

# CleanMDM: Clean Motion Diffusion Model for Multimodal Motion Cleanup

Zhe Li<sup>1,\*</sup> Shicheng Wang<sup>2,\*</sup> Bowen Cai<sup>3</sup> Huan Fu<sup>3,†</sup>

<sup>1</sup>Peking University <sup>2</sup>Light Origins <sup>3</sup>Alibaba

\*Equal contribution. †Corresponding author.

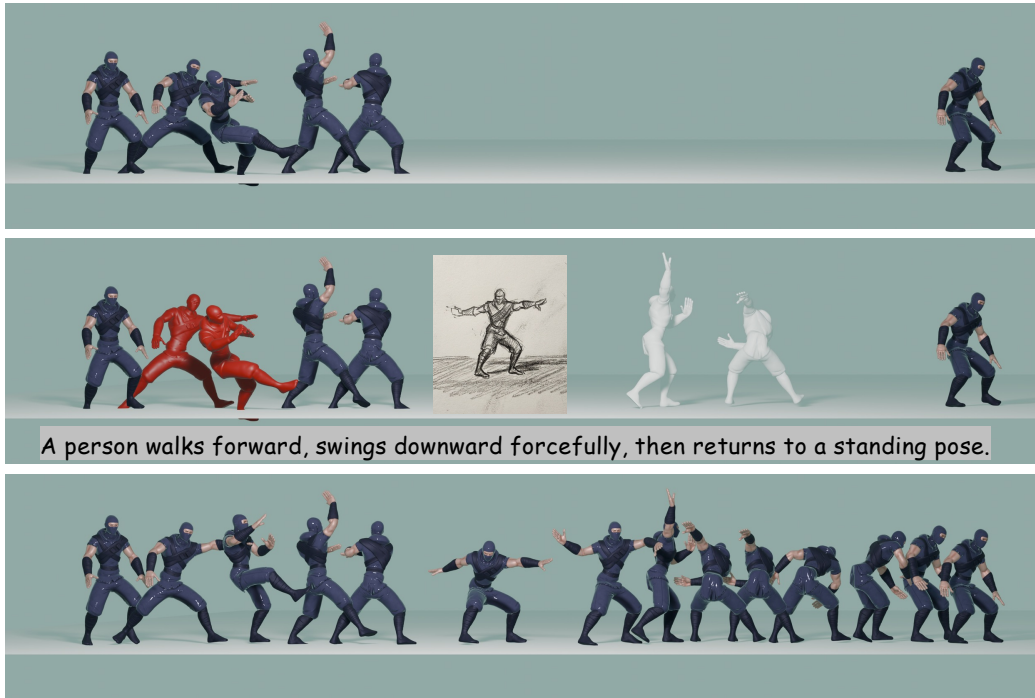


Figure 1: **Overview of motion cleanup.** **Top:** the input motion with a localized corrupted interval, represented as masked noisy motion. **Middle:** the same clip augmented with optional, plug-and-play conditions: corrupted frames are highlighted in red for visualization, a reference image provides sparse 2D keyframes, white silhouettes indicate sparse 3D keyframes, and a text prompt specifies high-level intent. **Bottom:** CleanMDM generates a clean and temporally consistent motion sequence.

## Abstract

Motion capture data is rarely directly usable, as they typically exhibit missing segments, jitter, drift and contact artifacts. Traditionally, corrupted motions are cleaned by animators through the manual identification of keyframes from noisy motion, subsequent keyframe correction, and interpolation between corrected keyframes to reconstruct coherent motion. While the rise of generative motion models has made automatic cleanup feasible, most approaches operate as black box denoisers with limited controllability, making it difficult to preserve reliable segments or enforce specific user intents. Inspired by animation workflows, we present **CleanMDM**, a unified multimodal motion cleanup framework that formulates cleanup as masked conditional generation with plug-and-play conditions. This single model supports arbitrary combinations of noisy 3D motion, sparse 2D keyframes, sparse 3D keyframes, and text. This design enables both automatic cleanup without additional

user annotation and controllable cleanup under multimodal guidance. To further improve motion realism, we incorporate the Latent Motion Quality Discriminator (**LMQD**) to better match kinematic distributions and reduce skating, jitter, and interpenetration artifacts, and we apply **Mesh-Aware Contact Projection** as a test-time optimization step to enhance contact and physical consistency. Experiments across multiple datasets demonstrate that **CleanMDM** consistently outperforms prior cleanup and generation baselines, and that low cost conditions (text and 2D keyframes) provide reliable controllability gains in multimodal cleanup scenarios.

## 1 Introduction

Motion data in modern animation pipelines is increasingly acquired from video-based motion capture Shen et al. [2024], Goel et al. [2023] and generative motion models Tevet et al. [2023], Chen et al. [2023], which are more accessible and scalable than traditional studio capture but frequently introduce missing observations, temporal jitter, drift, and contact artifacts such as foot sliding, floating, and ground penetration. Because these artifacts are visually amplified after skinning and rendering, animators still spend substantial time on manual cleanup and inbetweening, making motion repair a persistent production bottleneck Williams [2012], Holden [2018], Kovar et al. [2002].

A key characteristic of real-world motion corruption, largely overlooked by existing methods, is that degradation is almost always localized: only certain temporal intervals or body parts are affected, while the remaining motion stays reliable. For example, video-based estimates may exhibit lower-body drift while the upper body is essentially correct, or jitter may appear in only a handful of joints over a short window. This partial reliability has a direct consequence for method design: an ideal cleanup system should identify and preserve the trustworthy portions of a motion while selectively regenerating only the corrupted regions, and it must accept diverse forms of guidance depending on what information is available for a given shot.

Existing approaches largely fall into two categories, neither of which is specifically designed to handle partial reliability. **Restoration-style denoisers** Mu et al. [2025], Zhang et al. [2024b] map an entire noisy sequence to a clean motion in a black-box fashion. They improve average reconstruction error but treat every frame uniformly, providing limited controllability when animators need to lock key poses, preserve trustworthy segments, or repair only a specified interval. **Inbetweening and conditional generation** Cohan et al. [2024], Zhang et al. [2025a], Meng et al. [2025] take the opposite stance: they regard the corrupted region as entirely unobserved and reconstruct it from sparse constraints (keyframes, text). When applied to cleanup, this discards the partially reliable observations inside the corrupted interval, leading to over-reconstruction, style drift, and unstable contact behavior. Moreover, these two families of methods are typically developed and deployed as separate, modality-specific models, which fragments supervision, limits cross-modal knowledge transfer, and requires maintaining multiple pipelines. It is worth noting that motion cleanup is fundamentally different from motion generation: generation creates motion from scratch given sparse or abstract signals, whereas cleanup starts from an existing clip that is partially reliable and must preserve its trustworthy content. Recent multimodal generation models such as Li et al. [2025] accept heterogeneous inputs (2D/3D keyframes, text) but are designed for the generation task. They do not model per-frame reliability or enforce non-destructive preservation of the input draft, which limits their applicability to production cleanup scenarios requiring fidelity to the original performance.

We present **CleanMDM**, a unified multimodal motion cleanup framework that directly addresses the partial-reliability challenge. Our key insight is to formulate cleanup as *masked conditional generation* in a shared latent space. The central mechanism is *confidence-aligned multimodal fusion*. Heterogeneous conditions, including noisy 3D motion drafts with per-frame confidence masks, sparse 2D keyframes, sparse 3D keyframes, and text prompts, are encoded into a common latent representation and fused via frame-wise confidence weighting (Eq. 2). Unlike raw-space additive fusion used in generation models Li et al. [2025], our confidence-aligned design explicitly encodes the *reliability* of each frame: high-confidence frames are preserved, low-confidence frames are regenerated, and the boundary between the two is determined by the per-frame weights rather than a hard binary mask. This enables a smooth continuum from fully missing intervals (inbetweening) to partially degraded segments (denoising-style cleanup), all within a *single* model. Text and sparse 2D keyframes serve as low-cost, plug-and-play conditions that provide semantic and geometric guidance without requiring expensive 3D annotation. By jointly training on all modalities, CleanMDM

encourages cross-modal transfer: the richer supervision yields a stronger motion prior that benefits even the simplest auto-cleanup setting where no additional guidance is provided.

Two additional technical contributions improve motion quality beyond standard regression objectives. First, we introduce the **Latent Motion Quality Discriminator (LMQD)**, a lightweight adversarial regularizer that operates on entire sequences in the motion autoencoder’s latent space. Unlike prior kinematic-feature discriminators used in physics-based control Peng et al. [2021], Luo et al. [2023], LMQD back-propagates gradients directly through the latent representation, better capturing distributional characteristics of natural motion and more effectively penalizing perceptually salient artifacts (skating, jitter, penetration) that have small  $\ell_2$  magnitude but high visual salience. Second, we propose **Mesh-Aware Contact Projection**, a test-time optimization step that refines the full sequence with explicit constraints on SMPL mesh foot-sole vertices. Unlike conventional joint-level inverse kinematics Holden [2018], Yao et al. [2024], our formulation reasons about body-surface geometry relative to the ground plane, enforcing hard contact consistency that joint-only approaches structurally cannot achieve.

Our contributions are summarized as follows:

- **Confidence-aligned multimodal cleanup framework.** We formulate motion cleanup as masked conditional generation in a shared latent space with per-frame confidence weighting, enabling non-destructive repair that preserves reliable motion while regenerating corrupted regions — a capability that neither black-box denoisers nor generation-oriented models provide. A single CleanMDM model covers auto cleanup, inbetweening, and multimodal-guided repair.
- **Latent adversarial regularization.** We introduce LMQD, a latent-space sequence discriminator that provides distributional supervision beyond regression losses, yielding measurable improvements on contact and temporal-quality metrics.
- **Mesh-aware contact projection.** We propose a test-time optimization that constrains SMPL mesh vertices to enforce surface-level ground contact (foot sliding, penetration, and floating).

## 2 Related Works

### 2.1 Human Motion Generation

Recent progress in human motion synthesis is largely driven by generative models. Diffusion approaches (e.g. Tevet et al. [2023], Chen et al. [2023], Zhang et al. [2024a], Zhao et al. [2024], Zhang et al. [2024c, 2023b]) show their ability to model long-range spatio-temporal motion relationships, while VQ/tokenized pipelines Zhang et al. [2023a], Guo et al. [2024], Li et al. [2024], Jiang et al. [2023], Pinyoanuntapong et al. [2024], Zhang et al. [2024d] introduce discrete motion token modeling. Representations evolve from kinematics-rich descriptors (e.g., HumanML3D Guo et al. [2022]) toward more concise forms, including redundancy reduced designs Meng et al. [2024] and absolute coordinate modeling Meng et al. [2025]. Control has expanded beyond text to geometric cues and interactive editing interfaces Wang et al. [2025a], Zhong et al. [2025], Karunratanakul et al. [2023], Xie et al. [2023], and more recently to multimodal guidance Li et al. [2025], Luo et al. [2024], Xu et al. [2025], Zhang et al. [2025b], Chen et al. [2025]; some works also pair generation with physics for executability Yuan et al. [2023], Tevet et al. [2024], Tessler et al. [2024], Fan et al. [2025], Wang et al. [2025b]. Despite these advances, semantic conditions often underconstrain fine geometry and contacts, while physics constraints can disrupt motion timing and cause local limb deformations.

### 2.2 Motion Cleanup

Repairing motion captured from videos or produced as drafts is critical for production. Existing work largely follows two directions. Cleanup-style methods map corrupted sequences to clean motion via end-to-end denoising Mu et al. [2025], Zhang et al. [2024b], reducing reconstruction error but offering limited fine-grained control (e.g., preserving reliable segments or enforcing locked poses). Some approaches detect corrupted segments within a motion sequence and subsequently regenerate them. However, these approaches treat missing regions as uninformative and do not explicitly combine partially reliable observations with heterogeneous constraints. In this sense, inbetweening can be viewed as a repair mechanism, as inbetweening and conditional completion synthesize motion from sparse constraints such as keyframes or text Cohan et al. [2024], Zhang et al. [2025a]. In real world cleanup, corruption is often localized, and ignoring valid observations can increase ambiguity, causing over-reconstruction, style drift, and unstable contact-related artifacts, which are central to industrial

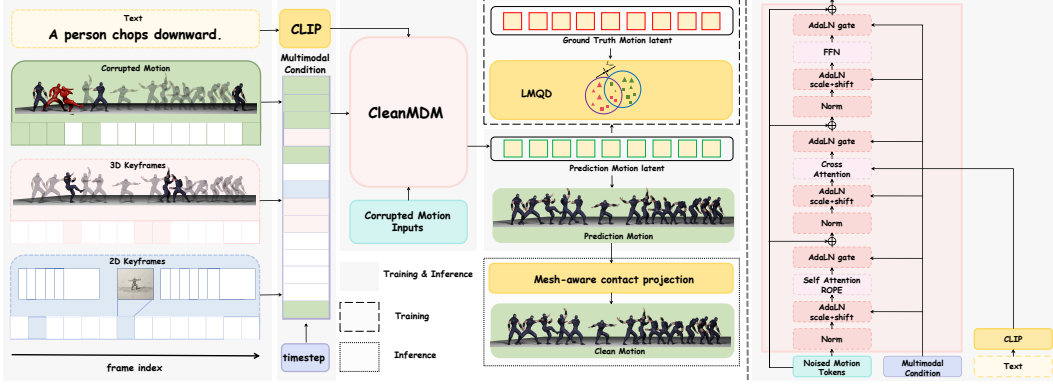


Figure 2: **CleanMDM framework and architecture.** **Left:** CleanMDM formulates motion cleanup as masked conditional generation in the autoencoder latent space, using a noisy 3D draft with frame-wise confidence and optional plug-and-play cues (sparse 2D/3D keyframes and text). Solid-bordered inputs are required; dashed-bordered inputs are optional. **Visualization:** semi-transparent skeletons indicate masked-out (unobserved) regions, and red skeletons indicate corrupted frames in the noisy draft. LMQD is used only during training for latent adversarial regularization, and an optional Mesh-Aware Contact Projection post-process is applied at inference to reduce contact artifacts. **Right:** CleanMDM block with frame weighted geometric fusion injected via AdaLN+gated residuals into self-attention (RoPE) and FFN; text is injected through cross-attention.

requirements. Recently, GenMo Li et al. [2025] unifies generation and estimation with multimodal inputs, but as a generation model it does not model per-frame reliability.

### 2.3 Motion Quality and Physical Plausibility

To mitigate motion artifacts such as sliding, floating, and penetration, prior work incorporates contact losses Tevet et al. [2023, 2024], physics-driven refinement Luo et al. [2023], Yao et al. [2024], IK post-processing Shen et al. [2024], and adversarial motion priors (e.g., AMP Peng et al. [2021]). These approaches address complementary aspects of motion quality; achieving both controllable repair and physical consistency typically requires combining learned priors with explicit contact enforcement, ideally avoiding heavy simulation during training.

## 3 Data Representation

We represent a motion clip as absolute joint positions  $\mathbf{X}_{1:T} \in \mathbb{R}^{T \times J \times 3}$  with  $J=22$ . Following ACMMDM Meng et al. [2025], we adopt absolute coordinates as the interface for cleanup because they align naturally with the spatial signals used in our task. Sparse 3D keyframes can be inserted in the same coordinate system, 2D keypoints are obtained by direct camera projection, and contact reasoning or inference-time IK can be formulated directly in metric space. Compared with relative or velocity-based representations, absolute coordinates also avoid reconstructing global trajectories by accumulating local offsets, where small frame-wise errors can propagate into long-horizon drift. This makes the representation especially suitable for non-destructive cleanup: reliable observations, sparse controls, and regenerated regions all remain comparable in a single world-space coordinate system.

All conditions are aligned to the latent temporal resolution produced by the motion autoencoder Kingma and Welling [2013]. With a downsampling factor of  $s=4$ , the latent length is  $T' = \lceil T/s \rceil$ .

**Noisy 3D observations.** We denote the corrupted joint sequence by  $\tilde{\mathbf{X}}_{1:T}$ . A frame-level corruption indicator or user-provided interval annotation is converted into a confidence weight and applied to  $\tilde{\mathbf{X}}_{1:T}$  before encoding, so unreliable frames contribute less to the noisy 3D condition. After encoding, we obtain noisy 3D condition tokens and their latent-level weights  $w_{\text{noisy}} \in [0, 1]^{T'}$ .

**Sparse clean 3D keyframes.** From frames marked as unreliable, we randomly sample a portion as clean 3D keyframes during training. These frames provide sparse ground-truth constraints inside degraded intervals. We store a binary keyframe mask and its latent-level weights  $w_{\text{clean}} \in [0, 1]^{T'}$ .

**2D keypoints.** We use 2D keypoints as a dense geometric condition rather than a supervised loss. For each clip, a camera path is sampled from  $\{\text{fixed}, \text{dolly in}, \text{dolly out}, \text{follow}, \text{orbit}\}$  to obtain

extrinsics  $(\mathbf{R}_t, \mathbf{p}_t)$ ; fixed intrinsics  $K$  are used for perspective projection. The 2D sequence is fed as the conditioning signal  $\mathbf{U}_{1:T}$ . We apply a standard crop and scale normalization to absolute image scale. During training, we mimic sparse detections by sampling a small portion of unreliable frames as 2D keyframes. Their latent-level weights are denoted by  $w_{2D} \in [0, 1]^{T'}$ .

$$\mathbf{u}_{t,j} = \Pi(K, \mathbf{R}_t, \mathbf{p}_t; \mathbf{X}_{t,j}), t \in [1:T], j \in [1:J]. \quad (1)$$

**Text.** Each clip is paired with a textual description. We tokenize the caption and extract token-level features using a frozen CLIP text encoder Radford et al. [2021]. These features are used as an additional conditioning context via cross-attention.

## 4 Method

CleanMDM formulates motion cleanup as masked conditional generation in a shared latent space. The central design principle is confidence-aligned multimodal fusion: heterogeneous conditions are encoded into a common latent representation and fused via per-frame confidence weighting, so that trustworthy frames are preserved while corrupted regions are selectively regenerated. This fundamentally distinguishes CleanMDM from generation models (e.g., GenMo Li et al. [2025]) that synthesize motion from scratch without modeling input reliability.

### 4.1 Latent Motion Modeling Backbone

CleanMDM performs motion cleanup in a learned latent space. Given an absolute joint trajectory  $\mathbf{X}_{1:T} \in \mathbb{R}^{T \times J \times 3}$ , we encode it with a motion autoencoder  $\mathcal{E}$  and decode with  $\mathcal{D}$ . The encoder downsamples only in time by a factor  $s=4$ , producing a latent sequence of length  $T' = \lceil T/s \rceil$  while keeping the joint dimension  $J$  unchanged. We train a conditional flow matching Transformer Lipman et al. [2022], Peebles and Xie [2023], Ho et al. [2020] to predict a velocity field in latent space, and we sample with an ODE solver Chen et al. [2018]. Working in latent space reduces computation, improves training stability, and allows us to integrate heterogeneous conditions at a common temporal resolution. Our backbone is a conditional Transformer over the noised latent motion tokens. We use RoPE positional encoding Su et al. [2024] and QK Normalization Henry et al. [2020] in self attention, and inject conditions by AdaLN modulation with residual gating, which allows the model to smoothly activate or deactivate guidance under missing or partial conditions. Text features from a frozen CLIP encoder are provided as context to a gated cross attention layer.

### 4.2 Multimodal Conditions and Fusion

CleanMDM treats motion cleanup as masked conditional generation with plug-and-play conditions. The core mechanism is *confidence-aligned fusion in latent space*: instead of directly concatenating raw inputs or using additive fusion in joint space, we encode all geometric conditions through the shared autoencoder and fuse them via per-frame confidence weighting. This design is motivated by two reasons. First, the autoencoder provides a shared coordinate system and scale for 3D motion signals, which makes fusion well-posed and improves robustness to partial corruptions. Second, latent fusion follows the successful recipe of latent diffusion in vision, where rich conditions are projected to the model width and injected into a generative backbone without forcing all modalities into a brittle raw-space representation. Unlike raw-space additive fusion used in generation models Li et al. [2025], our confidence-aligned design explicitly encodes the reliability of each frame, enabling non-destructive cleanup.

**Condition construction** For 3D conditions, we *mask in input space first*, then encode with the same  $\mathcal{E}$ : (i) noisy 3D observations use a frame-wise confidence derived from an interval annotation or automatic corruption indicator so unreliable intervals contribute less; (ii) sparse clean 3D keyframes use a binary mask so only selected frames are kept. The resulting latents are flattened into per-time tokens, and their masks are downsampled to the latent rate to obtain frame weights. For 2D keyframes, we mask in 2D space, downsample to  $T'$ , flatten each frame, and map it with a small MLP, without explicit 2D-to-3D lifting. Since the backbone learns a 3D motion latent from paired projected 2D cues, the MLP only needs to preserve image-plane geometry while the motion prior infers compatible 3D motion. This avoids the feature-space mismatch introduced by heavier pretrained 2D representations; Appendix A.7 (Table 9) confirms that it outperforms MotionBERT features.

**Weighted fusion** Cleanup differs from generation because the input draft is partially reliable. StableMotion Mu et al. [2025] targets corrupted motion but lacks a unified multimodal interface,

while GenMo Li et al. [2025] accepts heterogeneous inputs but does not encode which draft frames are trustworthy. We therefore expose reliability as frame weights, preserving high-confidence draft frames while letting optional keyframes dominate unreliable intervals.

Let  $\mathbf{y}_t^{\text{noisy}}$ ,  $\mathbf{y}_t^{\text{clean}}$ , and  $\mathbf{y}_t^{2D}$  be the condition tokens at latent frame  $t$ , and let  $w_{\text{noisy},t}$ ,  $w_{\text{clean},t}$ , and  $w_{2D,t}$  be their corresponding downsampled frame weights. Here  $\phi_{\text{noisy}}$  and  $\phi_{\text{clean}}$  are linear projections for 3D latent tokens, and  $\phi_{2D}$  is a lightweight MLP for 2D tokens. A modality is disabled by setting its weight to zero; the same mechanism also supports condition dropout and CFG Ho and Salimans [2022]. We fuse them by a weighted sum:

$$\mathbf{c}_t = w_{\text{noisy},t} \phi_{\text{noisy}}(\mathbf{y}_t^{\text{noisy}}) + w_{\text{clean},t} \phi_{\text{clean}}(\mathbf{y}_t^{\text{clean}}) + w_{2D,t} \phi_{2D}(\mathbf{y}_t^{2D}), \quad t \in [1:T']. \quad (2)$$

**Condition injection** We inject time aligned conditions through AdaLN modulation and residual gates, using the token  $\mathbf{c}_t$  as the conditioning signal for each block. Text is injected via cross attention: we extract token level features from a frozen CLIP text encoder, project them with MLP, and gate the cross-attention contribution to remain compatible with CFG Ho and Salimans [2022].

### 4.3 Latent Motion Quality Discriminator (LMQD)

Regression losses in flow matching do not fully capture perceptual motion quality. We draw inspiration from PHC Luo et al. [2023] in using adversarial supervision to improve motion realism, but instead apply the discriminator in the autoencoder latent space rather than on handcrafted kinematic features (joint positions, accelerations, power). Operating in latent space offers two advantages: (i) gradients flow directly to the generator without passing through the decoder, and (ii) the learned latent captures more intrinsic motion structure than kinematic descriptors. An ablation over five discriminator feature representations (Table 11) confirms that latent-space discrimination achieves the best overall quality. LMQD enforces a sequence-level distribution constraint and reduces skating, jitter, and penetration.

**Real and fake latents.** The discriminator  $D$  operates on whole latent sequences. Real samples are normalized GT latents  $\mathbf{z}_0^{\text{norm}}$ . Fake samples are built from the flow matching path. We sample  $(t, x_0, x_1)$  from the transport, obtain  $x_t$ , predict velocity  $v_\theta(x_t, t, \mathcal{C})$ , and flatten the sequence before feeding it to  $D$ . We form a denoised latent prediction as

$$\hat{x}_0 = x_0 + v_\theta(x_t, t, \mathcal{C}). \quad (3)$$

**Joint training with alternating updates.** We train CleanMDM and LMQD jointly with alternating optimization, following the standard practice used in CycleGAN Zhu et al. [2017], Isola et al. [2017] training codebases. At each iteration, we first freeze  $D$  and update the generator with a flow-matching objective plus an adversarial term, where  $\mathcal{L}_{\text{adv}}$  encourages  $D(\hat{x}_0)$  to be classified as real; we then freeze the generator and update  $D$  to separate  $\hat{x}_0$  from  $\mathbf{z}_0^{\text{norm}}$ .

$$\mathcal{L}_G = \mathcal{L}_{\text{flow}} + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}}, \quad (4)$$

### 4.4 Auto Cleanup and Inbetweening in One Model

CleanMDM unifies mask-guided inbetweening and blind auto cleanup because animation repair often alternates between the two: an animator may correct corrupted poses, lock keyframes, and then inbetween the damaged span. Using separate cleanup and inbetweening models would form a cascade in which the first stage can pass drift or over-smoothed poses to the second stage, while reliable regions may be rewritten twice. We instead train one model with a shared condition interface. For half of each training batch, we remove explicit interval supervision by marking all frames as clean in the corruption indicator, forcing the model to perform blind auto cleanup from the noisy draft. For the other half, we keep the interval annotation and train mask-guided cleanup. At inference, users may provide no interval annotation for automatic cleanup, provide corrupted intervals for inbetween-style repair, or additionally supply sparse 2D/3D keyframes and text prompts. All cases use the same condition dictionary, and each modality is enabled or disabled by its confidence weight.

### 4.5 Mesh-Aware Contact Projection

Motion sequences produced by animation pipelines or generative models often contain artifacts such as foot sliding, floating, and ground penetration. Manual cleanup is typically performed frame by frame. Classic CCD-based IK operates frame-locally without temporal context, while physics-driven correction (e.g., PHC Luo et al. [2023]) may introduce motion distortion when strictly tracking input keypoints. We therefore propose Mesh-Aware Contact Projection, a test-time optimization that refines

Table 1: **Comparison to prior work under three evaluation settings.** We report results for auto cleanup (auto cleanup from a noisy draft, no interval annotation), inbetween (mask guided interval completion), and multimodal cleanup (interval completion with additional 2D/3D keyframes and text). MPJPE(mm) reports global error, with local MPJPE in parentheses (root effects removed). Skate denotes foot skating ratio; Acc and Jitter measure second- and third-order temporal errors; Pen.F(%) and Pen.D(mm) denote ground penetration frequency and distance.

| Setting            | Method          | MPJPE(mm)↓          | Skate↓      | Acc Err↓    | Pen.F(%)↓   | Pen.D(mm)↓  | Jitter↓     | KF Err(mm)↓ | FID↓        | Div→        | M2M Score↑  | M2M R@3↑      |
|--------------------|-----------------|---------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|---------------|
| Auto cleanup       | GT              | 0.00                | 0.13        | 0.00        | 0.00        | 0.00        | 3.73        | 0.00        | 0.00        | 9.40        | 1.00        | 100%          |
|                    | StableMotion    | 59.36               | 0.20        | 2.24        | 0.00        | 1.25        | 7.90        | ✗           | 0.12        | 7.09        | 0.95        | 90.72%        |
|                    | CleanMDM        | <b>18.25(8.84)</b>  | 0.18        | 2.04        | 0.03        | 0.29        | 6.28        | ✗           | <b>0.12</b> | <b>9.17</b> | <b>0.99</b> | <b>99.92%</b> |
|                    | CleanMDM + MACP | 45.15(19.91)        | <b>0.14</b> | <b>1.88</b> | <b>0.00</b> | <b>0.00</b> | <b>3.74</b> | ✗           | 0.24        | 9.16        | 0.96        | 98.73%        |
| Inbetween          | GT              | 0.00                | 0.12        | 0.00        | 0.00        | 0.00        | 3.15        | 0.00        | 0.00        | 9.44        | 1.00        | 100%          |
|                    | ACMDM           | 54.4                | 0.21        | 4.13        | 0.00        | 0.54        | 10.71       | 21.9        | 0.39        | 9.65        | 0.95        | 95.11%        |
|                    | CondMDI         | 53.5                | 0.42        | 7.59        | 0.00        | 0.01        | 33.77       | 31.8        | 0.14        | 10.27       | 0.98        | 91.53%        |
|                    | CleanMDM        | <b>42.84(29.28)</b> | 0.18        | <b>2.21</b> | 0.15        | 1.04        | 6.23        | <b>14.8</b> | <b>0.11</b> | <b>9.04</b> | <b>0.97</b> | <b>96.47%</b> |
|                    | CleanMDM + MACP | 71.86(37.14)        | <b>0.12</b> | 2.27        | <b>0.00</b> | <b>0.00</b> | <b>4.24</b> | 51.44       | 0.19        | 9.02        | 0.93        | 92.30%        |
| Multimodal cleanup | GT              | 0.00                | 0.13        | 0.00        | 0.00        | 0.00        | 3.73        | 0.00        | 0.00        | 9.40        | 1.00        | 100%          |
|                    | GenMo           | 646.4               | 0.02        | 2.05        | 7.71        | 2.85        | <b>2.15</b> | 633.2       | 11.94       | 7.50        | 0.64        | 70%           |
|                    | CleanMDM        | <b>21.95(15.73)</b> | 0.18        | <b>1.98</b> | 0.06        | 0.76        | 5.54        | <b>11.4</b> | <b>0.11</b> | <b>9.23</b> | <b>0.99</b> | <b>99.82%</b> |
|                    | CleanMDM + MACP | 50.91(24.77)        | <b>0.14</b> | 2.00        | <b>0.00</b> | <b>0.00</b> | 3.75        | 45.46       | 0.26        | <b>9.24</b> | 0.96        | 98.62%        |

the entire sequence under explicit ground contact constraints. Let the environment be represented by a height field  $g_\psi(x, z)$  that returns the ground height at horizontal location  $(x, z)$ . This makes the formulation terrain-agnostic: a horizontal plane is the special case  $g_\psi(x, z) = h_g$ , where the scalar height  $h_g$  can be optimized jointly with the SMPL parameters.

Since standard SMPL foot keypoints may lie inside the mesh, we sample 54 vertices on each foot sole to better approximate the contact surface. Contact states for heel/toe are provided by a pre-trained estimator Mourrot et al. [2022] or upstream predictions. We solve a sequence-level optimization with a composite objective, the full definitions of which are provided in the Appendix A.3, A.4.

$$\mathcal{L} = \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{contact}}\mathcal{L}_{\text{contact}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} \quad (5)$$

## 5 Experiment

### 5.1 Experimental Setup

We build training clips from AMASS Mahmood et al. [2019] and MotionLLaMA Ling et al. [2024], converting each sequence to a unified SMPL Loper et al. [2023] 22-joint skeleton and absolute joint-position stream. Each clip is normalized by centering the first-frame root on the horizontal plane and standardizing joint coordinates. To emulate video-mocap degradations, we inject localized artifacts into random temporal intervals of clean clips, following the protocol in StableMotion Mu et al. [2025] for part of the noise design. The artifact families include joint jittering, foot sliding, horizontal drifting, and vertical floating. We record a per-frame defect label  $\ell \in \mathbb{N}^T$ , where  $\ell[t] = 0$  denotes a clean frame and non-zero values indicate corrupted frames. Full data filtering, corruption construction, and localization details are provided in Appendix A.2.

We use flow matching Lipman et al. [2022] with linear transport paths and velocity prediction. Sampling uses an Euler ODE solver with 50 steps. The generator learning rate is  $4 \times 10^{-4}$  and the discriminator learning rate is  $5 \times 10^{-5}$ , both with AdamW Loshchilov et al. [2017]. We set  $\lambda_{\text{adv}} = 0.01$ , condition dropout probability 0.1, and EMA decay 0.9999. The generator and discriminator are trained with alternating updates. Our model is trained on 2 H20 GPUs with a batch size of 128 for 600 epochs ( $\sim 16$  hours) and tested on 1 NVIDIA V100 GPU. Metric definitions are provided in Appendix A.5.

### 5.2 Ablation Settings

We denote CleanMDM variants by C0–C6 for compact reporting. All variants share the same latent flow matching backbone and the same input noisy draft; they differ only in enabled conditions and post-processing. Table 3 summarizes the configuration of each variant.

### 5.3 Comparison

**Evaluation settings.** Table 1 compares CleanMDM with representative restoration and conditional generation baselines under three settings. **(i) Auto cleanup** evaluates auto cleanup from a single noisy 3D draft without any interval annotation. To keep the comparison fair, multimodal baselines

Table 2: **Main ablation on multimodal conditions and regularizers.** C0–C6 differ only in the enabled conditions and post-processing (see Table 3).

| Method                      | MPJPE↓       | Skate↓      | Acc↓        | Pen.F↓      | Pen.D↓      | Jitter↓     | KF Err↓     | FID↓        | Div→        |
|-----------------------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| GT                          | 0.00         | 0.13        | 0.00        | 0.00        | 0.00        | 3.73        | 0.00        | 0.00        | 9.40        |
| C0                          | 35.45        | 0.19        | 2.12        | 0.07        | 0.95        | 5.73        | 12.7        | 0.09        | 8.04        |
| C1                          | 34.33        | 0.21        | 2.27        | 0.08        | 0.84        | 6.26        | 13.2        | <b>0.07</b> | 7.95        |
| C2                          | 27.45        | 0.20        | 2.17        | 0.07        | 0.89        | 6.12        | 13.3        | <b>0.07</b> | 8.05        |
| C3                          | 27.32        | 0.20        | 2.19        | 0.06        | 0.78        | 6.17        | 13.0        | 0.08        | 8.06        |
| C4                          | 28.87        | 0.19        | 2.07        | 0.05        | 0.72        | 5.74        | 12.2        | 0.13        | 9.17        |
| <b>C5 (CleanMDM)</b>        | <b>21.95</b> | 0.18        | <b>1.98</b> | 0.06        | 0.76        | 5.54        | <b>11.4</b> | 0.11        | 9.23        |
| <b>C6 (CleanMDM + MACP)</b> | 50.91        | <b>0.14</b> | 2.00        | <b>0.00</b> | <b>0.00</b> | <b>3.75</b> | 45.46       | 0.26        | <b>9.24</b> |

are restricted to the same noisy input, and we provide a default, non-informative prompt (“a person is doing something”). **(ii) Inbetween** evaluates mask guided cleanup where the corrupted interval is removed and the method must inbetween the missing segment. We follow the standard split used by CondMDI, corrupt motions in its test set, and apply the same interval masking for all methods. Text-based methods receive the caption associated with each sequence. **(iii) Multimodal cleanup** evaluates the full production oriented setup, where the corrupted interval is masked and additional user cues are available (sparse 3D keyframes, sparse 2D keyframes, and text). Since no existing cleanup method supports comparable multimodal inputs, we compare against GenMo Li et al. [2025] — a multimodal generation model that accepts 2D/3D keyframes and text — under the same masked-interval completion setting. We run GenMo with the same inputs; when required, we also provide its camera-related inputs following its inference protocol. Note that GenMo was designed for motion generation rather than cleanup; this comparison therefore highlights the gap between generation-oriented and cleanup-oriented approaches when applied to the same repair task.

**Main results.** Across all three settings, CleanMDM achieves the strongest overall motion quality. In **auto cleanup**, CleanMDM substantially improves reconstruction fidelity and distributional realism over StableMotion, while simultaneously reducing contact-related artifacts (skating and penetration) and temporal noise (acceleration error and jitter). In **inbetween**, CleanMDM outperforms ACMMDM and CondMDI on both geometric accuracy (MPJPE and keyframe error) and kinematic quality, indicating that the same backbone can handle interval completion and denoising within a unified formulation. In the most challenging **multimodal cleanup** setting, CleanMDM substantially outperforms GenMo (21.95 vs. 646.4 MPJPE), consistent with our expectation that a model designed for generation rather than cleanup does not explicitly preserve the input draft, resulting in lower reconstruction fidelity. Fig. 3 (A.1) visualizes representative examples under the same three settings, highlighting improved contact quality (e.g., reduced foot sliding) and stronger keyframe adherence compared with task matched baselines.

**Real-world validation.** Beyond synthetic corruption, we evaluate CleanMDM on real video-mocap outputs from animation, monocular video (HAA500, IDEA400, Kungfu) and multi-view video (HuMMan) sources, which contain artifacts beyond our training corruption types. Table 4 moves two representative public-dataset results (IDEA400 and HuMMan) into the main paper. On KungFu, CleanMDM cuts MPJPE roughly in half compared with StableMotion (69.4 vs. 127.8), while StableMotion attains a marginally lower jitter (9.2 vs. 12.0) by over-smoothing the motion in a way that alters its action identity; Appendix Fig. 5 visualizes this trade-off on a *Tornado Kick* clip, where StableMotion exhibits visible foot floating and a weakened rotational signature relative to the real-world input. Full public-dataset evaluations and real-world visualizations are provided in Appendix A.10 (Table 12, Figs. 5–6). These results confirm that CleanMDM’s motion prior generalizes to real-world degradations, not just the synthetic artifacts used during training.

**Effect of Mesh-Aware Contact Projection.** C6 applies our Mesh-Aware Contact Projection as a post-process. While global MPJPE can increase (due to sequence-level adjustments of root/ground alignment), C6 consistently reduces artifacts, including skating, penetration frequency/distance, and jitter, approaching the statistics of the ground truth. This behavior matches the role of Mesh-Aware Contact Projection as a projection step: it enforces hard contact consistency and temporal coherence, which are only weakly captured by regression-style training objectives. Therefore, CleanMDM provides a strong generative cleanup prior, while C6 further improves physical plausibility when strict contact quality is required. Fig. 4 shows a representative example on GVHMR outputs, where Mesh-Aware Contact Projection removes foot sliding and corrects floating artifacts.

Table 3: **C0–C6 configurations.** All variants share the same backbone.

| Config | Text | 2DKF | LMQD | 3DKF | MACP |
|--------|------|------|------|------|------|
| C0     |      |      |      |      |      |
| C1     | ✓    |      |      |      |      |
| C2     |      | ✓    |      |      |      |
| C3     | ✓    | ✓    |      |      |      |
| C4     | ✓    | ✓    | ✓    |      |      |
| C5     | ✓    | ✓    | ✓    | ✓    |      |
| C6     | ✓    | ✓    | ✓    | ✓    | ✓    |

Table 4: **Selected real-world public-dataset results.** Full results are in Appendix A.10, Table 12; MPJPE measures drift to the real-world input.

| Dataset | Method          | MPJPE<br>(mm)↓ | Jitter↓     | Slide<br>R↓ | Slide<br>A↓ |
|---------|-----------------|----------------|-------------|-------------|-------------|
| IDEA400 | Input           | 0.00           | 5.43        | 0.30        | 0.06        |
|         | StableMotion    | 40.10          | 5.10        | 0.45        | 0.14        |
|         | <b>CleanMDM</b> | <b>15.61</b>   | <b>4.94</b> | <b>0.25</b> | <b>0.05</b> |
| HuMMan  | Input           | 0.00           | 10.30       | 0.41        | 0.08        |
|         | StableMotion    | <b>84.02</b>   | 10.17       | 0.36        | 0.07        |
|         | <b>CleanMDM</b> | 84.72          | <b>7.80</b> | <b>0.32</b> | <b>0.06</b> |

#### 5.4 Ablation Studies

**Adding modalities improves controllable cleanup.** Table 2 shows that motion cleanup benefits from progressively adding complementary modalities. Starting from C0 (noisy 3D draft only), adding text (C1) provides coarse semantic and style guidance but yields limited improvements on geometric metrics. Adding sparse 2D keyframes (C2) consistently improves accuracy and keyframe adherence, suggesting that 2D cues act as an effective geometric scaffold even without explicit 2D-to-3D lifting. Combining 2D and text (C3) further improves the overall balance between fidelity and controllability.

Notably, the strongest gains appear when CleanMDM additionally leverages sparse 3D keyframes, which anchor the reconstruction inside corrupted intervals and reduce ambiguity. This trend supports the central motivation of multimodal cleanup: different cues compensate for each other under partial corruption, and a single model can exploit whichever signals are available at test time.

**LMQD improves distribution-level motion quality.** Comparing C3 and C4 isolates the effect of LMQD. Although MPJPE may not monotonically decrease, LMQD improves contact and temporal quality metrics (e.g., reduced penetration distance and jitter) and increases diversity relative to ground truth motion, indicating a better match to the motion manifold beyond point-wise regression. This supports our design choice of imposing a sequence-level adversarial regularizer in latent space: it is lightweight, compatible with flow matching, and directly targets artifacts that are perceptually salient yet poorly captured by deterministic reconstruction losses.

**Why latent fusion matters.** Replacing autoencoder-based 3D keyframe encoding with direct MLP projection in joint space (C5 (MLP), Appendix A.8 Table 10) degrades temporal quality dramatically (Jitter: 71.02 vs. 5.54; FID: 2.99 vs. 0.11). We attribute the gap to the autoencoder’s temporal downsampling bottleneck, which provides a normalized coordinate system and temporal-coherence prior that makes weighted fusion well-posed across heterogeneous signals; raw-space MLP encoding lacks this structure and creates a modality gap with the AE-encoded noisy draft. Moreover, multimodal training itself improves the motion prior: C5 outperforms C0 even when no extra conditions are given at inference (Appendix A.6, Table 8).

## 6 Conclusions, Limitations and Future Work

We presented CleanMDM, a unified framework for motion cleanup that formulates repair and inbetweening as masked conditional generation with plug-and-play multimodal conditions, including noisy 3D drafts, sparse 2D/3D keyframes, and text prompts. CleanMDM fuses heterogeneous geometric cues in a shared latent representation with frame-wise weighting, enabling the model to preserve reliable observations while regenerating corrupted regions under incomplete or mixed constraints. To better address perceptual artifacts that are weakly captured by regression-style objectives, we introduced LMQD, and further applied Mesh-Aware Contact Projection to enforce contact consistency and reduce skating and penetration. Experiments across auto repair, inbetweening, and multimodal cleanup demonstrate consistent improvements over prior baselines, with particularly strong gains on contact and temporal quality metrics.

A current limitation is that our evaluation relies on controlled corruption and a simplified ground model during Mesh-Aware Contact Projection. In future work, we plan to incorporate more realistic video mocap degradations, stronger learned contact priors, and additional control modalities and interactive authoring workflows for animation pipelines.

## References

- Z. Cai, D. Ren, A. Zeng, Z. Lin, T. Yu, W. Wang, X. Fan, Y. Gao, Y. Yu, L. Pan, et al. Humman: Multi-modal 4d human dataset for versatile sensing and modeling. In *European Conference on Computer Vision*, pages 557–577. Springer, 2022.
- B. Chen, Y. Li, Y. Zheng, Y.-X. Ding, and K. Zhou. Motion-example-controlled co-speech gesture generation leveraging large language models. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–12, 2025.
- R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- X. Chen, B. Jiang, W. Liu, Z. Huang, B. Fu, T. Chen, and G. Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18000–18010, 2023.
- J. Chung, C.-h. Wu, H.-r. Yang, Y.-W. Tai, and C.-K. Tang. Haa500: Human-centric atomic action dataset with curated videos. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13465–13474, 2021.
- S. Cohan, G. Tevet, D. Reda, X. B. Peng, and M. van de Panne. Flexible motion in-betweening with diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–9, 2024.
- K. Fan, S. Lu, M. Dai, R. Yu, L. Xiao, Z. Dou, J. Dong, L. Ma, and J. Wang. Go to zero: Towards zero-shot motion generation with million-scale data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13336–13348, 2025.
- S. Goel, G. Pavlakos, J. Rajasegaran, A. Kanazawa, and J. Malik. Humans in 4D: Reconstructing and tracking humans with transformers. In *ICCV*, 2023.
- C. Guo, S. Zou, X. Zuo, S. Wang, W. Ji, X. Li, and L. Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5152–5161, 2022.
- C. Guo, Y. Mu, M. G. Javed, S. Wang, and L. Cheng. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1900–1910, 2024.
- A. Henry, P. R. Dachapally, S. Pawar, and Y. Chen. Query-key normalization for transformers. *arXiv preprint arXiv:2010.04245*, 2020.
- J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- D. Holden. Robust solving of optical motion capture data by denoising. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.
- P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.
- B. Jiang, X. Chen, W. Liu, J. Yu, G. Yu, and T. Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023.
- K. Karunratanakul, K. Preechakul, S. Suwajanakorn, and S. Tang. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2151–2162, 2023.
- D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- L. Kovar, J. Schreiner, and M. Gleicher. Footskate cleanup for motion capture editing. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 97–104, 2002.

- J. Li, J. Cao, H. Zhang, D. Rempe, J. Kautz, U. Iqbal, and Y. Yuan. Genmo: A generalist model for human motion. *arXiv preprint arXiv:2505.01425*, 2025.
- Z. Li, W. Yuan, Y. He, L. Qiu, S. Zhu, X. Gu, W. Shen, Y. Dong, Z. Dong, and L. T. Yang. Lamp: Language-motion pretraining for motion generation, retrieval, and captioning. *arXiv preprint arXiv:2410.07093*, 2024.
- J. Lin, A. Zeng, S. Lu, Y. Cai, R. Zhang, H. Wang, and L. Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems*, 36:25268–25280, 2023.
- Z. Ling, B. Han, S. Li, J. Cheng, H. Shen, and C. Zou. Versatilemotion: A unified framework for motion synthesis and comprehension. *arXiv preprint arXiv:2411.17335*, 2024.
- Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023.
- I. Loshchilov, F. Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5):5, 2017.
- M. Luo, R. Hou, Z. Li, H. Chang, Z. Liu, Y. Wang, and S. Shan. M<sup>3</sup>gpt: An advanced multimodal, multitask framework for motion comprehension and generation. *Advances in Neural Information Processing Systems*, 37:28051–28077, 2024.
- Z. Luo, J. Cao, K. Kitani, W. Xu, et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10895–10904, 2023.
- N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, Oct. 2019.
- Z. Meng, Y. Xie, X. Peng, Z. Han, and H. Jiang. Rethinking diffusion for text-driven human motion generation. *arXiv preprint arXiv:2411.16575*, 2024.
- Z. Meng, Z. Han, X. Peng, Y. Xie, and H. Jiang. Absolute coordinates make motion generation easy. *arXiv preprint arXiv:2505.19377*, 2025.
- L. Mourot, L. Hoyet, F. L. Clerc, and P. Hellier. Underpressure: Deep learning for foot contact detection, ground reaction force estimation and footskate cleanup. In *Computer Graphics Forum*, volume 41, pages 195–206. Wiley Online Library, 2022.
- Y. Mu, H. Y. Ling, Y. Shi, I. B. Ojeda, P. Xi, C. Shu, F. Zinno, and X. B. Peng. Stablemotion: Training motion cleanup models with unpaired corrupted data. *arXiv preprint arXiv:2505.03154*, 2025.
- W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.
- X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021.
- M. Petrovich, M. J. Black, and G. Varol. Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9488–9497, 2023.
- E. Pinyoanuntapong, P. Wang, M. Lee, and C. Chen. Mmm: Generative masked motion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1546–1555, 2024.

- A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmlR, 2021.
- Z. Shen, H. Pi, Y. Xia, Z. Cen, S. Peng, Z. Hu, H. Bao, R. Hu, and X. Zhou. World-grounded human motion recovery via gravity-view coordinates. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.
- J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- C. Tessler, Y. Guo, O. Nabati, G. Chechik, and X. B. Peng. Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions on Graphics (TOG)*, 43(6):1–21, 2024.
- G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-or, and A. H. Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=SJ1kSy02jwu>.
- G. Tevet, S. Raab, S. Cohan, D. Reda, Z. Luo, X. B. Peng, A. H. Bermano, and M. van de Panne. Cload: Closing the loop between simulation and diffusion for multi-task character control. *arXiv preprint arXiv:2410.03441*, 2024.
- T. Wang, Z. Wu, Q. He, J. Chu, L. Qian, Y. Cheng, J. Xing, J. Zhao, and L. Jin. Stickmotion: Generating 3d human motions by drawing a stickman. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12370–12379, 2025a.
- Y. Wang, Q. Zhao, R. Yu, H. W. Tsui, A. Zeng, J. Lin, Z. Luo, J. Yu, X. Li, Q. Chen, et al. Skillmimic: Learning basketball interaction skills from demonstrations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17540–17549, 2025b.
- R. Williams. *The animator’s survival kit: a manual of methods, principles and formulas for classical, computer, games, stop motion and internet animators*. Macmillan, 2012.
- Y. Xie, V. Jampani, L. Zhong, D. Sun, and H. Jiang. Omnicontrol: Control any joint at any time for human motion generation. *arXiv preprint arXiv:2310.08580*, 2023.
- G. Xu, Y. Bian, A. Zeng, M. Shi, S. Huang, W. Li, L. Duan, and Q. Xu. Omnimotion-x: Versatile multimodal whole-body motion generation. *arXiv preprint arXiv:2510.19789*, 2025.
- H. Yao, Z. Song, Y. Zhou, T. Ao, B. Chen, and L. Liu. Moconvq: Unified physics-based motion control via scalable discrete representations. *ACM Transactions on Graphics (TOG)*, 43(4):1–21, 2024.
- Y. Yuan, J. Song, U. Iqbal, A. Vahdat, and J. Kautz. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16010–16021, 2023.
- J. Zhang, Y. Zhang, X. Cun, Y. Zhang, H. Zhao, H. Lu, X. Shen, and Y. Shan. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14730–14740, 2023a.
- M. Zhang, X. Guo, L. Pan, Z. Cai, F. Hong, H. Li, L. Yang, and Z. Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 364–373, 2023b.
- M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE transactions on pattern analysis and machine intelligence*, 46(6):4115–4128, 2024a.
- S. Zhang, B. L. Bhatnagar, Y. Xu, A. Winkler, P. Kadlec, S. Tang, and F. Bogo. Rohm: Robust human motion reconstruction via diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14606–14617, 2024b.

- Y. Zhang, R. Li, Y. Zhang, L. Pan, J. Wang, Y. Liu, and X. Li. A plug-and-play physical motion restoration approach for in-the-wild high-difficulty motions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13281–13292, 2025a.
- Z. Zhang, A. Liu, I. Reid, R. Hartley, B. Zhuang, and H. Tang. Motion mamba: Efficient and long sequence motion generation. In *European Conference on Computer Vision*, pages 265–282. Springer, 2024c.
- Z. Zhang, R. Liu, R. Hanocka, and K. Aberman. Tedi: Temporally-entangled diffusion for long-term motion synthesis. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024d.
- Z. Zhang, Y. Wang, W. Mao, D. Li, R. Zhao, B. Wu, Z. Song, B. Zhuang, I. Reid, and R. Hartley. Motion anything: Any to motion generation. *arXiv preprint arXiv:2503.06955*, 2025b.
- K. Zhao, G. Li, and S. Tang. Dartcontrol: A diffusion-based autoregressive motion model for real-time text-driven motion control. *arXiv preprint arXiv:2410.05260*, 2024.
- L. Zhong, C. Guo, Y. Xie, J. Wang, and C. Li. Sketch2anim: Towards transferring sketch storyboards into 3d animation. *ACM Transactions on Graphics (TOG)*, 44(4):1–15, 2025.
- J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

## A Supplementary Material

### A.1 Qualitative Comparisons

Figure 3 provides side-by-side qualitative comparisons under all three evaluation settings. In multi-modal cleanup (top), CleanMDM produces motion that closely follows the provided keyframes and text while maintaining natural contact, whereas GenMo drifts from the input draft. In auto cleanup (bottom-left), CleanMDM corrects localized foot sliding while preserving the reliable upper-body motion. In inbetweening (bottom-right), CleanMDM achieves smoother transitions and better keyframe adherence than CondMDI.

### A.2 Data

To emulate video mocap degradations, we synthesize corrupted motions by perturbing SMPL Loper et al. [2023] joint rotations and root translations, following the artifact injection protocol in Stable-Motion Mu et al. [2025] for part of the noise design. We apply a random subset of artifact types to a random temporal interval  $[s, e)$ , including joint jittering, foot sliding, horizontal drifting, and vertical floating. To ensure that corruptions remain localized and do not alter the clean segment outside the interval, we inject translation based artifacts in the velocity domain and integrate them only within  $[s, e)$ , while enforcing zero net displacement over the interval (i.e., the integrated root translation offset satisfies  $\Delta \mathbf{t}_e - \Delta \mathbf{t}_s = \mathbf{0}$ ), so that frames after  $e$  keep the original root trajectory unchanged. We record a per-frame defect label  $\text{det\_mask} \in \mathbb{N}^T$ , where  $\text{det\_mask}[t] = 0$  denotes a clean frame and non-zero values indicate corrupted frames.

**Data Sources** We build our training clips from AMASS Mahmood et al. [2019] and MotionLLaMA Ling et al. [2024]. MotionLLaMA originally contains approximately 48K motion-text pairs; however, part of this collection is noisy, lacks reliable text annotations, or comes from auxiliary sources that are less suitable for our setting. We therefore discard all MotionGV Fan et al. [2025] pseudo-labeled data, samples without text annotations, and most low-quality subsets from Motion Union, and retain only the subsets marked in Table 5. After filtering, about 28K MotionLLaMA clips remain for training. The final dataset is split into training, validation, and test sets with a ratio of 80:15:5.

**Absolute Joint Positions** We represent motion in absolute joint positions  $\mathbf{X}_{1:T} \in \mathbb{R}^{T \times J \times 3}$ , following ACMDM Meng et al. [2025]. This choice is motivated by the cleanup interface. First, absolute positions respond directly to spatial control signals: sparse 3D keyframes are already in the same coordinate system, and 2D keypoints are obtained through camera projection. Second, absolute coordinates avoid reconstructing global motion by accumulating local offsets, which can propagate small per-frame errors into long-horizon drift. Third, recent transformer and diffusion backbones Peebles and Xie [2023] can internalize many kinematic regularities from data, so a concise representation reduces modeling complexity without sacrificing fidelity. Finally, contact detection and inference-time IK operate naturally in the absolute domain, which facilitates stable enforcement of foot contact and ground constraints.

**Artifact types.** We simulate four common families: (i) **jittering** (high-frequency rotational noise on a subset of joints), (ii) **floating** (vertical root drift), (iii) **foot sliding** (spurious horizontal root motion causing skating), and (iv) **drifting** (slow horizontal root drift). The corrupted interval length is sampled to be localized (longer during training, shorter at test time).

**Localized corruption via integrated, zero-displacement perturbations.** For translation-based artifacts (foot sliding, drifting, floating), we inject noise in the *velocity* domain and integrate it *only* inside  $[s, e)$ , while enforcing approximately zero net displacement so that frames after  $e$  remain unchanged. Concretely, we generate a velocity perturbation  $\Delta \dot{\mathbf{t}}_{s:e}$ , taper it with a smooth window whose endpoints are zero, and remove its (windowed) mean so that the integrated offset cancels out:

$$\Delta \dot{\mathbf{t}}'_\ell = \Delta \dot{\mathbf{t}}_\ell - w_\ell \cdot \frac{\sum_{k=0}^{L-1} \Delta \dot{\mathbf{t}}_k}{\sum_{k=0}^{L-1} w_k + \epsilon}, \quad L = e - s, \quad (6)$$

followed by integration to obtain  $\Delta \mathbf{t}$  and applying  $\mathbf{t}'_{s:e} = \mathbf{t}_{s:e} + \Delta \mathbf{t}$  while keeping  $\mathbf{t}'_t = \mathbf{t}_t$  for  $t \notin [s, e)$ . In practice we additionally enforce an exact endpoint correction to eliminate numerical

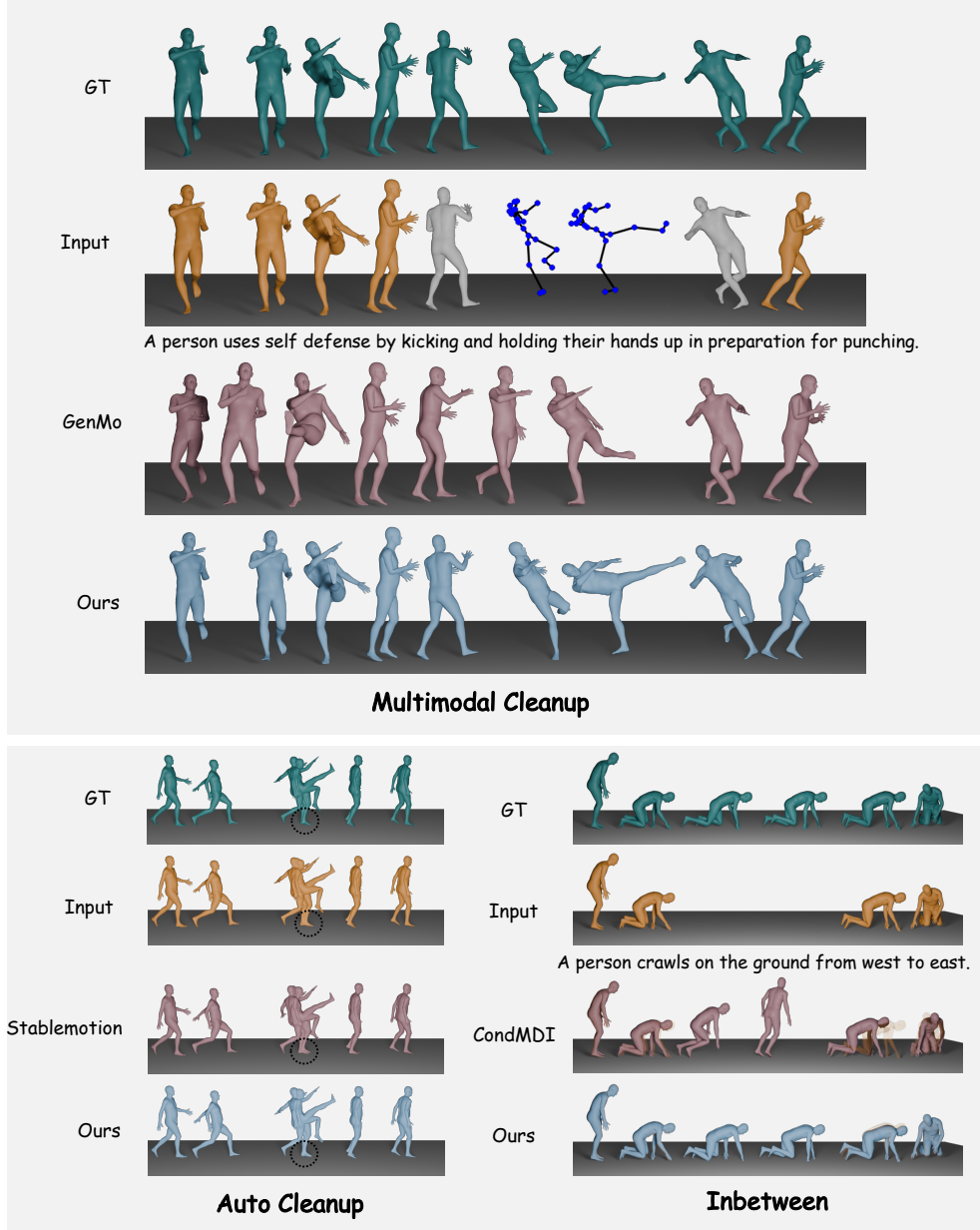


Figure 3: **Qualitative comparisons with task-matched baselines.** Top: **multimodal cleanup** compared with GenMo. The input consists of an unmasked noisy draft (yellow), sparse 3D keyframes (gray bodies), sparse 2D keyframes (blue skeleton), and a text prompt. Bottom-left: **auto cleanup** compared with StableMotion; CleanMDM corrects foot sliding (circled) while preserving the remaining motion. Bottom-right: **inbetween** compared with CondMDI; CleanMDM better satisfies the 3D keyframe constraints (yellow translucent overlays) and yields smoother transitions. Rows show *GT*, *Input*, *Baseline*, and *Ours*.

drift. This construction ensures the corruption is visually salient within the interval but does not affect the clean segment after it, enabling fair side-by-side evaluation.

**Relation to StableMotion.** Our artifact families and part of the injection protocol are adapted from StableMotion. A key difference is how we apply translation-based corruptions. Instead of directly perturbing root translations in position space (which can accumulate drift and contaminate frames outside the intended interval), we inject translation noise in the *velocity* domain and integrate it *only*

Table 5: MotionLLaMA-related subsets shown in the data manager.

| Source Group            | Dataset       | Used |
|-------------------------|---------------|------|
| Motion LLaMA            | HumanSC3D     | ✓    |
| Motion LLaMA            | FineDance     | ✓    |
| Motion LLaMA            | Fit3D         | ✓    |
| Motion LLaMA            | Hi4D          | ✓    |
| Motion LLaMA            | InterHuman    | ✗    |
| Motion LLaMA            | InterX        | ✓    |
| Motion LLaMA            | TruMans       | ✓    |
| Motion Union            | 100STYLE      | ✓    |
| Motion Union            | EgoBody       | ✗    |
| Motion Union            | Game Motion   | ✗    |
| Motion Union            | Platform      | ✗    |
| Motion Union            | Music         | ✗    |
| Motion Union            | KungFu        | ✗    |
| Motion Union            | IDEA400       | ✗    |
| Motion Union            | HuMMan        | ✗    |
| Motion Union            | HAA500        | ✗    |
| Motion Union            | Fitness       | ✗    |
| Motion Union            | Combat Motion | ✗    |
| Motion Union            | Animation     | ✗    |
| GVHMR Pseudo-label Data | MotionGV      | ✗    |

Table 6: **Effect of Mesh-Aware Contact Projection on video-mocap outputs.** We compare different post-processing strategies applied to GVHMR results on AIST++ and animation\_ref. Our Mesh-Aware Contact Projection improves contact quality (lower skating and penetration) and maintains a high success ratio, indicating stable optimization behavior across datasets.

| Method               | AIST++ (128) |                |             |             |             | animation_ref (20) |                |             |             |             |
|----------------------|--------------|----------------|-------------|-------------|-------------|--------------------|----------------|-------------|-------------|-------------|
|                      | Skate↓       | Jitter↓        | Pen.F↓      | Pen.D↓      | Succ.↑      | Skate↓             | Jitter↓        | Pen.F↓      | Pen.D↓      | Succ.↑      |
| GVHMR (CCD-IK)       | 0.13         | <b>23.1513</b> | 14.20       | 48.46       | <b>1.00</b> | 0.2594             | 48.0547        | 17.53       | 40.41       | <b>1.00</b> |
| GVHMR (CCD-IK) + PHC | 0.0913       | 30.7293        | <b>0.00</b> | <b>0.95</b> | 0.867       | 0.0725             | <b>34.2244</b> | <b>0.00</b> | 1.31        | 0.35        |
| GVHMR + MACP (ours)  | <b>0.056</b> | 27.3287        | 0.49        | 1.63        | <b>1.00</b> | <b>0.0583</b>      | 63.5704        | <b>0.00</b> | <b>0.78</b> | <b>1.00</b> |

within  $[s, e)$ , enforcing (approximately) zero net displacement over the interval. This construction ensures that the corrupted segment is visually salient while the clean motion before  $s$  and after  $e$  remains unchanged, enabling non-destructive cleanup evaluation with strictly preserved clean regions.

### A.3 Mesh-Aware Contact Projection: Additional Details

**General terrain formulation.** Given an input SMPL sequence with body pose  $\theta_{1:T}^{\text{in}}$  and root translation  $\mathbf{r}_{1:T}^{\text{in}}$ , MACP optimizes the body pose  $\theta_{1:T}$ , root translation  $\mathbf{r}_{1:T}$ , and, when needed, environment parameters  $\psi$ . Let  $\mathbf{v}_{t,i}^s = (x_{t,i}^s, y_{t,i}^s, z_{t,i}^s)$  denote the  $i$ -th sampled sole vertex of foot  $s \in \{L, R\}$  at frame  $t$ ; we use 54 vertices per foot. The environment is represented as a height field  $g_\psi(x, z)$ . For any sole vertex, define its signed height residual relative to the environment as

$$d_{t,i}^s = y_{t,i}^s - g_\psi(x_{t,i}^s, z_{t,i}^s). \quad (7)$$

The horizontal ground case used in our main experiments is the special case  $g_\psi(x, z) = h_g$ , where the scalar  $h_g$  is optimized together with the motion. If a scanned, reconstructed, or procedural terrain is available, the same losses apply by replacing the constant  $h_g$  with the corresponding height field.

**Objective.** Let  $c_{t,s} \in \{0, 1\}$  be the contact label for foot  $s$  at frame  $t$ , obtained from a pre-trained contact estimator Mourot et al. [2022] or from upstream predictions. The contact labels are thresholded at 0.5 in our implementation. Let  $\rho(\cdot)$  map axis-angle joint rotations to the 6D rotation

Table 7: **Loss weights used in Mesh-Aware Contact Projection.** The table expands the implementation weights for Eq. 8.

| Loss Category  | Loss Term                                      | Weight    |
|----------------|------------------------------------------------|-----------|
| Reconstruction | Rotation preservation                          | 1         |
| Reconstruction | Horizontal velocity preservation (non-contact) | $10^{-4}$ |
| Reconstruction | Vertical velocity preservation (non-contact)   | $10^{-3}$ |
| Foot contact   | Sole mesh height consistency                   | $10^{-1}$ |
| Foot contact   | Sole mesh non-penetration                      | $10^{-1}$ |
| Foot contact   | Zero velocity for contacting sole vertices     | $10^{-1}$ |
| Smoothness     | Rotational smoothness                          | $10^{-2}$ |
| Smoothness     | Translational smoothness                       | $10^{-4}$ |

representation, and let  $\Delta$  denote the first-order temporal difference. MACP minimizes

$$\begin{aligned} \mathcal{L}_{\text{MACP}} = & 1 \cdot \mathcal{L}_{\text{rot}} + 10^{-4} \cdot \mathcal{L}_{\text{vel}}^{xz} + 10^{-3} \cdot \mathcal{L}_{\text{vel}}^y \\ & + 10^{-1} \cdot \mathcal{L}_{\text{height}} + 10^{-1} \cdot \mathcal{L}_{\text{pen}} + 10^{-1} \cdot \mathcal{L}_{\text{lock}} \\ & + 10^{-2} \cdot \mathcal{L}_{\text{smooth}}^\theta + 10^{-4} \cdot \mathcal{L}_{\text{smooth}}^r. \end{aligned} \quad (8)$$

The reconstruction terms preserve the input motion:

$$\mathcal{L}_{\text{rot}} = \frac{1}{T} \sum_{t=1}^T \left\| \rho(\theta_t) - \rho(\theta_t^{\text{in}}) \right\|_2^2, \quad (9)$$

$$\mathcal{L}_{\text{vel}}^{xz} = \frac{1}{|\Omega_{\bar{c}}|} \sum_{(t,s) \in \Omega_{\bar{c}}} \left\| \Delta \mathbf{p}_t^{s,xz} - \Delta \mathbf{p}_t^{s,xz,\text{in}} \right\|_1, \quad (10)$$

$$\mathcal{L}_{\text{vel}}^y = \frac{1}{|\Omega_{\bar{c}}|} \sum_{(t,s) \in \Omega_{\bar{c}}} \left| \Delta p_t^{s,y} - \Delta p_t^{s,y,\text{in}} \right|. \quad (11)$$

Here  $\mathbf{p}_t^s$  denotes the ankle/toe foot joints and  $\Omega_{\bar{c}}$  indexes non-contact foot frames. The contact terms use the mesh sole vertices:

$$\mathcal{L}_{\text{height}} = \frac{1}{|\Omega_c|} \sum_{(t,s) \in \Omega_c} \frac{1}{|S_s|} \sum_{i \in S_s} (d_{t,i}^s)^2, \quad (12)$$

$$\mathcal{L}_{\text{pen}} = \frac{1}{|\Omega_{\bar{c}}|} \sum_{(t,s) \in \Omega_{\bar{c}}} \frac{1}{|S_s|} \sum_{i \in S_s} [\max(0, -d_{t,i}^s)]^2, \quad (13)$$

$$\mathcal{L}_{\text{lock}} = \frac{1}{|\Omega_{cc}|} \sum_{(t,s) \in \Omega_{cc}} \frac{1}{|S_s|} \sum_{i \in S_s} \left\| \mathbf{v}_{t+1,i}^s - \mathbf{v}_{t,i}^s \right\|_2^2, \quad (14)$$

where  $S_s$  is the sampled sole-vertex set for foot  $s$ ,  $\Omega_c$  indexes contact frames, and  $\Omega_{cc}$  indexes consecutive contact frames. Finally, temporal regularization suppresses jitter:

$$\mathcal{L}_{\text{smooth}}^\theta = \frac{1}{T-1} \sum_{t=1}^{T-1} \left\| \rho(\theta_{t+1}) - \rho(\theta_t) \right\|_2^2, \quad (15)$$

$$\mathcal{L}_{\text{smooth}}^r = \frac{1}{T-1} \sum_{t=1}^{T-1} \left\| \mathbf{r}_{t+1} - \mathbf{r}_t \right\|_2^2. \quad (16)$$

**Optimization.** We optimize the full sequence with Adam for 600 iterations using an initial learning rate of  $10^{-2}$  and decay it by 0.1 every 250 iterations. The mesh-based terms are evaluated on the sampled SMPL sole vertices, while the non-contact velocity preservation terms use ankle/toe joints to preserve the original swing-foot motion.

#### A.4 Additional Evaluation of Mesh-Aware Contact Projection

To further validate the effectiveness of our Mesh-Aware Contact Projection, we perform an auxiliary study on motions produced by a representative video-mocap system (GVHMR) under different

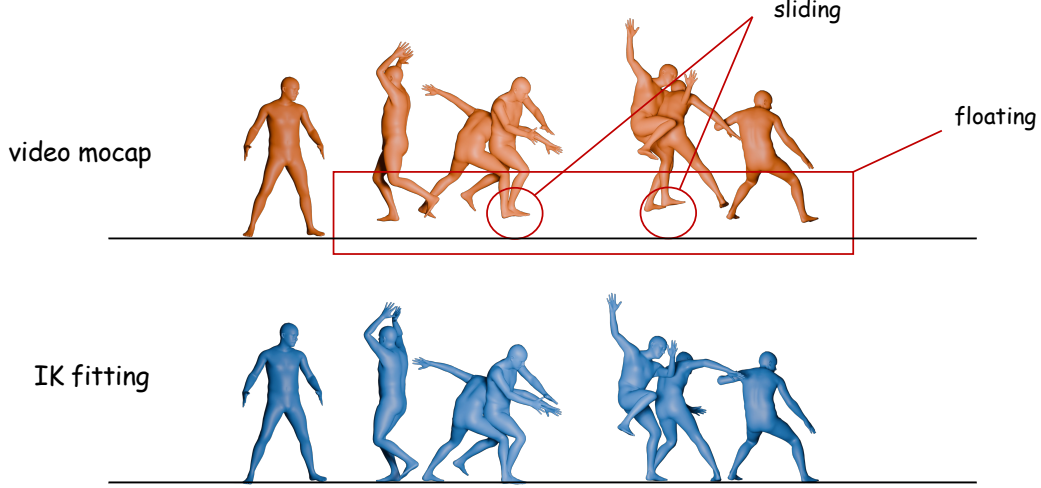


Figure 4: **Effect of test-time Mesh-Aware Contact Projection on video mocap outputs.** Top: raw GVHMR Shen et al. [2024] results exhibit contact artifacts, including foot sliding and occasional floating (highlighted). Bottom: applying our Mesh-Aware Contact Projection as a post-process enforces more consistent ground contact, reducing sliding and correcting floating while preserving the overall motion.

post-processing strategies. We evaluate on two datasets (AIST++ and animation\_ref) and report contact- and temporal-quality metrics, including Foot Skating Ratio, Jitter Error, Ground Penetration Frequency/Distance, together with a Success Ratio indicating whether the optimization converges to a feasible solution under our constraints.

Table 6 shows that applying our Mesh-Aware Contact Projection consistently improves the kinematic plausibility of GVHMR outputs, substantially reducing foot skating and ground penetration compared to the standard CCD-IK post-processing used in GVHMR. Figure 4 visualizes the effect of Mesh-Aware Contact Projection on GVHMR outputs. The raw video-mocap results (top) exhibit visible foot sliding and occasional floating. After applying our contact projection (bottom), ground contact becomes consistent, sliding is eliminated, and floating is corrected, while the overall motion trajectory and style are preserved. Compared to physics-based refinement (PHC) on top of CCD-IK, our method achieves a more reliable success ratio while still providing strong gains on penetration-related metrics. Overall, this study supports the role of Mesh-Aware Contact Projection as an effective, lightweight projection step that enhances contact consistency for video-mocap motions, and explains why we adopt it as a plug-in post-process in our pipeline.

## A.5 Evaluation Metrics

We evaluate reconstruction fidelity, motion realism, and physical/temporal plausibility. For distributional realism, we report **FID** computed on features from an independently trained motion autoencoder Ling et al. [2024], and **Diversity (Div)** as the average pairwise distance between randomly sampled generated embeddings. For reconstruction accuracy, we report **MPJPE** (mm) and report **local MPJPE** in parentheses after removing the global root translation.

For kinematic and contact quality, we report **Skate** Zhang et al. [2024b] as the foot skating ratio: a foot joint is considered near-ground if its height is within 0.08m above the ground level (estimated as the minimum GT height), and is considered skating if its horizontal speed exceeds 0.1m/s; Skate is the fraction of near-ground foot samples that satisfy the skating condition. We also report ground penetration as **Pen.F** (%) and **Pen.D** (mm), computed from toe joints: Pen.F is the percentage of toe samples below the ground by more than 0.05m, and Pen.D is the mean penetration depth over all toe samples.

For temporal quality, we report **Acc Err** ( $\text{m/s}^2$ ) Zhang et al. [2024b], the mean acceleration difference to GT computed by second-order finite differences scaled by fps, and **Jitter** Shen et al. [2024],

Table 8: **Multimodal training improves noisy-only denoising.** C0 and C5 are evaluated on the same auto repair setup with identical noisy inputs and *no* mask guidance or extra modalities at test time. Lower is better ( $\downarrow$ ) except Div ( $\uparrow$ ).

| Method | MPJPE $\downarrow$ | Skate $\downarrow$ | Acc Err $\downarrow$ | Pen.F $\downarrow$ | Pen.D $\downarrow$ | Jitter $\downarrow$ | FID $\downarrow$ | Div $\rightarrow$ |
|--------|--------------------|--------------------|----------------------|--------------------|--------------------|---------------------|------------------|-------------------|
| GT     | 0.00               | 0.13               | 0.00                 | 0.00               | 0.00               | 3.73                | 0.00             | 9.40              |
| C0     | 13.8               | 0.20               | 1.87                 | 0.07               | 1.09               | 5.77                | 0.10             | 6.53              |
| C5     | <b>12.8</b>        | <b>0.19</b>        | <b>1.78</b>          | <b>0.03</b>        | <b>0.79</b>        | <b>5.49</b>         | <b>0.06</b>      | <b>8.14</b>       |

computed from third-order finite differences (scaled by  $\text{fps}^3$  and normalized as in our evaluator). Finally, we also report **Keyframe Error (KF Err)**, defined as MPJPE computed *only* on the user-specified control frames (keyframes) used for inbetweening/constraint satisfaction. **M2M Score / M2M R@3** Petrovich et al. [2023] measure motion-to-motion retrieval consistency between generated and ground truth motions.

#### A.6 Does Multimodal Training Help Even for Noisy-only Denoising?

We further test whether multimodal training improves motion denoising even when *no additional modalities are provided at inference*. Specifically, we compare C0 (noisy 3D only) and C5 (our full multimodal model) under the same **auto repair** setting: both models take the *same* noisy motion as input, and we do not provide interval masks or any extra conditions at test time. As shown in Table 8, C5 consistently outperforms C0 on reconstruction fidelity and motion quality metrics (e.g., skating, penetration, temporal errors), suggesting that multimodal supervision improves the learned motion prior beyond single-modality training. We attribute the gains to (i) richer effective training pairings that reduce overfitting to a single modality under limited motion data, (ii) complementary supervisory signals that encourage more accurate conditional modeling during training, and (iii) a more general representation of motion dynamics that transfers back to denoising even when extra cues are absent at inference.

#### A.7 Additional Ablations on Condition Encoders

We provide additional ablations on how textual and 2D conditions are encoded and injected. All metrics follow the main paper, and all settings are identical to the corresponding baseline variant unless otherwise specified.

**2D condition encoder: lightweight MLP vs. MotionBERT features.** To verify that our model can effectively utilize 2D cues, we ablate the design of the 2D condition encoder while keeping all other components, training settings, and data identical. Specifically, we compare (i) a lightweight 2D encoder that directly flattens the downsampled 2D keypoints ( $22 \times 2$  per frame) and maps them to the model latent space with a small MLP, versus (ii) a MotionBERT-based 2D feature extractor whose outputs are used as the 2D condition tokens. As shown in Table 9, directly encoding 2D keypoints with a simple MLP consistently outperforms the MotionBERT-based alternative across our evaluation suite. This indicates that, under our training setup, the model can learn useful 2D-to-motion correspondence from raw keypoints and does not require an explicit 2D-to-3D lifting module before conditioning. Since the generative backbone already operates in a learned 3D motion latent space, the 2D branch mainly needs to preserve observed image-plane geometry and project it to the model width. MotionBERT features are stronger for generic 3D pose representation learning, but in our cleanup setting they may introduce a pretrained representation bias and a feature-space mismatch with the autoencoder latents used by the noisy 3D draft and clean 3D keyframes. The lightweight MLP therefore gives a better trade-off: it preserves direct geometric constraints while letting the cleanup prior infer the compatible 3D motion.

#### A.8 Ablation on Latent-Space vs. Joint-Space 3D Conditioning

A central design choice in CleanMDM is to encode all 3D geometric conditions (noisy draft and clean keyframes) through the *same* motion autoencoder before fusion, rather than projecting raw joint positions with an MLP. To isolate the effect of this choice, we compare the full model C5

Table 9: **Ablation on the 2D condition encoder.** All settings are identical except for the 2D branch.

| Method             | MPJPE↓       | Skate↓      | Acc Err↓    | Pen.F↓      | Pen.D↓      | Jitter↓     | FID↓        | Div→        |
|--------------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| GT                 | 0.00         | 0.13        | 0.00        | 0.00        | 0.00        | 3.73        | 0.00        | 9.40        |
| C2 (2D MLP; ours)  | <b>27.45</b> | <b>0.20</b> | <b>2.17</b> | <b>0.07</b> | <b>0.89</b> | <b>6.12</b> | <b>0.07</b> | 8.05        |
| C2 (MotionBERT 2D) | 31.45        | 0.21        | 2.24        | 0.10        | <b>0.89</b> | 6.25        | 0.12        | <b>9.20</b> |

Table 10: **Latent-space vs. joint-space 3D conditioning (multimodal cleanup).** C5 (MLP) replaces the autoencoder-based 3D keyframe encoder with an MLP in joint space. All other components are identical to C5.

| Method        | MPJPE↓       | Skate↓      | Acc Err↓    | Pen.F↓      | Pen.D↓      | Jitter↓     | KF Err↓     | FID↓        | Div→        |
|---------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| GT            | 0.00         | 0.13        | 0.00        | 0.00        | 0.00        | 3.73        | 0.00        | 0.00        | 9.40        |
| C5 (MLP)      | 35.59        | <b>0.10</b> | 14.94       | 0.08        | 0.73        | 71.02       | 30.44       | 2.99        | 8.91        |
| C5 (CleanMDM) | <b>21.95</b> | 0.18        | <b>1.98</b> | <b>0.06</b> | <b>0.76</b> | <b>5.54</b> | <b>11.4</b> | <b>0.11</b> | <b>9.23</b> |

(CleanMDM) against C5 (MLP), a variant that replaces autoencoder-based 3D keyframe encoding with a lightweight MLP that maps raw joint positions directly to the model width. All other components — the flow-matching backbone, text conditioning, 2D keyframe branch — remain identical.

As shown in Table 10, C5 (MLP) degrades dramatically compared to C5 on temporal quality (Jitter: 71.02 vs. 5.54, Acc: 14.94 vs. 1.98) and distributional realism (FID: 2.99 vs. 0.11), while also showing worse reconstruction accuracy (MPJPE: 35.59 vs. 21.95) and keyframe adherence (KF Err: 30.44 vs. 11.4). Three observations emerge:

**(i) The AE bottleneck provides a temporal-coherence prior.** The autoencoder’s temporal down-sampling ( $d=4$ ) forces the encoder to produce representations that are inherently smooth across frames. An MLP encoding each frame independently lacks this inductive bias. As a result, C5 (MLP) can match individual keyframe positions but produces incoherent transitions between them, explaining the catastrophic jitter ( $12.8\times$ ) and acceleration error ( $7.5\times$ ).

**(ii) Shared representation space stabilizes weighted fusion.** In C5, both the noisy 3D draft and clean 3D keyframes are encoded by the same autoencoder, so the confidence-weighted sum in Eq. (2) operates in a common latent space with aligned scale and temporal rate. In C5 (MLP), the MLP-encoded keyframe tokens occupy a different representation space from the AE-encoded noisy draft, creating a modality gap that destabilizes fusion and undermines conditioning quality.

**(iii) Distributional quality requires representation-level alignment.** The  $27\times$  FID degradation indicates that raw-space encoding does not merely reduce point-wise accuracy — it fundamentally disrupts the generated motion distribution. C5 (MLP) produces motions that deviate sharply from natural motion statistics, even though its skating ratio is nominally lower (0.10 vs. 0.18), suggesting that the low skating comes at the cost of unnatural, jerky motion rather than genuine physical plausibility.

## A.9 Ablation on LMQD Feature Representations

In the main paper, we introduce the *Latent Motion Quality Discriminator* (**LMQD**) as a lightweight adversarial regularizer that operates on entire sequences and improves kinematic realism beyond regression-style objectives. Here we ablate the *feature representation* fed to LMQD to verify that the latent sequence is the most effective choice.

**Setup.** We isolate this factor in a pure denoising setting (blind auto repair): all variants share the same diffusion backbone, training protocol, and noisy inputs, and differ only in the discriminator feature. We compare a diffusion-only baseline (**C0**) against **C0+LMQD** with different discriminator inputs: (i) **latent seq** (ours), and several alternatives computed from joint-space motion statistics, including (ii) **power seq**, (iii) **joint seq**, (iv) **AMP seq**, a handcrafted state feature sequence inspired

Table 11: **Ablation on LMQD feature representations (denoising).** All variants use the same denoising setup; only the discriminator feature is changed.

| Method                     | MPJPE↓      | Skate↓        | Acc Err↓    | Pen.F↓      | Pen.D↓      | Jitter↓       |
|----------------------------|-------------|---------------|-------------|-------------|-------------|---------------|
| C0 (diffusion only)        | 13.8        | 0.2008        | <b>1.87</b> | 0.07        | 1.09        | 6.5286        |
| C0+LMQD (latent seq, ours) | <b>11.8</b> | <b>0.1780</b> | 1.96        | <b>0.02</b> | <b>0.46</b> | <b>6.3115</b> |
| C0+LMQD (power seq)        | 13.3        | 0.1813        | 1.96        | 0.05        | 0.91        | 6.4056        |
| C0+LMQD (joint seq)        | 13.3        | 0.1843        | 2.01        | 0.04        | 0.65        | 6.5149        |
| C0+LMQD (AMP seq)          | 13.3        | 0.1847        | 2.01        | 0.04        | 0.69        | 6.5257        |
| C0+LMQD (accel seq)        | 13.3        | 0.1848        | 2.01        | 0.04        | 0.66        | 6.5167        |

Table 12: **Generalization on public datasets.** We report MPJPE to the real-world input motion (lower indicates less drift), together with temporal smoothness (Jitter) and contact quality (Slide R/A). These public datasets do not provide ground-truth cleaned motions. CleanMDM consistently reduces sliding and jitter while maintaining fidelity to the real-world input motion, indicating a transferable motion prior that improves cleanup without over-smoothing.

| Dataset   | Method          | MPJPE (mm)↓  | Jitter↓      | Slide R↓    | Slide A↓    |
|-----------|-----------------|--------------|--------------|-------------|-------------|
| IDEA400   | Input           | 0.00         | 5.43         | 0.30        | 0.06        |
|           | StableMotion    | 40.10        | 5.10         | 0.45        | 0.14        |
|           | <b>CleanMDM</b> | <b>15.61</b> | <b>4.94</b>  | <b>0.25</b> | <b>0.05</b> |
| Animation | Input           | 0.00         | 17.60        | 0.54        | 0.15        |
|           | StableMotion    | 138.61       | <b>11.27</b> | 0.60        | 0.11        |
|           | <b>CleanMDM</b> | <b>69.40</b> | 15.95        | <b>0.50</b> | <b>0.11</b> |
| HAA500    | Input           | 0.00         | 20.78        | 0.62        | 0.14        |
|           | StableMotion    | 147.50       | <b>10.93</b> | 0.58        | 0.11        |
|           | <b>CleanMDM</b> | <b>57.83</b> | 12.87        | <b>0.53</b> | <b>0.10</b> |
| HuMMan    | Input           | 0.00         | 10.30        | 0.41        | 0.08        |
|           | StableMotion    | <b>84.02</b> | 10.17        | 0.36        | 0.07        |
|           | <b>CleanMDM</b> | 84.72        | <b>7.80</b>  | <b>0.32</b> | <b>0.06</b> |
| KungFu    | Input           | 0.00         | 16.63        | 0.62        | 0.12        |
|           | StableMotion    | 127.84       | <b>9.15</b>  | 0.56        | 0.11        |
|           | <b>CleanMDM</b> | <b>69.40</b> | 11.95        | <b>0.56</b> | <b>0.11</b> |

by the adversarial motion prior used in PHC/AMP-style pipelines Luo et al. [2023], Peng et al. [2021] (e.g., concatenating joint positions/velocities and contact-related signals over time), and (v) **accel seq**

**Results.** Table 11 shows that using the *latent sequence* as the LMQD input provides the best overall trade-off. Compared to **C0**, **C0+LMQD (latent seq)** consistently reduces foot skating and ground penetration while improving MPJPE, and it outperforms alternative feature choices under the same training budget. This supports the design used throughout our method: the autoencoder latent offers a compact, temporally aligned representation that is well-suited for sequence-level discrimination, yielding a stronger distributional constraint on motion quality.

## A.10 Generalization to Public Datasets

**Cleanup on out-of-domain motions.** We evaluate on five public datasets spanning different capture modalities. **IDEA400** Lin et al. [2023] and **KungFu** Lin et al. [2023] are monocular video-based motion capture subsets from Motion-X, covering everyday actions and martial arts sequences respectively. **HAA500** Chung et al. [2021] is a monocular video dataset containing 500 human action categories. **HuMMan** Cai et al. [2022] is a multi-view RGB-D dataset with higher 3D reconstruction accuracy due to synchronized multi-camera capture. **Animation** Ling et al. [2024] consists of artist-authored keyframe animation data; unlike video-based sources, it exhibits inherently higher jitter values owing to the discrete keyframe interpolation process rather than capture noise. Table 12 evaluates our best cleanup model (without Mesh-Aware Contact Projection) on several public datasets and compares it to StableMotion. Across datasets, our method consistently reduces sliding and jitter relative to StableMotion, while keeping the motion closer to the real-world input motion (MPJPE).

Table 13: **Complete user-study results.** Mean ratings from 10 professional animators. Higher is better.

| Setting            | Method          | Artifacts $\uparrow$ | Naturalness $\uparrow$ | Faithfulness $\uparrow$ |
|--------------------|-----------------|----------------------|------------------------|-------------------------|
| Auto cleanup       | StableMotion    | 3.45                 | 3.61                   | 2.33                    |
|                    | <b>CleanMDM</b> | <b>4.13</b>          | <b>4.41</b>            | <b>4.63</b>             |
| Inbetween          | CondMDI         | 2.91                 | 3.08                   | 2.29                    |
|                    | <b>CleanMDM</b> | <b>3.93</b>          | <b>4.07</b>            | <b>4.33</b>             |
| Multimodal cleanup | GenMo           | 3.20                 | 3.50                   | 3.66                    |
|                    | <b>CleanMDM</b> | <b>3.93</b>          | <b>4.12</b>            | <b>4.22</b>             |

This suggests that CleanMDM learns a transferable motion prior that improves temporal stability and contact behavior without over-smoothing or drifting the underlying performance.

**KungFu visualization and the Jitter trade-off.** Figures 5 and 6 present two representative KungFu (Motion-X) clips, (a) *Tornado Kick* and (b) *Play basketball*, comparing the real-world input, StableMotion, and CleanMDM as horizontally-unfolded SMPL mesh sequences. The figures make the Jitter row of Table 12 interpretable: StableMotion’s lower Jitter on KungFu (9.2 vs. 12.0) is achieved by aggressively smoothing the temporal trajectory, which simultaneously detaches the support foot from the ground (floating) and weakens the kick’s rotational signature, yielding a much higher MPJPE (127.8 vs. 69.4). CleanMDM instead preserves the action’s spatial structure and contact behavior, at the cost of a slightly higher numerical jitter that mirrors the high-frequency content of the input motion itself. This is consistent with the role of CleanMDM as a generative cleanup prior conditioned on the input rather than a temporal smoother.

Since jitter is difficult to assess from static unfolded meshes, Figs. 5 and 6 focus on visible ground-ing/contact artifacts such as floating and foot-ground consistency. Together, these static examples show that CleanMDM preserves the action semantics while reducing penetration/floating and producing smoother motion with less visible jitter.

### A.11 User Study

**Protocol.** We conduct a perceptual study to evaluate whether CleanMDM improves motion quality in ways that are salient to animators. We consider three practical tasks: **auto cleanup**, **inbetween**, and **multimodal cleanup**. For each task, we randomly sample 10 clips from the test set and compare CleanMDM against a task-matched baseline (StableMotion for auto cleanup, CondMDI for inbetween, and GenMo for multimodal cleanup). For each clip, we present a *paired* comparison (CleanMDM vs. baseline) with randomized left/right order and anonymized method names.

**Questions.** To reduce rater fatigue while capturing production-relevant qualities, raters score each method on three criteria: **Artifacts**, **Naturalness**, and **Faithfulness**. Artifacts summarize contact and temporal issues (e.g., foot skating, ground penetration, and jitter). Naturalness evaluates whether the motion looks physically and stylistically plausible. Faithfulness measures consistency with the reference (ground truth for our benchmark; or the provided constraints for multimodal cleanup). All scores are given on a 0–5 scale, where higher is better.

**Participants and results.** We recruit 10 professional animators to perform the study. All participants are colleagues who volunteered; no monetary compensation was provided. Table 13 reports mean ratings aggregated over all clips and participants for each task. CleanMDM is consistently preferred over task-matched baselines across all three criteria, with the largest gains observed in *Faithfulness* and *Artifacts*.

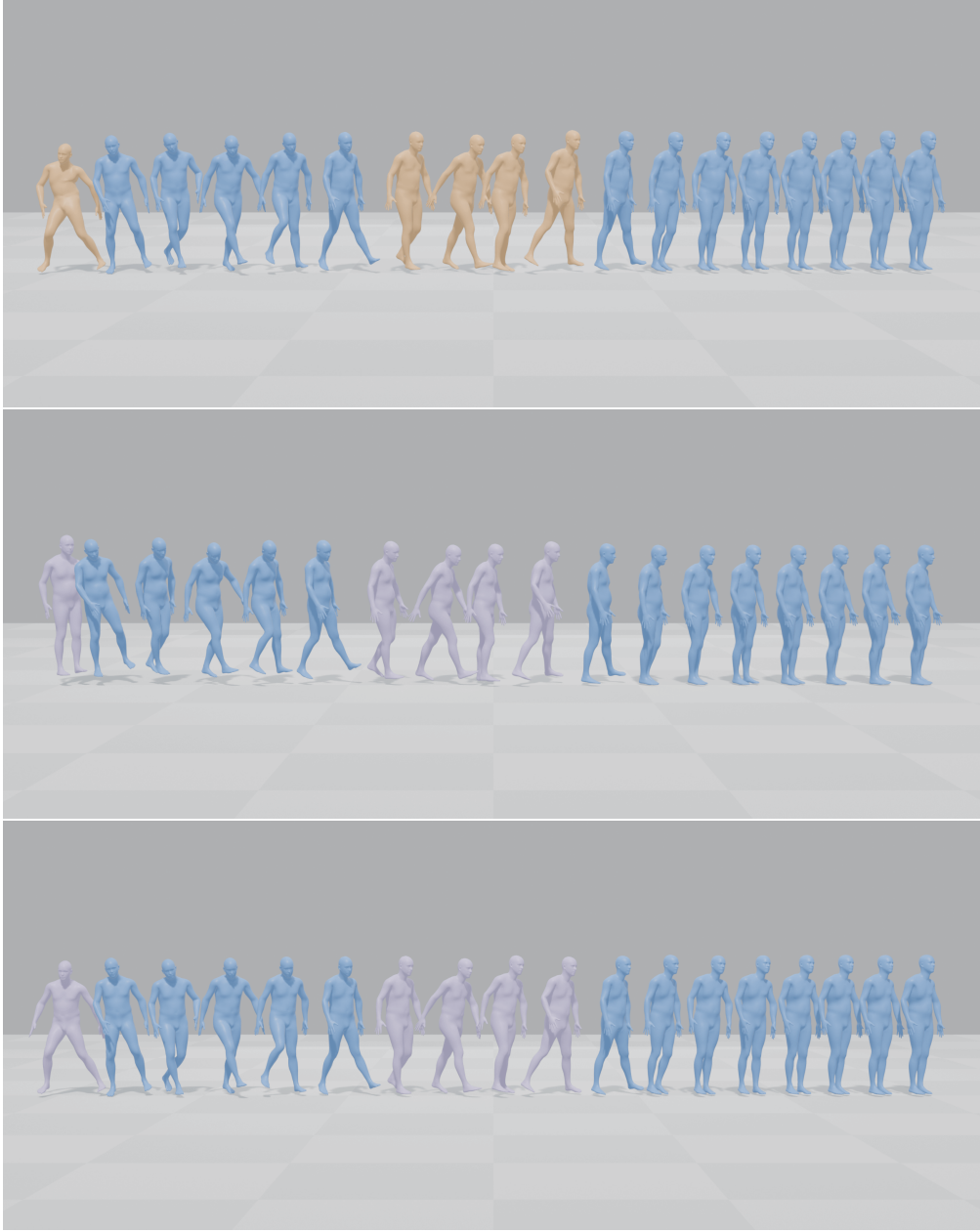


Figure 5: (a) **Real-world cleanup on KungFu Tornado Kick (Motion-X) Lin et al. [2023]**. Top to bottom: real-world input, StableMotion Mu et al. [2025], and **CleanMDM (ours)**. Orange bodies mark low-quality frames in the real-world input, while purple bodies mark the corresponding repaired frames in StableMotion and CleanMDM. StableMotion shows visible support-foot floating and dampened rotation; CleanMDM closely tracks the input motion. See the preceding paragraph for the MPJPE/Jitter trade-off, and Fig. 6 for a second example on the same dataset.

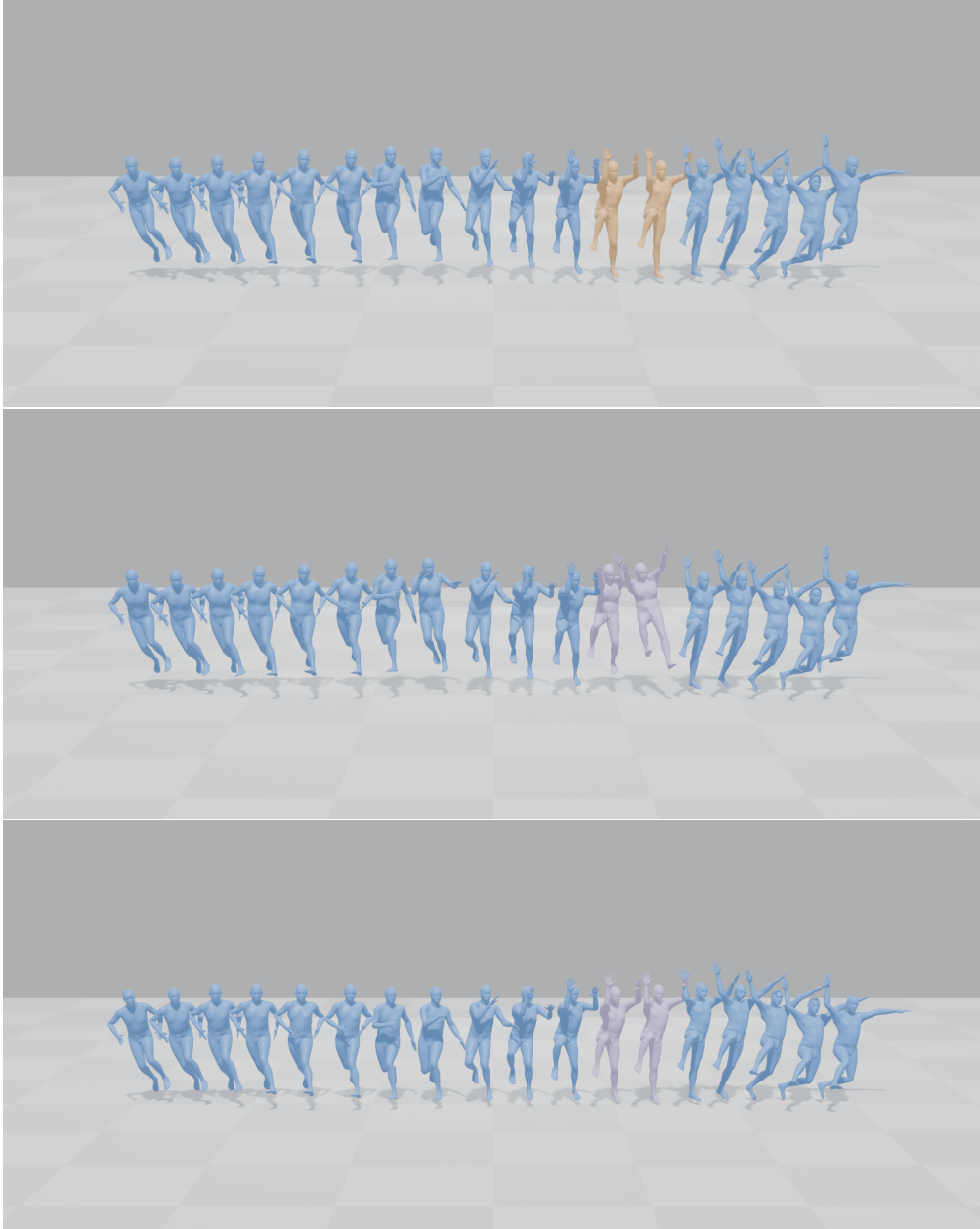


Figure 6: **(b) Additional KungFu example: *Play basketball*.** Top to bottom: real-world input, StableMotion Mu et al. [2025], and **CleanMDM (ours)**. Orange bodies mark low-quality frames in the real-world input, while purple bodies mark the corresponding repaired frames in StableMotion and CleanMDM. StableMotion lifts the subject off the ground in one of these frames, breaking the grounded action semantics; CleanMDM preserves the input contact behavior. Companion to Fig. 5.