), *SAME* [Lee et al. 2023], and *R2ET* [Zhang et al. 2023a] (using only its skeleton retargeting module). We also report a copy baseline, which naively applies the source joint rotations and root translations to the target skeleton. Quantitative performance is measured using *Mean Squared Error (MSE)* on global positions, normalized by character height to account for scale differences.

Quantitative Results. Table 3 presents the comparative results. UniMotion achieves the *lowest MSE across all methods*, significantly outperforming both the copy baseline and dedicated retargeting networks. We report errors for two decoding strategies: *direct prediction*, which is reconstructing from the position channels ($p_{t,j}$ in Eq. 2), and *FK reconstruction*, recomputing positions from predicted rotations applied to the target skeleton. The low error in the direct prediction variant confirms that the model successfully learns the correlation between the source and target skeletons. It is not merely “hallucinating” rotations; it is generating a physically consistent pose sequence that respects the target’s bone constraints. Furthermore, the ablation study (Table 3, bottom rows) underscores the critical role of our *joint token* architecture. Figure 6 further visualizes retargeting outcomes across different methods.

5 Conclusion and Limitations

We introduced a unified motion generation model jointly conditioned on text and skeleton, enabling text-based motion editing and retargeting with the same inference procedure. This shared framework simplifies production workflows and enables new interactive editing tools. Our current implementation assumes a consistent joint ordering and requires a well-defined rest pose for each target character. Extreme morphology changes (e.g., non-humanoid characters) and highly out-of-distribution motions (e.g., climbing) still remain challenging. Besides, very long text prompts can introduce small velocity differences during text-based editing, resulting in subtle edits.

References

- Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020. Skeleton-aware networks for deep motion retargeting. *ACM Trans. Graph.* 39, 4, Article 62 (Aug. 2020), 14 pages. doi:10.1145/3386569.3392462
- Adobe Systems Inc. 2018. Mixamo. <https://www.mixamo.com> Accessed: 2017-09-28; 2018-12-27.
- Nikos Athanasiou, Alpar Ceske, Markos Diomataris, Michael J. Black, and Gül Varol. 2024. MotionFix: Text-Driven 3D Human Motion Editing. In *SIGGRAPH Asia 2024 Conference Papers*.
- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibao Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. 2025. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631* (2025).
- Zhefeng Cao, Ben Liu, Sen Li, Wei Zhang, and Hua Chen. 2025. G-DReaM: Graph-conditioned Diffusion Retargeting across Multiple Embodiments. *arXiv:2505.20857* [cs.RO] <https://arxiv.org/abs/2505.20857>
- Ling-Hao Chen, Shunlin Lu, Wenxun Dai, Zhiyang Dou, Xuan Ju, Jingbo Wang, Taku Komura, and Lei Zhang. 2025. Pay Attention and Move Better: Harnessing Attention for Interactive Motion Generation and Training-free Editing. *arXiv:2410.18977* [cs.CV] <https://arxiv.org/abs/2410.18977>
- Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. 2023. Executing your Commands via Motion Diffusion in Latent Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18000–18010.
- Manolo Canales Cuba, Vinícius do Carmo Melício, and João Paulo Gois. 2025. FlowMotion: Target-Predictive Conditional Flow Matching for Jitter-Reduced Text-Driven Human Motion Generation. *arXiv:2504.01338* [cs.GR] <https://arxiv.org/abs/2504.01338>
- Ke Fan, Shunlin Lu, Minyue Dai, Runyi Yu, Lixing Xiao, Zhiyang Dou, Junting Dong, Lizhuang Ma, and Jingbo Wang. 2025. Go to Zero: Towards Zero-shot Motion Generation with Million-scale Data. *arXiv:2507.07095* [cs.CV] <https://arxiv.org/abs/2507.07095>
- Inbar Gat, Sigal Raab, Guy Tevet, Yuval Reshef, Amit Haim Bermano, and Daniel Cohen-Or. 2025. AnyTop: Character Animation Diffusion with Any Topology. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, Article 13, 10 pages. doi:10.1145/3721238.3730621
- Michael Gleicher. 1998. Retargeting motion to new characters. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 33–42. doi:10.1145/280814.280820
- Chuan Guo, Inwoo Hwang, Jian Wang, and Bing Zhou. 2025. SnapMoGen: Human Motion Generation from Expressive Texts. *arXiv:2507.09122* [cs.CV] <https://arxiv.org/abs/2507.09122>
- Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. 2024. MoMask: Generative Masked Modeling of 3D Human Motions. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1900–1910. doi:10.1109/CVPR52733.2024.00186
- Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Jingyu Li, and Li Cheng. 2022. Generating Diverse and Natural 3D Human Motions From Text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5152–5161.
- Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. 1993. *Solving Ordinary Differential Equations I: Nonstiff Problems* (2 ed.). Springer Series in Computational Mathematics, Vol. 8. Springer.
- Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2022. Prompt-to-prompt image editing with cross attention control. (2022).
- Seokhyeon Hong, Chaelin Kim, Serin Yoon, Junghyun Nam, Sihun Cha, and Junyong Noh. 2025. SALAD: Skeleton-aware Latent Diffusion for Text-driven Motion Generation and Editing. *arXiv:2503.13836* [cs.CV] <https://arxiv.org/abs/2503.13836>
- Yiheng Huang, Yang Hui, Chuanchen Luo, Yuxi Wang, Shibao Xu, Zhaoxiang Zhang, Man Zhang, and Junran Peng. 2024a. StableMoFusion: Towards Robust and Efficient Diffusion-based Motion Generation Framework. *arXiv preprint arXiv: 2405.05691* (2024).
- Yiming Huang, Weilin Wan, Yue Yang, Chris Callison-Burch, Mark Yatskar, and Lingjie Liu. 2024b. Como: Controllable motion generation through language guided pose code editing. In *European Conference on Computer Vision*. Springer, 180–196.
- Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. 2025. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19721–19730.
- Sunmin Lee, Taeho Kang, Jungnam Park, Jehsee Lee, and Jungdam Won. 2023. SAME: Skeleton-Agnostic Motion Embedding for Character Animation. In *SIGGRAPH Asia 2023 Conference Papers* (Sydney, NSW, Australia) (SA '23). Association for Computing Machinery, New York, NY, USA, Article 45, 11 pages. doi:10.1145/3610548.3618206
- Zhengyuan Li, Kai Cheng, Anindita Ghosh, Uttaran Bhattacharya, Liangyan Gui, and Aniket Bera. 2025. SimMotionEdit: Text-Based Human Motion Editing with Motion Similarity Prediction. (2025). *arXiv:2503.18211* [cs.CV]
- Jongin Lim, Hyung Jin Chang, and Jin Young Choi. 2019. PMnet: Learning of Disentangled Pose and Movement for Unsupervised Motion Retargeting. In *British Machine Vision Conference (BMVC)*.
- Xingchao Liu, Chengyue Gong, and Qiang Liu. 2022. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003* (2022).
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. *arXiv:1711.05101* [cs.LG] <https://arxiv.org/abs/1711.05101>
- Zichong Meng, Yiming Xie, Xiaogang Peng, Zeyu Han, and Huaizu Jiang. 2024. Re-thinking Diffusion for Text-Driven Human Motion Generation. *arXiv preprint arXiv:2411.16575* (2024).
- Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang (Eds.). 2022. *Sentence-T5: Scaling up Sentence Encoder from Pre-trained Text-to-Text Transfer Transformer*. <https://aclanthology.org/2022.findings-acl.146/>
- William Peebles and Saining Xie. 2022. Scalable Diffusion Models with Transformers. *arXiv preprint arXiv:2212.09748* (2022).
- Mathis Petrovich, Michael Black, and Gül Varol. 2023. TMR: Text-to-Motion Retrieval Using Contrastive 3D Human Motion Synthesis. doi:10.48550/arXiv.2305.00976
- Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. 2023. FateZero: Fusing Attentions for Zero-shot Text-based Video Editing. *arXiv:2303.09535* (2023).
- Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. 2023. Human Motion Diffusion Model. In *The Eleventh International Conference on Learning Representations*. <https://openreview.net/forum?id=SJ1kSyO2jwu>
- Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2018. Neural Discrete Representation Learning. *arXiv:1711.00937* [cs.LG] <https://arxiv.org/abs/1711.00937>
- Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. 2018. Neural Kinematic Networks for Unsupervised Motion Retargeting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zijie Ye, Jia-Wei Liu, Jia Jia, Shikun Sun, and Mike Zheng Shou. 2024. Skinned Motion Retargeting with Dense Geometric Interaction Perception. *Advances in Neural Information Processing Systems* (2024).
- Jiaxu Zhang, Junwu Weng, Di Kang, Fang Zhao, Shaoli Huang, Xuefei Zhe, Linchao Bao, Ying Shan, Jue Wang, and Zhigang Tu. 2023a. Skinned Motion Retargeting with Residual Perception of Motion Semantics & Geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13864–13872.
- Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. 2023b. T2M-GPT: Generating Human Motion from Textual Descriptions with Discrete Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Jia-Qi Zhang, Miao Wang, Fu-Cheng Zhang, and Fang-Lue Zhang. 2025. Skinned Motion Retargeting With Preservation of Body Part Relationships. *IEEE Transactions on Visualization and Computer Graphics* 31, 9 (2025), 4923–4936. doi:10.1109/TVCG.2024.3423426
- Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. 2022. MotionDiffuse: Text-Driven Human Motion Generation with Diffusion Model. *arXiv:2208.15001* [cs.CV] <https://arxiv.org/abs/2208.15001>

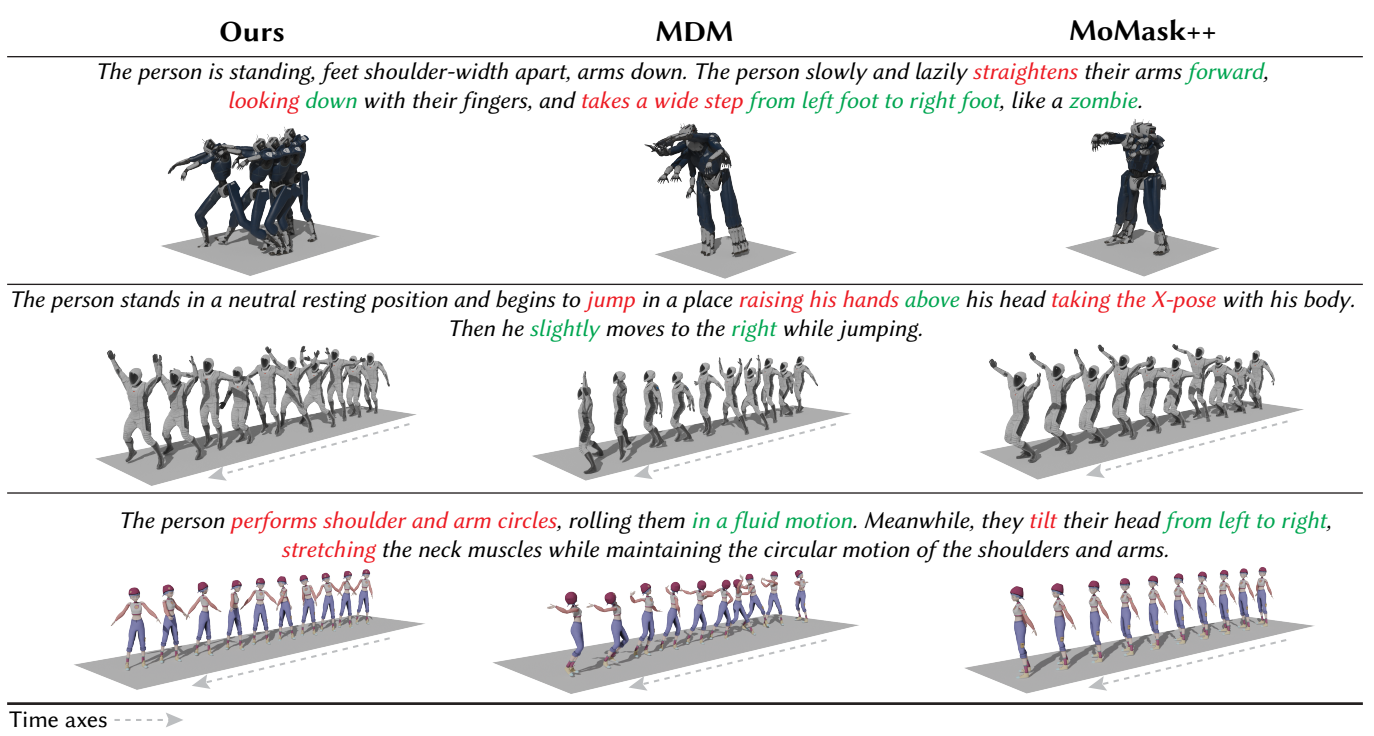


Fig. 5. Qualitative comparison on text-to-motion generation. For visualization, motions with little or no root translation are manually time-shifted. Prompt words in red denote actions, while words in green indicate motion modifiers. Our model successfully synthesizes coherent full-body motions that faithfully reflect fine-grained textual details over long time spans. This temporal coherence is critical for our downstream editing tasks, where the model must preserve the narrative structure of the source motion.

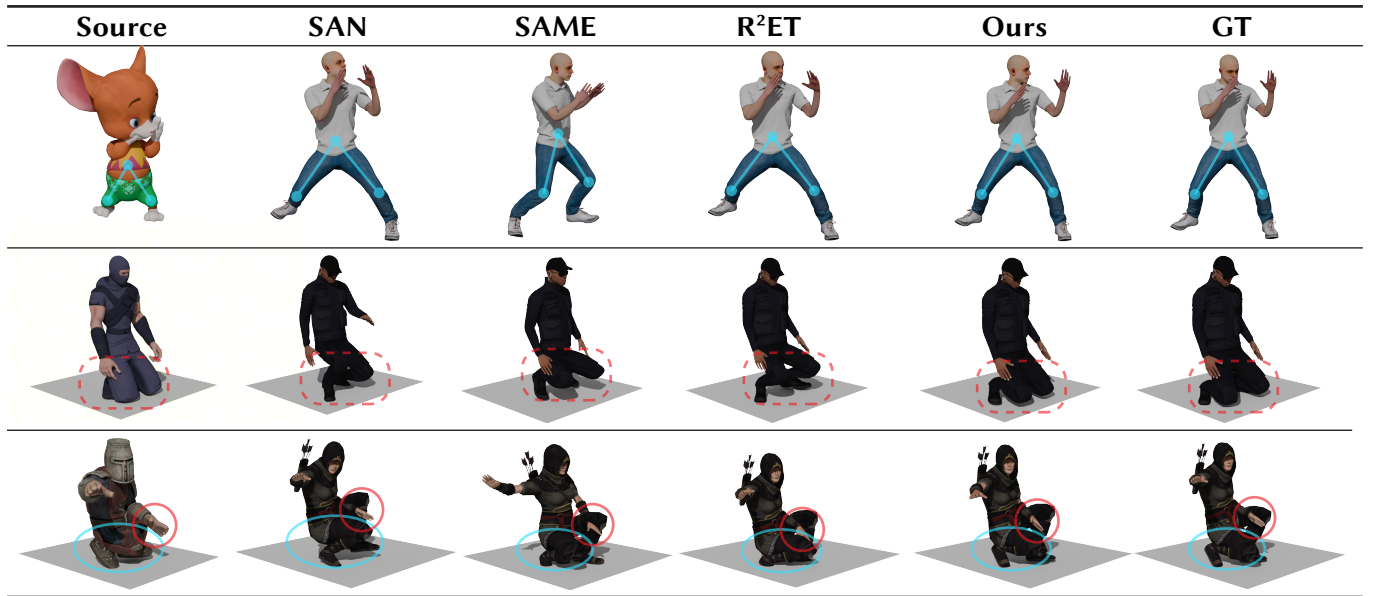


Fig. 6. Qualitative retargeting comparison. We compare against SAN [Aberman et al. 2020], SAME [Lee et al. 2023], and R²ET [Zhang et al. 2023a]. The proposed method better preserves fine-grained local motion and adapts to varying skeleton proportions. As highlighted in the zoomed-in regions, baseline methods often introduce artifacts such as unnatural twisting or over-stretching when adapting to different skeletons. In contrast, UniMotion preserves delicate local details (e.g., leg bending scale, hand facing direction) while naturally adapting the global motion to fit the new body shape.

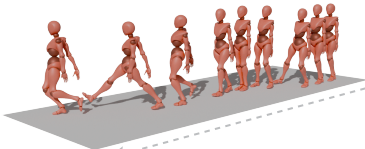
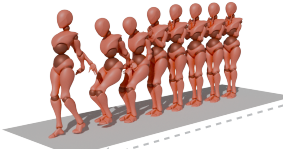
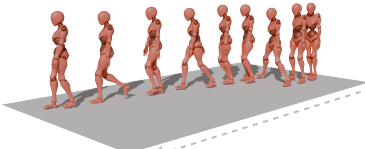
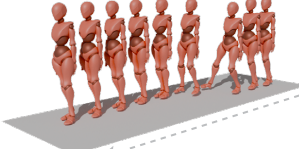
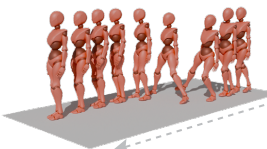
Source Text	The person stands and waves the hands.	A person takes a long step to their right. Then they turn to their left and walk forward , stopping and standing straight in a neutral resting position.	A person stands in a neutral resting position. Then he leans his torso forward slightly, gathers his arms, bent at the elbows, in front of his waist , then outstretches his arms apart and up playfully, and putting his right leg forward finally.
Edit Text	The person jumps and waves the hands.	A person takes a long step to their right. Then they stand still and do not move.	A person stands in a neutral resting position. Then he leans his torso backward slightly, gathers his arms, bent at the elbows, in front of his head , then outstretches his arms apart and up playfully, and putting his right leg forward finally.
Edit Type	Modify action	Add/Delete actions	Body part modification
Source Motion			
MDM			
MoMask++			
Ablation			
Ours			
Time axes ----->			

Fig. 7. Qualitative results on text-based motion editing. Edited prompt words are highlighted in red and green. For visualization, motions shifted as in Fig. 5. Our method demonstrates superior adaptability across all edit types. Unlike mask-based approaches (MDM) which can introduce stiffness at mask boundaries, or regenerative baselines (MoMask++) which often drift from the source semantics, UniMotion produces accurate, coherent edits.