SUPERMOTION: SOURCE-PRESERVING DENOISING FOR TEXT-DRIVEN HUMAN MOTION EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-driven human motion editing aims to realize a requested change while preserving compatible source content. Existing diffusion editors rely largely on learned conditioning for preservation of the unedited part, yet their outputs can lose temporal detail as denoising proceeds. We propose the **Source-Preserving Denoising framework (SuperMotion)**, which explicitly reuses the source at each reverse step for source preservation. We first align the source motion to the output timeline and predict a preservation gate that controls reuse across frames and feature dimensions. A clean-space source anchor then utilizes the learned preservation gate to blend the predicted clean motion with the aligned source and passes the corrected estimate directly to the sampling posterior. Because the aligned source is a realized motion rather than a regression output, the anchor injects sample-level temporal detail that a reconstruction-trained denoiser tends to smooth away. To learn effective source reuse, we supervise the anchored estimate against the editing target and match its second temporal differences through a temporal high-frequency loss. These objectives require no explicit edit masks. Extensive experiments show that SuperMotion improves editing accuracy, reaching 33.20% full-pool R@1 on MotionFix, while reducing temporal-detail attenuation and preserving motion dynamics as it realizes the requested changes. Ablations confirm that the learned preservation gate is responsible for the gain and that it reuses the source to retain the unedited content properly.

1 INTRODUCTION

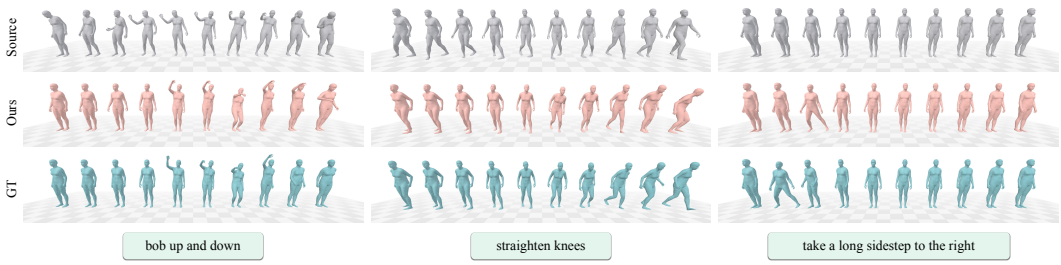


Figure 1: **Motion editing requires both change and preservation.** Given a source motion (gray), the edit should realize the requested modification while retaining compatible source content. Our outputs (salmon) and reference targets (turquoise) illustrate three instructions, "bob up and down," "straighten knees," and "take a long sidestep to the right." Each strip shows successive poses.

Text-driven human motion editing modifies an existing movement according to a textual instruction. A request such as "straighten knees" specifies a change while preserving other aspects of the source movement (Figure 1). A successful edit must therefore balance instruction compliance with preservation of compatible source content. This balance requires more than copying a predefined region. Source and reference target motions may differ in duration, coordination, and execution, so content that should be retained cannot be preserved exactly, but only approximately. The editor must determine how to use the source while accommodating the requested change.

Advances in text-conditioned motion generation provide a foundation for motion editing (Petrovich et al., 2022; Tevet et al., 2022b; Zhang et al., 2024; Guo et al., 2024). Existing editors build on these models by improving how they use the source motion and instruction, learning motion similarity, and

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.

introducing objectives for editing body parts and preserving source content (Athanasiou et al., 2024; Li et al., 2025; Guo et al., 2025; Yang et al., 2025; Shi et al., 2026). These methods can produce the requested action, but the resulting movement may be too subdued or too exaggerated compared with the reference target. Retrieval scores measure how well the generated motion matches the target in a learned representation, but do not directly measure these differences in movement. We therefore examine how temporal detail changes in the generated motion and throughout denoising.

Concretely, we define a *detail ratio*: the magnitude of second-order temporal differences of the running clean-motion estimate $\hat{x}_0$ at a given reverse step, normalized by the same statistic computed on the ground-truth target. A well-calibrated reverse process should drive this ratio toward $1\times$ and hold it there. Figure 2 shows that prior diffusion editors behave otherwise: the median ratio decreases along the reverse trajectory, crosses $1\times$, and keeps falling, reaching $0.33\times$ for TMED (Athanasiou et al., 2024), $0.66\times$ for OmniME (Shi et al., 2026), and $0.71\times$ for SimMotionEdit (Li et al., 2025). The decline concentrates in the final low-noise steps, where the estimate already matches the target detail level and further denoising suppresses local temporal variation instead of refining it. A denoiser trained with a reconstruction objective approximates a posterior mean (Efron, 2011), and such distortion-optimal estimates are necessarily smoother than individual samples (Blau & Michaeli, 2018). The effect is especially costly in motion editing. The content that should be preserved already exists in the source, but regenerating the sequence from noise forces the model to resynthesize it, after which it is pulled toward the mean.

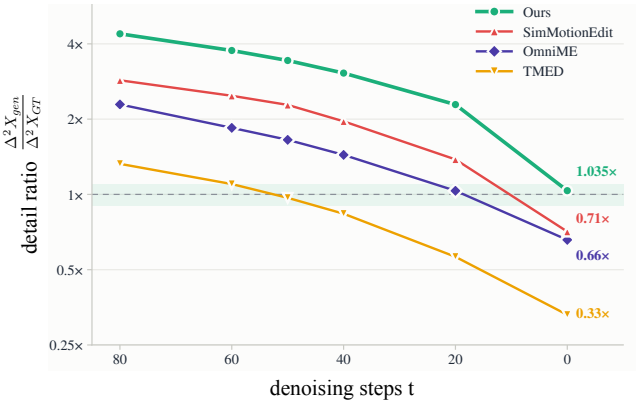


Figure 2: **Detail attenuation during denoising.** Median generated-to-target second-order difference magnitude at selected timesteps, with denoising proceeding from left to right. The plotted baselines finish below the target reference of one, while the shown SuperMotion run ends at $1.035\times$. Shading marks a fixed reference band from 0.9 to 1.1. Agreement in this statistic alone does not establish a correct edit.

These observations motivate explicit control over source reuse during denoising. In frameworks that reconstruct the target from noise, source conditioning provides information about the input motion, but retaining its content still depends on the denoiser's predictions (Athanasiou et al., 2024; Li et al., 2025). The source could also provide a clean signal against which those predictions are adjusted at each reverse step. Related ideas appear in diffusion-based image editing and inpainting, where spatial masks or known regions guide the preservation of existing content (Avrahami et al., 2022; Lugmayr et al., 2022; Couairon et al., 2022). Applying this principle to motion editing requires accounting for differences between the source and target sequences. Because no preservation mask is supplied, the model must learn how strongly to reuse the aligned source while allowing the instruction to change the movement.

We introduce the **So**u**r**ce-**P**res**er**ving Denoising framework (SuperMotion) to learn this balance within the reverse process. Its central component, a *Clean-Space Source Anchor*, blends the current clean-motion estimate with a temporally aligned source through a learned *Source Preservation Gate*. The gate assigns a reuse weight to each frame and feature dimension at each denoising step. Applied after classifier-free guidance and clean-motion reconstruction, the anchor produces the estimate used directly by the sampling posterior. This placement gives the gate a consistent interpretation as a convex blending weight and avoids the inverse noise-scale factor required to express the same clean-motion correction in velocity space. Supervision against the editing target trains the gate to adjust source reuse without an explicit edit mask. We train SuperMotion with a source-anchor loss that learns source reuse from the editing target and a temporal high-frequency loss that matches second temporal differences.

We conduct extensive experiments to evaluate our method. The results show that SuperMotion improves instruction compliance while reducing temporal-detail attenuation. Ablation studies show that supervising the anchored prediction against the editing target is critical to learning effective source reuse. Our contributions are threefold.

1. We identify temporal-detail attenuation as a systematic failure of diffusion motion editors. Their clean-motion estimates lose second-order temporal variation during late denoising, and several published editors score below a copy-source baseline.

2. We propose SuperMotion, which reuses a temporally aligned source at each reverse step through a learned per-frame, per-dimension gate on the clean-motion estimate, trained from the editing target without edit masks.

3. SuperMotion reaches 79.17% batch R@1 and 33.20% full-pool R@1 on MotionFix and 46.88% batch R@1 on STANCE Adjustment, surpassing the strongest prior editor by 1.88, 1.18, and 3.13 points. It restores the detail ratio to 1.035, against 0.66 to 0.71 for prior editors, confirming that the learned preservation gate recovers high-frequency detail from the source motion.

2 Related Work

Human Motion Generation. Text-driven human motion generation has progressed through continuous latent modeling, diffusion-based synthesis, and discrete motion modeling. Guo et al. (Guo et al., 2022) investigate diverse text-conditioned motion synthesis and introduce HumanML3D, supporting subsequent advances in language-driven animation. TEMOS (Petrovich et al., 2022) learns compatible text and motion distributions through variational training, while MotionCLIP (Tevet et al., 2022a) aligns motion representations with the CLIP embedding space. Diffusion-based methods, including MDM (Tevet et al., 2022b), MotionDiffuse (Zhang et al., 2024), and FLAME (Kim et al., 2023), generate motions through text-conditioned iterative denoising. MLD (Chen et al., 2023) instead performs diffusion in a compact latent space to reduce computational cost. Discrete motion representations provide an alternative to continuous-space generation. T2M-GPT (Zhang et al., 2023a) combines vector quantization with autoregressive token prediction, while MotionGPT (Jiang et al., 2023) models motion and language within a unified token-based framework. MoMask (Guo et al., 2024) and MMM (Pinyoanuntapong et al., 2024) adopt masked motion modeling, enabling iterative token reconstruction and flexible generation. Beyond global text conditioning, FineMoGen (Zhang et al., 2023b) introduces fine-grained spatio-temporal control, and MaskControl (Pinyoanuntapong et al., 2025) incorporates spatial constraints into masked motion generation. Together, these approaches improve the diversity, efficiency, and controllability of human motion synthesis.

Our focus is source-conditioned editing rather than synthesizing a new realization from text alone. SuperMotion retains a diffusion backbone and introduces a learned source-reuse operator to preserve compatible motion content during denoising.

Human Motion Editing. Human motion editing modifies an existing sequence while retaining content compatible with the requested change. Early generative approaches enable editing through semantic latent manipulation or constrained reconstruction. MotionCLIP (Tevet et al., 2022a) supports editing in a language-aligned latent space, while MDM (Tevet et al., 2022b) and FLAME (Kim et al., 2023) support motion completion and editing through constrained diffusion. MMM (Pinyoanuntapong et al., 2024) reconstructs masked motion tokens, and FineMoGen (Zhang et al., 2023b) supports fine-grained editing through spatio-temporal descriptions. These approaches establish useful editing capabilities, although their control interfaces and supervision differ from paired instruction-based editing. MotionFix (Athanasiou et al., 2024) establishes a benchmark of source–instruction–target triplets and introduces a source-conditioned diffusion editor. Subsequent work improves instruction alignment, motion preservation, and editing versatility. SimMotionEdit (Li et al., 2025) jointly trains motion editing and motion-similarity prediction, while PartMotionEdit (Yang et al., 2025) combines part-aware modulation with part-level similarity supervision and bidirectional text–motion interaction. MotionReFit (Jiang et al., 2025) expands editing supervision through MotionCutMix and uses an autoregressive diffusion model with a motion coordinator. MotionLab (Guo et al., 2025) unifies generation and editing through a motion–condition–motion formulation based on rectified flows. OmniME (Shi et al., 2026) combines intermediate-feature, preservation, and positive–negative semantic supervision. Recent work further considers both spatial preservation and temporal rhythm through non-uniform temporal alignment (Lin et al., 2026).

SuperMotion uses a learned, feature-wise source anchor to modify the clean-motion estimate after guidance and before each posterior update. Its distinction lies in this explicit sampling-time operator, rather than preservation supervision or source blending as a general objective.

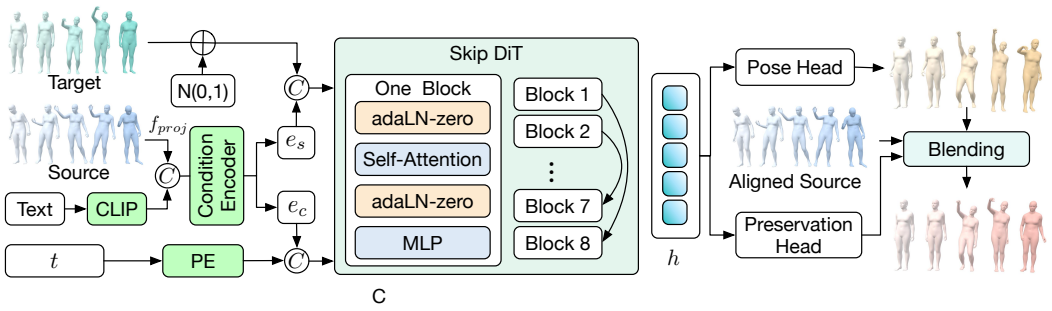


Figure 3: Overall architecture. A Skip DiT takes the noised motion x_t, a condition encoder over source and text, and the diffusion timestep. Shared target tokens feed a pose head that predicts velocity v and a preservation head that predicts a per-frame, per-dimension gate g. After converting v to $\hat{x}_0$, blending pulls the estimate toward the time-aligned source where g is high and leaves it to the pose head where g is low.

3 METHOD

Given a source motion and an editing instruction, our goal is to generate a motion that realizes the requested change while retaining compatible source content. Our Source-Preserving Denoising (SuperMotion) makes source reuse an explicit operation inside the reverse process, applied to the clean-motion estimate at every step and controlled by a gate learned from the editing target alone.

3.1 PROBLEM FORMULATION

Text-based 3D motion editing takes a source sequence $s \in \mathbb{R}^{T_s \times D}$ and a natural-language instruction c, and produces a motion $\hat{x}_0 \in \mathbb{R}^{T \times D}$ that carries out c. We use the MotionFix feature of D=207 z-normalized dimensions per frame (pelvis translation increments, 6D orientation, z-orientation increments, 21-joint body pose, and 22 local joint coordinates) at 30 fps.

The instruction typically changes only part of the source. Over the remaining frames and coordinates the target coincides with the source, so the underlying edit is close to an identity mapping. Diffusion editors nevertheless regenerate the full sequence from Gaussian noise with s and c used only as conditions, so every reverse step re-synthesizes coordinates that could have been copied. A denoiser trained with a reconstruction loss approximates the posterior mean $\mathbb{E}[x_0 \mid x_t]$, and such distortion-optimal estimates are smoother than any individual sample (Blau & Michaeli, 2018). The identity part of the edit is therefore re-estimated from noise at every step and pulled toward a mean it does not belong to.

Let x_0 be the clean target motion and

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I) \tag{1}$$

the corresponding noised state at diffusion time t. A reverse step must recover a clean estimate from x_t that realizes c and remains close to the source wherever c does not ask for a change. SuperMotion enforces the second requirement as an operation on the clean estimate rather than leaving it to conditioning.

3.2 ARCHITECTURE OVERVIEW

As shown in Figure 3, our implementation builds on SimMotionEdit (Li et al., 2025). A four-layer Transformer condition encoder maps the linear projection of source motion s and the CLIP ViT-L/14 text embedding (Radford et al., 2021) of c to token sequences e_s and e_c, respectively. The projected noisy-motion tokens are concatenated with e_s and processed by a diffusion Transformer of eight DiT blocks (Peebles & Xie, 2023) with hidden width 512 and long skip connections between symmetric blocks following U-ViT (Bao et al., 2023). After removing the source tokens, the target features $h \in \mathbb{R}^{T \times 512}$ feed two heads. All quantities computed from x_t depend on the timestep, which we leave implicit.

The pose head outputs $\hat{v} \in \mathbb{R}^{T \times D}$ under the v-parameterization (Salimans & Ho, 2022), giving the clean-motion estimate

$$\hat{x}_0 = \sqrt{\bar{\alpha}_t}\, x_t - \sqrt{1-\bar{\alpha}_t}\, \hat{v}. \tag{2}$$

Unlike SimMotionEdit, which predicts x_0 directly, we predict velocity v. The velocity loss equals $(\mathrm{SNR}(t)+1)\,\|x_0-\hat{x}_0\|_2^2$ with $\mathrm{SNR}(t)=\bar{\alpha}_t/(1-\bar{\alpha}_t)$ (Kingma et al., 2021), and therefore places more weight on clean-motion errors at low noise, where the detail loss of Section 1 concentrates. The preservation head, an MLP on the same h, outputs the preservation gate, which indicates how much of each frame and pose coordinate should be retrieved from the aligned source. Then, we use the predicted gate value to blend the estimation $\hat{x}_0$ with the aligned source motion (Section 3.3).

At inference, we use classifier-free guidance (Ho & Salimans, 2022) with separate source and text scales, following MotionFix (Athanasiou et al., 2024).

$$\hat{v}_{\mathrm{cfg}} = \hat{v}_{\varnothing} + w_s\big(\hat{v}_s - \hat{v}_{\varnothing}\big) + w_c\big(\hat{v}_{s,c} - \hat{v}_s\big), \tag{3}$$

where w_s and w_c scale the source and text guidance, respectively. We convert $\hat{v}_{\mathrm{cfg}}$ to $\hat{x}_0$ by Eq. 2 and apply the anchor with the gate from the fully conditioned branch.

3.3 Source-Preserving Denoising

Aligning the source. Source and target may differ in duration. We linearly resample the source onto the output timeline,

$$\bar{s} = \mathrm{Align}(s, T) \in \mathbb{R}^{T\times D}, \tag{4}$$

which yields a reference of the same length as $\hat{x}_0$ without assuming that corresponding frames contain identical movements. Residual misalignment is absorbed by the gate.

Preservation gate. From the target-token features, the preservation head predicts

$$g = \mathrm{sigmoid}\big(f_\phi(h)\big), \qquad g \in (0,1)^{T\times D}, \tag{5}$$

where f_ϕ is an MLP. Each entry is the source-reuse weight for one frame and one feature dimension, and the dependence on t through h lets reuse vary along the reverse trajectory.

Clean-space source anchor. The anchor forms the corrected estimate

$$\widetilde{x}_0 = \hat{x}_0 + g \odot (\bar{s} - \hat{x}_0) = (1-g)\odot \hat{x}_0 + g \odot \bar{s}. \tag{6}$$

Because $g \in (0,1)$, every corrected coordinate lies between the denoiser prediction and the aligned source. The anchor addresses the detail loss of Section 1 directly. The learned estimate $\hat{x}_0$ is trained as a regressor and inherits the smoothing of a posterior mean, whereas $\bar{s}$ is a realized motion and carries sample-level temporal detail. On coordinates the instruction leaves unchanged, $\bar{s}$ is close to x_0, so Eq. 6 injects detail the target should have without asking the network to regenerate it. Conditioning provides the same information, but only as network input. Recovering the source then requires the denoiser to learn an identity map through its full depth, which Figure 2 in Section 1 shows it does not reliably do. The anchor bypasses this by blending the estimate with the source at a learned, per-coordinate strength.

Optimal gate. Holding $\hat{x}_0$ fixed and minimizing the anchor loss of Eq. 9 over $g \in (0,1)^{T\times D}$, the closure of the sigmoid range, decouples across coordinates. Each coordinate is a convex quadratic in $g_{i,d}$, so its constrained minimizer is the projection of the unconstrained one onto $[0,1]$,

$$g^*_{i,d} = \mathrm{clip}\left(\frac{x_{0,i,d} - \hat{x}_{0,i,d}}{\bar{s}_{i,d} - \hat{x}_{0,i,d}},\, 0,\, 1\right), \tag{7}$$

defined wherever $\bar{s}_{i,d} \neq \hat{x}_{0,i,d}$ (elsewhere the anchor is inactive and $g_{i,d}$ is immaterial). The sigmoid gate takes values in (0, 1), so it approaches an optimum at 0 or 1 without reaching it. The unclipped ratio is the fraction of the way from the denoiser prediction to the source at which the target lies. It equals one where the target coincides with the source and zero where it coincides with the prediction. The bound $(0,1)$ is a design choice and we ablate other choices in Sec. 4.3. Where the target lies outside the segment between prediction and source, the gate saturates and the residual is left to the denoiser. Within this bound, Eq. 7 makes two predictions about the learned gate. First, writing the edit as $\delta = x_0 - \bar{s}$ gives $g^* = \mathrm{clip}\big(1 + \delta/(\bar{s} - \hat{x}_0),\, 0,\, 1\big)$, so an edit can only lower the optimum below one and never raise it, and larger edits tend to yield a smaller optimal gate. Second, the optimum vanishes as $t \to 0$. The denoiser prediction approaches the target, the numerator goes to zero wherever $\bar{s} \neq x_0$, and the gate releases the source in the final steps regardless of how much it reused earlier. The learned gate reproduces both behaviors.

Table 1: Comparison on MotionFix.

Method	B R@1↑	B R@2↑	B R@3↑	B AvgR↓	FR@1↑	FR@2↑	FR@3↑	F AvgR↓
Copy source	74.79	85.42	91.25	1.88	30.24	44.47	54.35	15.53
MDM (Tevet et al., 2022b)	4.03	7.56	10.48	15.55	0.10	0.10	0.10	-
MDM-BP (Tevet et al., 2022b)	39.10	50.09	54.84	6.46	8.69	14.71	18.36	180.99
MotionLab (Guo et al., 2025)	54.53	70.02	77.38	3.42	-	-	-	-
TMED (Athanasiou et al., 2024)	62.90	76.51	83.06	2.71	14.51	21.72	28.73	56.63
MotionReFit (Jiang et al., 2025)	66.63	80.05	84.98	2.64	14.13	23.52	30.53	54.06
SimMotionEdit (Li et al., 2025)	70.62	82.92	88.12	2.38	25.49	39.33	49.21	23.49
PartMotionEdit (Yang et al., 2025)	73.96	85.83	90.21	1.92	27.27	45.06	53.36	16.24
Han et al. (Han & Kim, 2026)	74.38	88.54	92.08	1.92	29.45	45.26	54.55	16.42
OmniME (Shi et al., 2026)	77.29	88.54	91.88	1.79	32.02	50.20	**59.88**	13.06
Ours	**79.17**	**89.17**	**92.92**	**1.68**	**33.20**	**50.79**	59.09	**12.87**

Anchoring within reverse diffusion. The corrected estimate replaces the clean target in the Gaussian posterior,

$$p_{\theta,\phi}(x_{t-1} \mid x_t, s, c) = q(x_{t-1} \mid x_t, \widetilde{x}_0), \tag{8}$$

so source content enters the sample at every reverse step rather than only through the network input. We apply the correction to $\hat{x}_0$ rather than to $\hat{v}$. The two are related by a factor $(1-\bar{\alpha}_t)^{-1/2}$ that grows without bound as $t \to 0$. Anchoring in clean space keeps g a bounded convex weight at every timestep.

3.4 Training Objective

In addition to the velocity prediction loss, we apply two losses to the anchored estimate $\widetilde{x}_0$. We write them for a single target $x_0 \in \mathbb{R}^{T\times D}$ with T valid frames, index frames by i and dimensions by d, and exclude padded frames.

Source-anchor loss. We supervise the corrected prediction against the target,

$$\mathcal{L}_{\text{anc}} = \frac{1}{TD}\sum_{i=1}^{T} \|\widetilde{x}_{0,i} - x_{0,i}\|_2^2 . \tag{9}$$

Because $\widetilde{x}_0$ depends on both heads, this loss trains the denoiser and the gate jointly, and its per-coordinate minimizer over g is Eq. 7. The editing target thus supervises source reuse without an explicit preservation mask.

Temporal high-frequency loss. Reconstruction alone does not constrain local temporal variation. We match second-order temporal differences between the anchored prediction and the target. With $\Delta^2 x_i = x_{i+1} - 2x_i + x_{i-1}$,

$$\mathcal{L}_{\text{hf}} = \frac{1}{(T-2)D}\sum_{i=2}^{T-1} \left\|\Delta^2\widetilde{x}_{0,i} - \Delta^2 x_{0,i}\right\|_2^2 . \tag{10}$$

This term penalizes both excess and deficit of variation at each frame and dimension, and its gradient reaches the gate through the anchor.

Overall objective. We retain the velocity loss $\mathcal{L}_v$, a masked mean squared error within each feature group averaged across groups, and the auxiliary similarity loss $\mathcal{L}_{\text{cls}}$ from SimMotionEdit (Li et al., 2025). The complete objective is

$$\mathcal{L} = \mathcal{L}_v + \lambda_{\text{anc}}\mathcal{L}_{\text{anc}} + \lambda_{\text{hf}}\mathcal{L}_{\text{hf}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}}, \tag{11}$$

with $\lambda_{\text{anc}} = 3$, $\lambda_{\text{hf}} = 0.5$, and $\lambda_{\text{cls}} = 0.05$. Losses are aggregated over valid frames and averaged over sampled timesteps and noise.

4 Experiment

4.1 Dataset and Evaluation Metrics

Datasets. We evaluate on MotionFix (Athanasiou et al., 2024) and STANCE Adjustment from OmniME (Shi et al., 2026). MotionFix provides 6,730 (source, instruction, target) triplets: 5,387 for training, 330 for validation, and 1,013 for testing, at 30 fps. STANCE is a separate 20 fps benchmark with 4,411 triplets.

Table 2: Comparison on STANCE Adjustment.

Method	B R@1↑	B R@2↑	B R@3↑	B AvgR↓	FR@1↑	FR@2↑	FR@3↑	F AvgR↓
Copy source	40.10	58.33	64.58	5.13	17.35	22.96	30.10	27.23
MotionReFit (Jiang et al., 2025)	25.52	34.90	45.83	8.27	9.18	14.29	16.84	48.74
TMED (Athanasiou et al., 2024)	39.58	49.48	60.94	6.26	13.78	22.96	30.10	34.38
SimMotionEdit (Li et al., 2025)	43.23	59.38	68.23	4.67	20.92	**31.63**	37.24	24.02
OmniME (Shi et al., 2026)	43.75	56.25	66.15	4.66	**22.45**	**31.63**	36.22	22.77
Ours	**46.88**	**63.02**	**71.88**	**3.72**	18.37	31.12	**38.27**	**19.21**

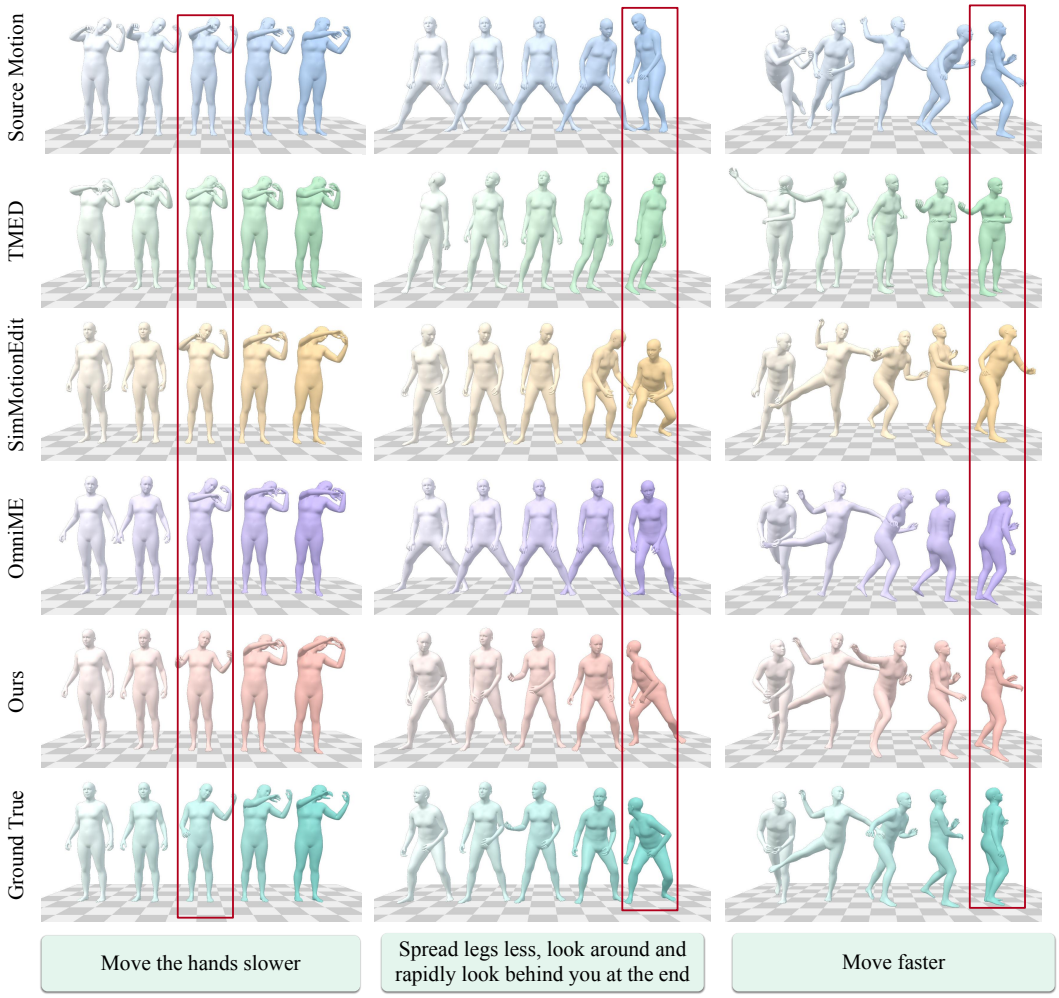


Figure 4: **Qualitative comparison on MotionFix dataset.** Relative to prior methods, ours follows the instruction on the first two examples and keeps the unedited motion on the third.

Motion retrieval. We use a pretrained TMR encoder (Petrovich et al., 2023) to rank reference target motions by their embedding similarity to each generated motion. We report R@1, R@2, R@3, and average rank for batch retrieval and full-pool retrieval as previous methods (Athanasiou et al., 2024; Shi et al., 2026). Higher recall and lower average rank indicate better performance.

Implementation details. We build on SimMotionEdit (Li et al., 2025), using a frozen CLIP ViT-L/14 text encoder and a 207-dimensional motion representation at 30 fps. We initialize SuperMotion from the trained backbone and jointly train the denoiser and preservation head using AdamW with a learning rate of 10^{-4} and a batch size of 128 on a single NVIDIA A100 GPU. We use velocity prediction with a squared-cosine noise schedule and 300 diffusion steps. At inference, both source and text guidance scales are set to 3.0, and source anchoring is applied after guidance at every reverse step.

Table 3: Ablation study on different components (MotionFix).

Method	B R@1↑	B R@2↑	B R@3↑	B AvgR↓	FR@1↑	FR@2↑	FR@3↑	F AvgR↓
Ours	**79.17**	**89.17**	92.92	**1.68**	**33.20**	**50.79**	**59.09**	**12.87**
w/o gate	74.79	87.08	91.04	1.95	29.45	44.86	54.74	16.22
tanh gate	76.04	89.58	92.29	1.75	29.84	46.05	55.34	13.35
per-frame	74.17	89.17	**93.54**	1.79	29.64	46.44	57.11	14.12
w/o $\mathcal{L}_{\text{anc}}$	76.25	87.92	92.71	1.89	27.27	43.48	55.14	15.66
w/o $\mathcal{L}_{\text{hf}}$	76.88	88.54	91.46	1.75	30.24	44.66	55.14	13.61
w/o skip	76.25	89.70	92.29	1.73	28.06	43.48	53.16	13.70
w/o $\hat{x}_0$ (anchor in v)	76.67	87.92	92.29	1.88	31.03	46.64	54.35	16.25

4.2 Comparison with State-of-the-Art

Quantitative Comparison. Table 1 and Table 2 report quantitative comparisons. On MotionFix, our method ranks first on seven of eight metrics, with OmniME (Shi et al., 2026) marginally ahead on **FR@3**. A more informative reference is the *copy source* baseline, which returns the input unchanged and still reaches 74.79% **BR@1**. Most prior editors fall below it, indicating that regenerating the target from noise destroys more source content than the edit itself introduces. Only OmniME and our method exceed this baseline, and ours does so by the larger margin on both batch ($+4.38$) and full-pool ($+2.96$) R@1. On STANCE Adjustment, our method leads all batch-retrieval metrics and improves full-pool AvgR from 22.77 to 19.21. **FR@1** is lower than OmniME, while **FR@3** is higher and full-pool AvgR is lower, indicating fewer severe ranking failures rather than more top-1 matches. Across the two benchmarks, SuperMotion is the only method that exceeds the copy-source baseline on every retrieval metric on both benchmarks.

Qualitative Comparison. Figure 4 presents qualitative comparisons on MotionFix. The first two examples assess instruction compliance. For *move the hands slower*, the baselines keep the arm moving at the source speed and leave its trajectory unchanged, whereas our result reduces the hand velocity while keeping the arm path and body configuration of the source. For *spread legs less, look around and rapidly look behind you at the end*, which combines a continuous adjustment with a terminal action, only our method produces both, with the final turn matching the ground-truth timing. The third example, "*move faster*", instead probes source preservation. The instruction changes only the temporal profile, yet TMED and SimMotionEdit resynthesize the limb trajectories and drift from the source configuration. Our result accelerates the sequence while retaining the original motion pattern, consistent with the detail-ratio behavior reported in Section 1.

4.3 Ablation Study

Setup. Table 3 ablates each component on MotionFix. All variants are trained with the same schedule and seed.

Effectiveness of the preservation gate. Fine-tuning the base model for the same schedule without the preservation gate g ("*w/o gate*" in Table 3) reduces **FR@1** from 33.20 to 29.45 and raises **F AvgR** from 12.87 to 16.22. The improvement, therefore, stems from the proposed mechanism rather than from additional optimization. To verify the effectiveness of the range of the preservation gate, we replace the sigmoid gate $g = \sigma(\cdot) \in (0, 1)$ with $\tanh$, so $g \in (-1, 1)$ ("*tanh gate*" in Table 3). Positive g still copies the source; negative g pushes $\hat{x}_0$ away from it. For "tanh gate" **FR@1** falls to 29.84, near the no-gate control ("w/o gate" with 29.45). The gate has to be one-sided. Restricting the correction to preservation, rather than introducing an additional degree of freedom, is therefore essential. In our framework, we predict a per-frame, per-dimension gate $g \in (0, 1)^{T \times D}$. "per-frame" replaces it with one scalar per frame, broadcast across all pose channels, so every joint shares the same copy strength at time t. This run reaches 29.64 **FR@1**, 3.56 below Ours (33.20), and trails on every full-pool metric, consistent with the learned gate keeping whole feature channels high or low across time, which a per-frame scalar cannot express.

Effectiveness of training objectives. "*W/o* L_{hf}" turns off the second-difference term in total objective. **FR@1** falls from 33.20 to 30.24, and batch R@1 from 79.17 to 76.88. These results verify that high-frequency loss is able to help model generate accurate motion instructed by input text. Removing the anchor loss (*w/o* $\mathcal{L}_{\text{anc}}$) is more harmful. The gate retains the capacity to reuse the source but receives no supervision on which coordinates admit reuse, since $\mathcal{L}_v$ does not propagate to it. **BR@1** stays at 76.25 while **FR@1** drops to 27.27, below the model without any gate value (29.45).

Structure Evaluation. Removing the U-ViT skips ("*w/o skip*" in Table 3) while keeping the gate lowers **BR@1** from 79.17 to 76.25 and **FR@1** from 33.20 to 28.06. The skips carry high-frequency motion features directly to the late blocks, and the preservation head depends on those features to decide where reuse is acceptable.

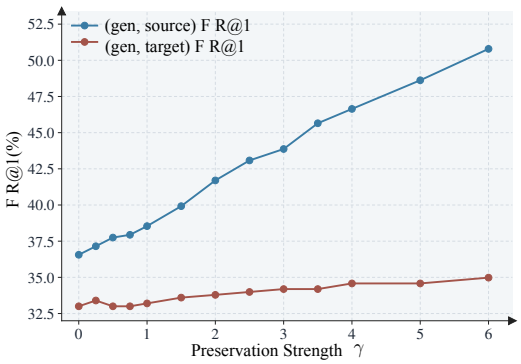
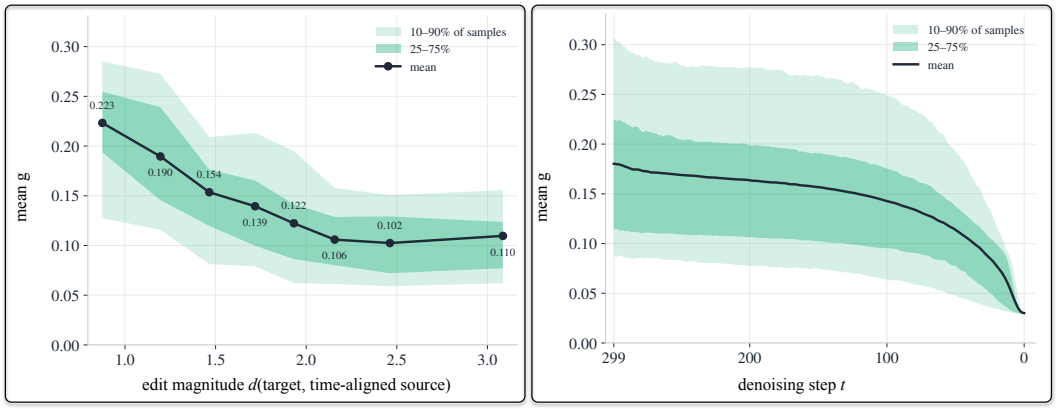


Figure 5: Preservation strength at inference. Scaling the learned gate by γ without retraining raises retrieval against the source far more than against the target.

Preservation strength. We scale the learned gate by γ at inference without retraining (Figure 5). At $\gamma = 0$ the anchor is removed, yet **FR@1** against the target stays at 33.0, close to 33.20 at $\gamma = 1$ and well above the model trained without the gate (29.45). Training through the anchor thus shapes the denoiser itself. Increasing γ to 6 improves **FR@1** against the target by only 1.78, from 33.20 to 34.98, but improves **FR@1** against the source by 12.25, from 38.54 to 50.79. At $\gamma = 6$ the generated motion retrieves its own source (50.79 **FR@1**) more accurately than its target (34.98 **FR@1**). The additional gain is thus obtained by copying rather than by editing, and we report $\gamma = 1$ throughout.



(a) Illustration of gate strength vs. edit magnitude

(b) Illustration of gate strength along the reverse process

Figure 6: **Behavior of the learned gate.** (a) Mean gate value g against edit magnitude $d(\text{target}, \text{time-aligned source})$, with interquartile and 10–90% bands across samples in each bin. (b) Mean gate over frames and channels along the reverse trajectory, with the same bands.

Gate Learning. Equation 7 says the optimal gate should behave in two ways: it should be smaller when the edit is larger, and it should go to zero at the end of denoising. Figure 6 shows the learned gate does both. In Figure 6(a), the mean gate drops from 0.223 in the smallest-edit bin to about 0.10 in the largest bins. These results verify that the gate assigns lower reuse weights to sequences that require larger modifications. In Figure 6(b), reuse stays near 0.18 for most of the reverse process and then falls to 0.03 over the last fifty steps. The source helps while the trajectory is still being decided, but the aligned source is not the target, so holding onto it at $t = 0$ would introduce error instead of detail. From Figure 6, without any edit mask, supervision from the editing target alone teaches the gate how much of the source to reuse and when to release the source motion.

5 Conclusion

In this work, we analyze the denoising process of existing methods and identify the temporal-detail attenuation as a systematic failure of diffusion motion editors. We propose Source-Preserving Denoising, which makes source reuse an explicit operation at every reverse step by blending the clean-motion estimate toward the aligned source through a learned per-frame, per-dimension gate. Extensive experiments demonstrate that SuperMotion consistently outperforms state-of-the-art editors on both MotionFix and STANCE Adjustment. Ablations attribute the improvement to the learned preservation gate, which enables selective source reuse and preserves high-frequency temporal detail that competing methods attenuate.

6 AI USE STATEMENT

The manuscript was drafted by the authors. We then used Claude Opus 5.5 to polish the prose and to check typos, asking it to improve sentence-level wording while leaving the technical content unchanged. After that pass, we read the revised text ourselves and corrected every phrasing that no longer matched what we had meant. All claims, experimental choices, and reported numbers are the authors'.

REFERENCES

Nikos Athanasiou, Alpár Cseke, Markos Diomataris, Michael J Black, and Gül Varol. Motionfix: Text-driven 3d human motion editing. In *SIGGRAPH Asia*, 2024.

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, 2022.

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *CVPR*, 2023.

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *CVPR*, 2018.

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *CVPR*, 2023.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *CVPR*, 2022.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *CVPR*, 2024.

Ziyan Guo, Zeyu Hu, De Wen Soh, and Na Zhao. Motionlab: Unified human motion generation and editing via the motion-condition-motion paradigm. In *ICCV*, 2025.

Gyojin Han and Junmo Kim. Cross-axis feature fusion with joint-wise motion difference prediction for text-based 3d human motion editing. *arXiv preprint arXiv:2606.01014*, 2026.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. In *NeurIPS*, 2023.

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 *CVPR*, 2025.

Jihoon Kim, Jiseob Kim, and Sungjoon Choi. Flame: Free-form language-based motion synthesis & editing. In *AAAI*, 2023.

Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *NeurIPS*, 2021.

Zhengyuan Li, Kai Cheng, Anindita Ghosh, Uttaran Bhattacharya, Liangyan Gui, and Aniket Bera. Simmotionedit: Text-based human motion editing with motion similarity prediction. In *CVPR*, 2025.

Shaohui Lin, Zhenwu Shi, Jingyu Gong, Jiao Xie, Yu Zhou, Baochang Zhang, Lizhuang Ma, and Chia-Wen Lin. Spatial temporal synergy: Balancing change and invariance in text driven 3d human motion editing. *arXiv preprint arXiv:2608.16008*, 2026.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *ECCV*, 2022.

Mathis Petrovich, Michael J Black, and Gül Varol. Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. In *ICCV*, 2023.

Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. Mmm: Generative masked motion model. In *CVPR*, 2024.

Ekkasit Pinyoanuntapong, Muhammad Saleem, Korrawe Karunratanakul, Pu Wang, Hongfei Xue, Chen Chen, Chuan Guo, Junli Cao, Jian Ren, and Sergey Tulyakov. Maskcontrol: Spatio-temporal control for masked motion synthesis. In *ICCV*, 2025.

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 *ICML*, 2021.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

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. *arXiv preprint arXiv:2605.30969*, 2026.

Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *ECCV*, 2022a.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022b.

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.

J Zhang, Y Zhang, X Cun, S Huang, Y Zhang, H Zhao, H Lu, and X Shen. T2mgpt: Generating human motion from textual descriptions with discrete representations. In *CVPR*, 2023a.

Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. Finemogen: Fine-grained spatio-temporal motion generation and editing. In *NeurIPS*, 2023b.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *TPAMI*, 46(6): 4115–4128, 2024.

A Evaluation Protocol

MotionFix retrieval. The generation loader writes 836 sequences, consisting of a random half of the official test split (506 examples) and the 330 validation examples. The TMR gallery (Athanasiou et al., 2024) is the evaluator's test set, so full-pool metrics rank each query against the *506-id test pool*, not against all 836 generated files. Validation outputs never enter that gallery. Batch retrieval sorts the same IDs, shuffles them, and forms disjoint batches of 32, leaving 26 examples unused (480 scored queries). A batch percentage is therefore not a statistic over all 506 examples.

Published MotionFix numbers use this pool. Table 1 reports SuperMotion at 33.20 full-pool R@1.

STANCE Adjustment. The published protocol ranks against 196 IDs. OmniME is quoted from its paper. TMED, SimMotionEdit and MotionRefit have no official STANCE checkpoint, so those rows in Table 2 are our runs of their official code.

B Baseline Implementations

MotionFix rows in Table 1 follow the numbers reported by each paper. The notes below record only the representation conversions used for the detail-ratio and copy-line measurements.

TMED and SimMotionEdit. Official checkpoints, evaluated with their released loaders at guidance 2.0.

MDM. A pretrained text-to-motion model conditioned on the edit instruction only. HumanML3D features are converted to joints, fitted to SMPL, and aligned into the MotionFix world frame.

MDM-BP. The same MDM backbone, with GPT body-part labels. Unmentioned parts are copied from the source in feature space at every reverse step, so the median detail ratio of $1.01\times$ is obtained by construction.

MotionRefit. Evaluated zero-shot from their STANCE-trained weights. We convert the source to their SMPL-X 20 fps format, fix the task to regeneration with the raw instruction, and align frames and the floor both ways.

MotionLab. Treated as one task of their unified generative model. Table 1 quotes their Source-TextMetrics, which do not report a full-pool protocol.

OmniME. Quoted from the paper on both benchmarks. The detail-ratio and copy-line entries in Table 4 use their official MotionFix checkpoint.

C Additional Metrics

Representation. The denoiser operates on the 207-dimensional MotionFix feature of Section 3.1. The preservation gate and the clean-space source anchor are applied in this space. Exported TMR features drop the 22 local joint coordinates and convert increments to absolute SMPL parameters, yielding 135 dimensions (translation 3, global orientation 6D, body pose 126). Distances in Table 4 use the 135-d export.

Detail ratio. Section 1 defines the detail ratio on the running clean-motion estimate during denoising. Dataset-level numbers in Table 4 use the same second-order difference magnitude on the exported 135-d pose, aggregated as the median over $N{=}836$ generated files. A value of $1\times$ matches the target's second-difference magnitude but does not imply a correct motion. MotionLab's median is $1.40\times$. The $1.01\times$ of MDM-BP is produced by copying source features.

Copy source. Let $d(\cdot,\cdot)$ be the mean per-frame L_2 distance on the 135-d pose. Time-aligning the source onto the target timeline and scoring that copy gives $d(s, x_0) = 1.745$. Any editor whose $d(\hat{x}_0, x_0)$ exceeds 1.745 ends farther from the target than the copy-source baseline, as TMED does at 1.885. SuperMotion sits at 1.586, with 54.8% of outputs closer to the source than to the target, which is less source-biased than OmniME (60.8%) and TMED (60.9%). This is the same control as Figure 5. Raising the gate scale improves retrieval by copying, not by editing.

Table 4: Copy-versus-edit diagnostic on MotionFix (N=836).

Method	Detail ratio	$d(\text{gen}, \text{tgt})$	$d(\text{gen}, \text{src})$	Closer to src
TMED	0.330	1.885	1.791	60.9%
SimMotionEdit	0.708	1.661	1.616	56.1%
OmniME	0.658	1.642	1.496	60.8%
SuperMotion	1.035	1.586	1.563	54.8%
Copy source	–	1.745	0	–

D Sampling Procedure

Algorithm 1. Source-preserving reverse step.

Input. Source s, instruction c, length T, trained denoiser and preservation head, noise schedule.

1. Align $\bar{s} \leftarrow \text{Align}(s, T)$ by linear resampling. Draw $x_T \sim \mathcal{N}(0, I)$.
2. For $t = T, \ldots, 1$, repeat the following steps.
   (a) Evaluate the unconditional, source-only, and source-plus-text branches at the same x_t.
   (b) Form the guided velocity $\hat{v}_{\text{cfg}}$ by Eq. (3).
   (c) Convert $\hat{x}_0$ by Eq. (2).
   (d) Read $g = \sigma(f_\phi(h))$ from the fully conditioned branch.
   (e) Set $\tilde{x}_0 = \hat{x}_0 + g \odot (\bar{s} - \hat{x}_0)$ by Eq. (6).
   (f) Sample the DDPM posterior $q(x_{t-1} \mid x_t, \tilde{x}_0)$. Do not convert the correction back to velocity.
3. Return the final clean motion.

Guidance is applied to the velocity branches before the blend. The gate is not itself guided, because extrapolating g would break the convex-combination interpretation of Eq. (6). All three branches are evaluated at the same x_t, since a replica mismatch under velocity prediction produces an inconsistent reverse step.

E Preservation-Head Architecture

Section 4.1 gives the training setup. The preservation head is a three-layer MLP on the same target-token features $h \in \mathbb{R}^{T \times 512}$, with widths $512 \to 1024 \to 1024 \to 207$ and SiLU activations. The last layer is zero-initialized with bias -6, so $g = \sigma(-6) \approx 0.00247$ and the sampler matches the parent model at initialization. The velocity loss is applied to the ungated prediction, while $\mathcal{L}_{\text{anc}}$ and $\mathcal{L}_{\text{hf}}$ are applied to $\tilde{x}_0$.

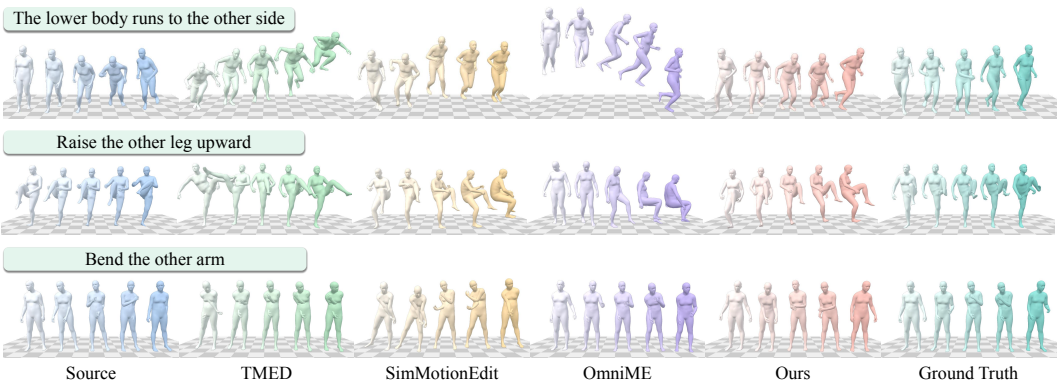


Figure 7: Qualitative comparison on STANCE Adjustment. Each instruction targets a specific limb or side. Left to right: source, TMED, SimMotionEdit, OmniME, SuperMotion, and the reference target.

F QUALITATIVE RESULTS ON STANCE ADJUSTMENT

Figure 7 shows results on STANCE Adjustment, where each instruction targets a specific limb or side. In *the lower body runs to the other side*, the baselines introduce a full-body reorientation or an airborne pose absent from the source, while our result redirects the lower body alone. In *raise the other leg upward* and *bend the other arm*, the baselines frequently act on the wrong limb or apply the change to both sides. Our method selects the intended one and leaves the remaining joints close to the source.