
VideoMDM: Towards 3D Human Motion Generation From 2D Supervision

Amir Mann
Technion
Haifa, Israel

Gal Harari
Technion
Haifa, Israel

Merav Keidar
Technion
Haifa, Israel

Or Litany
Technion & NVIDIA
Haifa, Israel

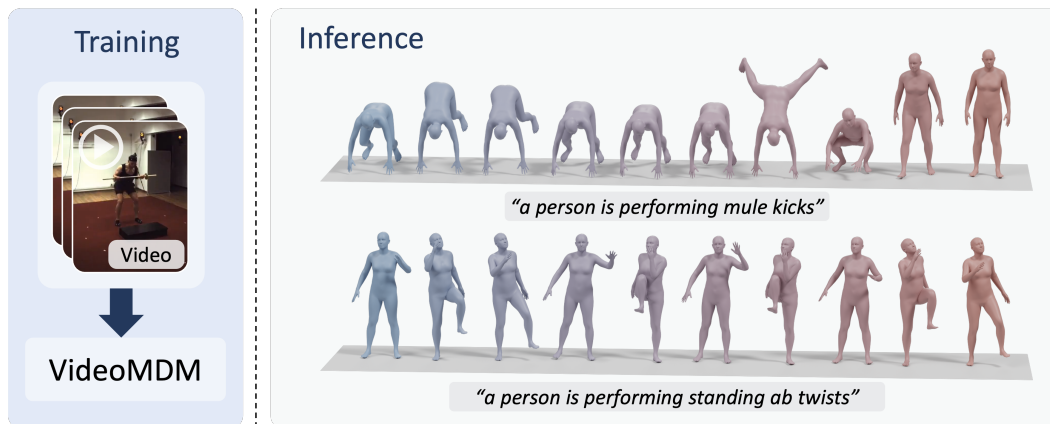


Figure 1: **We demonstrate VideoMDM on monocular videos of human activities.** Our framework trains 3D text-to-motion diffusion models using 2D pose sequences extracted from videos. Left: representative training videos. Right: generated motions using the trained model from text prompts. Despite relying solely on 2D supervision, VideoMDM attains motion fidelity approaching that of fully 3D-supervised training.

Abstract

We introduce VideoMDM, a diffusion-based framework that trains 3D human motion priors directly from accurate 2D poses extracted from monocular videos, without any 3D ground truth. A pretrained 2D-to-3D lifter provides approximate 3D pose sequences that serve as a noisy teacher: these are diffused, denoised by the model in 3D, and supervised in 2D by reprojecting the prediction and comparing against accurate keypoints. We show that, under mild assumptions, a depth-weighted 2D reprojection loss is equivalent in expectation to direct 3D supervision, and we adapt standard 3D motion regularizers — velocity consistency and over-parameterized representation alignment — to this 2D setting. Unlike methods that lift 2D to 3D only at inference, VideoMDM learns a coherent 3D motion manifold during training. On HumanML3D it nearly closes the gap to fully 3D-supervised MDM (FID 0.88 vs 0.54); On real video datasets Fit3D and NBA the method learns to generate motions consistently preferred by humans, with strong quantitative results.

1 Introduction

Generating realistic 3D human motion is central to animation, gaming, simulation, and embodied AI. Diffusion-based motion models such as MDM [46] have recently achieved striking realism when trained on motion-capture (MoCap) data. Yet, their success remains tightly coupled to the availability of high-quality 3D supervision: MoCap datasets are captured in controlled studio environments and span only a narrow subset of real-world movement. Models trained on them inherit limited diversity and fail to capture the richness of human motion observed in the wild.

At the same time, vast amounts of online video depict human actions across diverse environments, identities, and viewpoints. Harnessing such in-the-wild data could enable scalable and diverse 3D motion generation. However, most videos are monocular, lacking the multi-view cues necessary for reliable 3D reconstruction. While monocular 3D pose and motion estimators [57, 41, 48, 25] remain noisy and ambiguous, 2D keypoint detectors [5, 8, 18] have reached high accuracy and robustness. The key challenge, therefore, is how to train a generative 3D motion model using only accurate 2D supervision derived from monocular video.

We address this challenge by introducing VideoMDM, a diffusion-based framework for training 3D human motion models entirely from 2D pose supervision. Unlike prior approaches that triangulate 2D motions to 3D only at inference time [19, 25], which prevents the model from learning a consistent 3D prior, or that depend on 3D supervision for fine-tuning [15], VideoMDM trains a diffusion model natively in 3D space using only 2D supervision. Our formulation opens, for the first time, a path towards large-scale training of 3D text-to-motion diffusion from monocular videos without any MoCap data.

Building on cross-modality training of Image-to-3D diffusion [34], we adopt a noisy-teacher scheme: a pretrained 2D-to-3D lifter produces approximate 3D pose sequences from 2D inputs; these are diffused, denoised by the model in 3D, and supervised in 2D by reprojecting the prediction and comparing against accurate keypoints from video. This design lets the model learn a coherent 3D motion manifold grounded entirely in 2D observations.

To make 2D-only diffusion training practical for 3D motion, we introduce a depth-aware reprojection loss that, under mild assumptions on data and camera distribution, is provably equivalent in expectation to standard 3D MSE supervision (Section 3.4, proof in Section A). We further adapt two standard 3D motion regularizers to the 2D setting: a depth-weighted 2D velocity loss for temporal coherence, and a motion representation alignment loss that supervises the over-parameterized motion channels — joint rotations, joint velocities, and foot contacts — via ray-projection pseudo-targets, since no 3D ground truth is available for them (Section 3.5).

We evaluate VideoMDM in three regimes. (i) On a 2D-only version of HumanML3D [13], where 2D poses are obtained by projecting MoCap, VideoMDM achieves FID 0.88 — nearly closing the gap to fully 3D-supervised MDM (FID 0.54) and improving on the strongest 2D-supervised baseline by roughly $\times 2$. (ii) On Fit3D [9] — a real-world setting, where training uses monocular fitness video with extracted 2D keypoints and no 3D supervision — VideoMDM halves joint error against WHAM [41] on motions far outside the lifter’s distribution (MPJPE 111 vs 228 mm) and produces $5.5\times$ smoother motion (Accel 3.16 vs 17.66 m/s^2). And is preferred by humans in generation against all baselines. (iii) On the NBA dataset [19], VideoMDM is preferred over MAS in 64% of pairwise human comparisons. Together, these results show that 2D supervision alone is sufficient to learn 3D motion priors that are coherent, perceptually realistic, and capable of generalizing beyond the lifter that bootstrapped them.

Our main contributions are:

1. **The first 2D-supervised diffusion training framework for 3D human motion priors**, enabling high-fidelity prior learning directly from monocular videos without any 3D supervision.
2. **A stabilization strategy for condition-free denoising**, leveraging depth-aware weighting and reprojection consistency to keep the 3D denoising dynamics geometrically anchored.
3. **A 2D-adapted formulation of strong 3D motion regularizers**, enforcing natural motion through velocity-based and representation-level constraints.

2 Related Work

Human Motion Generation in 3D. Generating human motions in 3D is largely driven by deep neural networks. VAEs [2, 13] were early approaches, while diffusion models [54, 6, 20] such as MDM [46] substantially improved fidelity. VQ-VAEs [47] paired with autoregressive [58] and bidirectional autoregressive [14, 35, 17] models have established a new state-of-the-art quality. These approaches typically rely on high-quality 3D motion from MoCap systems, e.g., HumanML3D [13] built from AMASS [31] and A2M [12], which contains approximately 14 thousand motion sequences.

3D Asset Generation with 2D Priors. Generating 3D content from 2D data has been widely explored. Methods leverage strong 2D diffusion priors via score distillation [36, 33, 51, 52] or fine-tune 2D models for novel-view consistency [28, 40, 26, 27, 10, 43] on smaller, curated 3D asset datasets [7]. Other works train generative models directly in 3D [1, 29, 38, 32, 53]. “A Lesson in Splat” [34] formalizes diffusion training for 3D Gaussian Splatting [21] under 2D supervision. A common theme is the reliance on 2D image priors for 3D generation.

2D Pose Extraction from Video. 2D human pose estimation has become highly reliable. OpenPose [5] introduced Part Affinity Fields, AlphaPose [8] improved robustness under occlusion, HRNet [49] preserved high-resolution features, and RTMPose [18] achieves strong real-time accuracy.

Video-to-3D Pose Extraction. Recovering 3D motion from monocular video remains challenging due to depth ambiguity and camera-to-world conversion. WHAM [41] uses a feed-forward model with image features and 2D poses to infer world coordinates, while COIN [23] applies 3D motion diffusion inpainting for iterative refinement. Other approaches [50, 56] exploit SLAM cues [42, 30] to ground their predictions. Complementary lines of work focus on lifting from 2D keypoints: MotionBERT [57] provides a supervised temporal lifting baseline trained on 3D data. Training only on 2D data, ElePose [48] learns a normalizing-flow prior on 2D poses and uses it to steer a lifting model toward better 3D reconstructions. MVLift [25] employs epipolar-constrained 2D diffusion to create pseudo-3D motions to supervise the training of a final multiview 2D model. Recent works [55, 24] combine generation and pose estimation by jointly training multiple 3D motion-related tasks. However, all of these methods still exhibit a non-negligible error gap with respect to ground-truth motions, even within their training datasets.

3D Human Motion Generation from 2D Data. Generating plausible 3D motion supervised directly from 2D sequences remains comparatively underexplored. Existing approaches perform inference-time lifting from 2D priors to 3D: MAS [19] performs multi-view ancestral sampling from several 2D motion diffusion models and omits root trajectory. Motion-2-to-3 [15] trains a 2D diffusion model and then adds consistency layers learned from 3D motion data, relying on 3D supervision to recover root trajectory. These works point to the promise of scaling 2D-centric training while improving world-frame trajectory modeling. In contrast, we train a 3D-native model that learns motions with any root trajectory solely from 2D supervision, without any 3D ground truth.

3 Method

3.1 Preliminaries: Cross-Modality Diffusion for 3D Generation from 2D Supervision

Diffusion models are typically trained under a *same-modality assumption*: both the diffused input and its supervision target belong to the same domain. In 3D generative modeling, this requires large datasets of ground-truth 3D samples, thus significantly limiting scalability. *Lesson in Splats (LIS)* [34] showed that this constraint can be relaxed: by combining approximate 3D estimates with clean 2D supervision, one can train a 3D denoiser without access to any real 3D data.

Specifically, LIS introduces a weak lifter implemented as a deterministic *2D-to-3D predictor*, which serves as a noisy teacher reconstructing approximate 3D Gaussian-splat scenes from single images. For high-noise diffusion timesteps ($t > t^*$), these inaccurate outputs are perturbed with sufficient noise to produce samples statistically aligned with those drawn from the (unknown) clean 3D distribution. The model is trained to denoise these samples while supervision is applied in 2D via differentiable rendering. At low-noise regimes, LIS employs a multi-step denoising scheme: for timesteps $t < t^*$, the input is first further diffused to a higher level $t' > t^*$ and then denoised through a short sequence of DDIM [44] steps down to t , where 2D supervision is applied. This strategy ensures the model experiences training samples at low-noise which are critical for generating high-frequency geometric details.

3.2 Problem Setup and Formulation

We are given a collection of monocular human-motion videos, each captured by a static camera. For each video, we extract 2D joint trajectories $\mathbf{y} \in \mathbb{R}^{J \times 2 \times F}$, where J denotes the number of joints and F the number of frames. Empirically, 2D keypoint extraction is highly accurate and robust [19]. To obtain an approximate 3D signal, we assume access to a pretrained 2D-to-3D lifter L_ϕ that produces approximate 3D joint trajectories $\tilde{x}_0 = L_\phi(\mathbf{y}) \in \mathbb{R}^{J \times 3 \times F}$, serving as a noisy teacher (Section 3.1). Such 2D-to-3D lifting networks are well established in the human-motion literature [57, 48, 41]. When no camera parameters are available they can be estimated by solving PnP on y and \tilde{x}_0 , see Section I.

Our objective is to train a 3D generative diffusion model $D_\theta(x_t, t, \mathbf{c})$ that can sample realistic 3D motion trajectories, optionally conditioned on a text embedding \mathbf{c} , while using only 2D supervision. We denote by $\Pi_c(\cdot) = K[R|\mathbf{t}_c](\cdot)$ the camera projection operator, with intrinsics K , rotation R , and translation \mathbf{t}_c , which maps 3D joint coordinates to 2D. We denote by x 3D motion and by y 2D motions, by \hat{x}_0 the model predictions, by x_t motions diffused to time t and $x^{(f)}$ the motion at frame f .

3.3 Method Overview

The overall framework is illustrated in Fig. 2. As in standard diffusion training we diffuse the lifter’s 3D predictions $x_t = \sqrt{\alpha_t}\tilde{x}_0 + \sqrt{1 - \alpha_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$, and train the network to recover the clean 3D motion \hat{x}_0 [46, 37]. Crucially, while the denoiser operates in 3D, supervision is applied in the 2D domain through the projection operator $\Pi_c(\cdot)$. With appropriate depth-aware weighting, this 2D objective provides a principled surrogate for 3D supervision as introduced in Section 3.4 and proved in Section A. To make 2D supervision effective for 3D motion diffusion, several geometric and temporal adaptations are required.

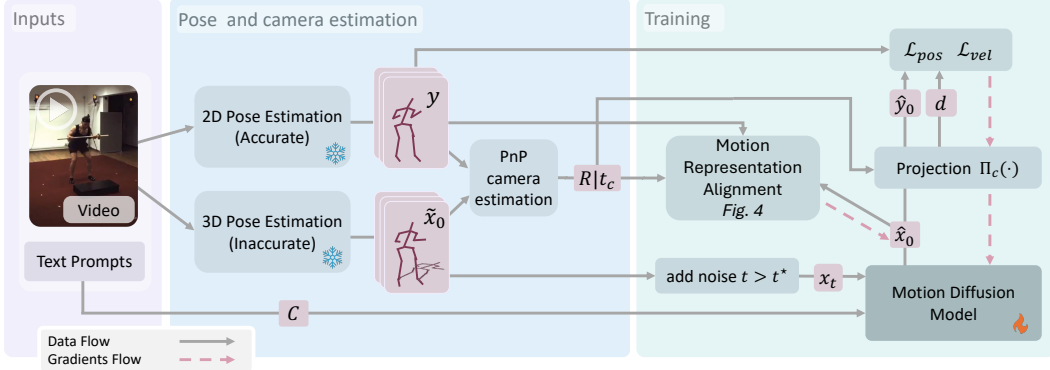


Figure 2: **VideoMDM training.** From monocular video, we extract accurate 2D keypoints and approximate 3D poses. A motion diffusion model is trained to denoise the 3D poses, after diffusing them to a high timestep, under multi-source supervision: (i) 3D representation alignment, and (ii) 2D reprojection and velocity consistency with the accurate 2D pose.

3.4 Depth-Aware Weighting

A naive 2D reprojection loss is not equivalent to 3D MSE: because perspective projection divides by camera depth d , the 2D error is implicitly $1/d$ -weighted relative to the underlying 3D error, downweighting distant joints and overweighting near ones. Multiplying the loss by d removes this scaling, produces a loss equivalent in expectation to direct 3D MSE supervision (see Section A). This equivalence holds under two mild assumptions on the data distribution: that the predicted joint depth d matches the true motion depth in the corresponding camera frame, and that training cameras are sampled with uniformly distributed azimuth.

Let $d \in \mathbb{R}^{J \times 1 \times F}$ denote the predicted depth in the camera coordinate system \hat{x}_0 .

We define:

$$\mathcal{L}_{\text{pos}} = \|d \odot \mathbf{1}_{\{d > d_{\min}\}} \odot (\Pi_c(\hat{x}_0) - y)\|_2^2, \quad (1)$$

where \odot denotes element-wise multiplication across joints and frames. The truncation $\mathbf{1}_{\{d > d_{\min}\}}$ drops joints whose predicted depth falls below d_{\min} , primarily preventing unreliable gradients when joints are predicted behind or very close to the camera, where the projection equations and the equivalence above no longer hold.

3.5 Natural Motion Regularization

2D Velocity Loss. Generative motion networks are commonly regularized using additional geometric losses [46, 39]. We adapt the 3D velocity loss to our 2D setup to enforce temporal similarity between the generated motions and the supervision.

$$\mathcal{L}_{\text{vel}} = \sum_f \|w^{(f)} \odot ((\hat{y}_0^{(f)} - \hat{y}_0^{(f-1)}) - (y^{(f)} - y^{(f-1)}))\|_2^2, \quad (2)$$

where \odot denotes joint-wise multiplication, $w^{(f)} = d^{(f)} \odot \mathbf{1}_{\{d^{(f)} > d_{\min}\}}$ (following Section 3.4), and $\hat{y}_0 = \Pi_c(\hat{x}_0)$.

Motion Representation Alignment. Motion generation models commonly adopt the motion representation of [13], which includes root velocity, joint positions, joint rotations, joint velocities, and binary foot-contact labels. This over-parameterized representation often leads to motions that better match human perceptual judgments. Among these, the rotation, joint-velocity, and foot-contact channels are redundant in the sense that they can be derived directly from the joint positions. We denote these redundant components collectively by \mathbf{r} and the process of deriving them $\Gamma(x)$. As MDM operates on the noised concatenation of x and \mathbf{r} , we partition its $(A_J + B_J)$ -channel output into two components: the first A_J channels represent the predicted root and joint positions \hat{x}_0 (used previously), while the remaining B_J channels constitute the redundant representations \hat{r}_0 (discussed solely here). Further details in Section J.

Since no 3D ground-truth supervision is available for \mathbf{r} , we derive 2D-consistent pseudo-targets by applying the ray-projection operator (illustrated in Figure 3a and formally derived in Section B) to the predicted 3D motion, producing 2D aligned 3D motion, and converting the result to the over-parameterized representation:

$$\mathbf{r}' = \text{stop_gradient}(\Gamma(P_{\Pi}(\hat{x}_0, y))), \quad (3)$$

where Γ calculates the redundant channels from a 3D motion sequence. We then supervise the corresponding denoised outputs, as shown in Figure 3a, using $\mathcal{L}_{\text{repr}} = \|\hat{\mathbf{r}}_0 - \mathbf{r}'\|_2^2$. This provides an indirect 2D-based supervisory signal for the redundant motion channels, helping the model remain consistent with both its own predictions and the available 2D data throughout generation.

3.6 Training Objective and Scheme

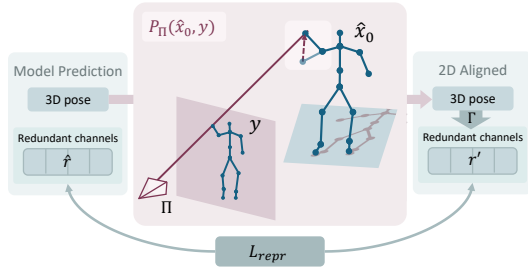
The overall loss combines our proposed terms:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{pos}} \mathcal{L}_{\text{pos}} + \lambda_{\text{vel}} \mathcal{L}_{\text{vel}} + \lambda_{\text{repr}} \mathcal{L}_{\text{repr}} \quad (4)$$

Training begins with a warm up phase of pretraining on approximate 3D motions predicted by the lifter L_{ϕ} , similar to standard MDM training, to initialize the motion prior. We then adopt the LIS-style schedule with threshold t^* . For $t > t^*$, we apply the full loss $\mathcal{L}_{\text{total}}$. For $t \leq t^*$, we further apply multi-step denoising. This staged procedure stabilizes learning and allows the model to progress from lifter-based supervision to fully 2D-derived constraints.

4 Experiments

We evaluate VideoMDM in three complementary settings. Section 4.1 uses a 2D-only version of HumanML3D, isolating the supervision regime from pose-estimation errors. Section 4.2 trains on real monocular video from Fit3D, demonstrating that the learned prior extends to fitness motions far outside the lifter’s distribution. Section 4.3 compares against MAS on the centered NBA dataset on which MAS was designed to operate, providing a head-to-head comparison under conditions favorable to the baseline.



(a) **Motion Representation Alignment.** An illustration of camera ray projection and its utilization for \mathcal{L}_{repr} . Each joint is projected to the closest point along the ray through its 2D location and the camera center.

Implementation Details. For Sections 4.1 and 4.3, we first warm up by pretraining on the lifter’s approximate 3D motions for 400K batches of 64 samples to initialize the motion prior, using the lifted samples of the respective dataset, then train with our full objective for an additional 200K batches. For Section 4.2, since the dataset is small, we initialize from the model trained on synthetic data (Section 4.1) and fine-tune for 20K batches. We select λ_{vel} , t^* , and the number of denoising steps below t^* by Bayesian search over the validation FID on Section 4.1 using MVLift [25] as the lifter; the resulting values (Section F) are reused unchanged across all models for the synthetic and Fit3D experiments. Validation FID stayed within 0.7–2 across most searched configurations, indicating low sensitivity to these choices. For the NBA experiment, we perform the same search using a random strategy over the training set FID (as no validation split is available). FID spans 7-11 across all configurations tested.

4.1 Text-to-3D Motion from 2D Poses

HumanML3D [13] consists of 14,616 motion sequences sourced from AMASS [31] and HumanAct12 [12], paired with 44,970 textual descriptions, standardized to 20 FPS and capped at 10 seconds. We construct a 2D-only version of HumanML3D by sampling random cameras, rendering the 2D motions and lifting them by either MotionBERT [57] or MVLift [25]. For evaluation in 3D we follow T2M [13], using MDM [46] inference and evaluation code. Further details in Section C.

Compared Methods. We compare three classes of methods, distinguished by the supervision they require. *3D-supervised upper bound:* MDM [46] trained on the original 3D motions. *2D-supervised baselines:* MAS [19], trained as a text-conditioned, non-centered variant; and MDM trained on lifter outputs (MotionBERT or MVLift) treated as 3D ground truth — these baselines use no camera information. *Our method:* three variants that supervise in 2D via reprojection and therefore require camera parameters. *Ours/MotionBERT* and *Ours/MVLift* use ground-truth cameras with their corresponding lifters as teachers. *Ours/MVLift (PnP)* estimates the camera via PnP from the MVLift teacher (see Section I), matching the information available to the 2D-supervised baselines.

Results. As shown in Table 1, under matched supervision *Ours/MVLift (PnP)* reduces FID by 0.21 against *MDM/MVLift*, demonstrating the value of 2D-reprojection training over training MDM directly on lifted 3D. With GT cameras the advantage grows: *Ours/MVLift* reaches FID 0.88, nearly closing the gap to the 3D-supervised upper bound (0.54), empirically supporting the loss-equivalence claim of Section 3.4, highlighting that the remaining gap is attributed to the camera estimation error. With improved camera estimation methods, our method directly benefits.

Figure 4 confirms the quantitative picture: *MDM/MotionBERT* inherits lifter artifacts and exhibits sliding, *MDM/MVLift* is better but still lacks coherence, while *VideoMDM* produces clean trajectories that accurately follow the prompt. Additional results are available in the supplementary material.

4.2 Real-Video Training: Beyond the Lifter Distribution

Our central claim is that 2D supervision from monocular video unlocks 3D motion distributions inaccessible to MoCap-bound training. We test this on Fit3D [9]: real videos using 2D-pose extraction, and motions far outside the distribution captured by the lifters. Fit3D contains 611 training sequences

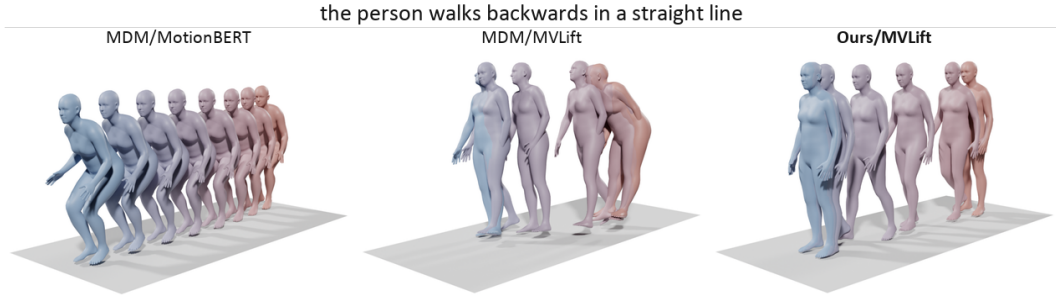


Figure 4: Qualitative comparison on HumanML3D for the prompt “the person walks backwards in a straight line”. Frames progress from blue to red. The MDM-trained-on-lifter baselines inherit teacher artifacts (foot sliding, drifting, unrealistic poses), while *VideoMDM* generates a clean trajectory consistent with the prompt.

Method	FID↓	Diversity→	R Precision (top 3)↑	Multimodal Dist↓	Multimodality↑
Ground-truth	.0016±.000	9.459±.052	0.796±.002	2.975±.009	—
MDM (3D data)	0.544±.044	9.559±.086	0.611±.007	5.566±.027	2.799±.072
MAS	22.056±.009	6.236±.051	0.383±.000	6.416±.000	—
MDM/MotionBERT	5.660±.156	8.198±.048	0.666±.006	4.112±.016	2.453±.110
MDM/MVLift	1.671±.094	8.793±.048	0.719±.005	3.514±.018	2.375±.105
Ours/MVLift (PnP)	1.462±.097	9.130±.052	0.714±.007	3.527±.031	2.692±.087
Ours/MotionBERT	1.454±.090	9.533±.060	0.681±.007	3.677±.030	2.457±.050
Ours/MVLift	0.876±.090	9.630±.068	0.721±.005	3.450±.028	2.449±.110

Table 1: Text-to-motion models trained in 2D, evaluated on the HumanML3D test split. **Red**, **orange**, and **yellow** indicate 1st, 2nd, and 3rd place per column among 2D-supervised methods. X/Y denotes a method X trained on lifter Y . *Ours* variants use ground-truth cameras unless marked (PnP). Remarkably, *Ours/MVLift* achieves performance only 0.332 FID away from 3D-supervised MDM.

across 37 fitness exercises, with synchronized video and accurate 3D ground truth used only for evaluation. Many of these motions have no analog in HumanML3D (e.g. mule kicks, burpees, stretches). For each sequence we randomly select one camera, extract 2D poses with RTMPose [18]. Approximate 3D poses are obtained from the same video using WHAM [41], a dedicated video-to-3D method, serving as the noisy teacher. Camera positions are provided in the dataset or estimated using PnP. We initialize VideoMDM from Ours/MVLift trained in Section 4.1 and fine-tune for 20K batches. Further details in Section C.

Evaluation. Fit3D is too small for reliable generative metrics (FID). We instead evaluate VideoMDM in two ways: as a 2D-to-3D lifter Table 2, and with a human survey Figure 5a. We repurpose VideoMDM as a lifter via inference-time guidance: at every denoising step, we project the predicted clean motion onto the camera rays of the observed 2D keypoints, $\hat{x}_0 \leftarrow P_{\Pi}(\hat{x}_0, y)$ (Section G); this provides a direct probe of prior quality against corresponding ground-truth 3D motions.

Metrics. We report Mean Per-Joint Position Error (MPJPE) in mm, Procrustes-aligned MPJPE (PA-MPJPE) in mm using per-frame scale, rotation, and translation alignment, Percentage of Correct Keypoints (PCK) at 50 mm and 100 mm thresholds, and acceleration error (m/s^2) from second-order finite differences. For generative quality we report KID [3] – which is considered more reliable for smaller sample size – in the HumanML3D evaluation VAE [13], using subsampling with replacement (100 trials) for stable means and standard deviations.

Compared Methods. On 2D-to-3D motion lifting, we compare against WHAM [41], *MVLift* [25], and MDM trained on HumanML3D guided in the same way as ours. For human preference we compare against generative text-to-3D methods: MDM/WHAM and MDM/MVLift and on the lifted motions themselves.

Results – motion lifting. As shown in Table 2, *Ours* achieves the best results across most metrics: MPJPE drops by a factor of 2 and 4 compared to WHAM and MDM, respectively. *Ours (PnP)* retains most of this gap. Crucially, both our variants excel in Accel and KID, indicating smooth motions that

Model	MPJPE (mm)	PA-MPJPE (mm)	PCK@ 50mm(%)	PCK@ 100mm(%)	Accel (m/s ²)	KID (\pm std)
MDM	440.33	114.06	6.34	24.20	7.71	0.050 \pm 0.010
WHAM	228.47	51.12	2.89	17.39	17.66	0.063 \pm 0.008
MVLift	283.06	94.45	5.84	22.99	3.14	0.028 \pm 0.006
Ours/WHAM (PnP)	185.81	74.03	15.49	52.82	3.04	0.013 \pm 0.003
Ours/WHAM	111.24	61.69	22.26	62.80	3.16	0.011 \pm 0.003

Table 2: Lifting evaluation on Fit3D held out subject. Our method achieves best KID and MPJPE both on GT and PnP cameras.

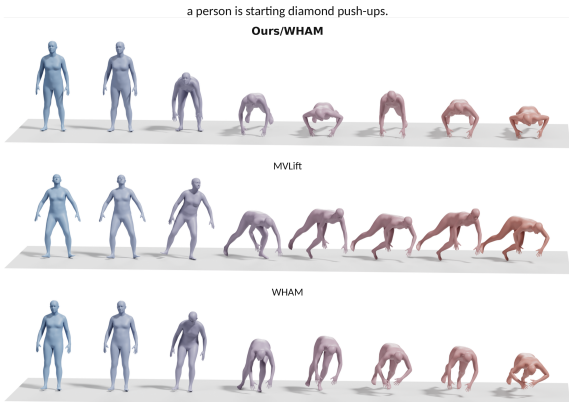
are statistically aligned with the true 3D motion distribution. WHAM retains an edge on PA-MPJPE, which removes scale and rotation and rewards local pose accuracy — consistent with WHAM’s pose-supervised objective and its use of image features, which carry local detail unavailable in the 2D skeleton alone.

Results – text-to-motion. For reliable evaluation of conditional 3D motion in the absence of large scale data, we turn to human inspection. Figure 5a reports the percentage of human preferred motions between each baseline and ours (e.g. 40.0% on MDM/WHAM means ours was preferred 60.0% of the time). While Ours/WHAM consistently outperforms all baselines, Ours/WHAM (PnP) loses to the WHAM baselines. Inspecting the down-voted motions, we observed these suffered from poor alignment with the ground, which was visually unsatisfying. We hypothesize that a ground-aware inference or post-processing could significantly boost visual appearance, yet we left the raw results for full transparency.

Method	Ours/WHAM pref.	Ours/WHAM (PnP) pref.
MDM/WHAM	40.0%	62.1%
WHAM	38.7%	60.3%
MVLift	25.0%	42.3%
MDM/MVLift	11.3%	12.5%

(a) Human Preference Survey results for the Fit3D test set text prompts. MDM-based models (Ours, MDM/WHAM, and MDM/MVLift) receive only text prompts; WHAM and MVLift also receive the original video.

Figure 6a shows an example of a Fit3D exercise prompt. Ours/WHAM produces a coherent and text-aligned motion, while baselines suffer from misalignment to this motion which is outside their train set. Additional video examples are in the supplementary material.



(a) Results of generation by VideoMDM trained on Fit3D against the lifter baselines, the lifters having access to the 2D data while Ours only to the text prompt.

w/o Variant	FID ↓	Diversity →	R-Prec ↑	MM-Dist ↓
distance weighting	1.27	8.84	0.72	3.48
multistep for $t < t^*$	9.85	7.15	0.45	5.60
\mathcal{L}_{vel}	1.58	9.19	0.69	3.62
\mathcal{L}_{repr}	5.75	8.25	0.60	4.42
ray proj for \mathcal{L}_{repr}	2.72	8.91	0.63	4.09
Ours	1.05	9.60	0.71	3.50

(b) Ablations of our method using PnP cameras and MVLift as a lifter, mean values over 5 replications on the HumanML3D validation set.

4.3 Unconditional 3D Generation from NBA videos

The NBA dataset, released with MAS [19], provides single-view basketball sequences with 16-joint AlphaPose [8] detections. Since MAS was designed for centered motion generation, the dataset was

Method	Human \uparrow Pref.	FID \downarrow	Diversity \rightarrow	Precision \uparrow	Recall \uparrow	Recall \uparrow \uparrow
Training Data	—	1.05 \pm .02	8.97 \pm .05	0.73 \pm .01	0.73 \pm .01	0.86 \pm .01
ElePose	—	10.76 \pm .45	9.72 \pm .05	0.28 \pm .02	0.58 \pm .03	0.45 \pm .01
MotionBERT	—	30.22 \pm .26	9.57 \pm .09	0.04 \pm .00	0.34 \pm .04	0.04 \pm .01
MAS	36.0%	5.38\pm.06	9.47\pm.06	0.50 \pm .01	0.60\pm.01	0.68 \pm .00
Ours/ElePose	64.0%	7.18 \pm .09	7.93 \pm .04	0.94\pm.00	0.10 \pm .00	0.89\pm.01

Table 3: Evaluation on the NBA dataset. Human preference is reported between MAS and our VideoMDM. Other metrics follow MAS [19]. VideoMDM is preferred nearly two thirds of votes.

collected accordingly: all motions are centered per frame (root at the origin), and no text prompts are provided. The release also includes lifted centered motions from MotionBERT [57] and ElePose [48]. We use the latter as our noisy teacher. Further technical details are in Section C.

Methods and Results. We compare against MAS and report results in Table 3. The standard evaluation protocol, released with MAS, computes metrics in the embedding of a 2D VAE trained on the same data. Inspecting the VAE’s encode–decode reconstructions reveals lower fidelity than the corresponding HumanML3D evaluation VAE, suggesting the VAE-based metrics here are a less reliable indicator of perceptual quality. We therefore conducted a human preference study (Section H) in which participants compared our VideoMDM and MAS outputs in pairwise settings. Across 200 votes, VideoMDM was preferred in nearly two thirds, aligning with our own visual inspection of the generations. Generated motion examples for both models are included in the supplementary material.

For completeness, we also report the standard MAS protocol metrics. We further note an issue with how Recall is computed in this protocol: for each real sample, it asks whether some generated sample lies within a threshold defined by the *generated*-distribution spread, biasing the metric toward exaggerated diversity. We additionally report Recall \uparrow (Section E), which anchors the threshold to the real-distribution spread instead and better reflects whether the generated motions cover the real distribution. VideoMDM falls slightly behind MAS on FID and Diversity, but achieves substantial gains in Precision and the best Recall \uparrow , indicating that our generations spread less diversely, but otherwise similarly to the real distribution in the VAE embedding.

4.4 Ablations

We ablate each component of VideoMDM on the HumanML3D validation set, using PnP-estimated cameras with MVLift as the teacher. Each row in Figure 6b reports the mean over 5 replicate runs with one component removed: *distance weighting* drops the \mathcal{L}_{pos} loss defined in (Section 3.4); *multistep for $t < t^*$* disables multistep denoising entirely (i.e. $t^* = 0$) and trains using the noised lifter prediction directly across all t [34]; \mathcal{L}_{vel} and $\mathcal{L}_{\text{repr}}$ drop the corresponding losses by setting $\lambda_{\text{vel}} = 0$ or $\lambda_{\text{repr}} = 0$; and *ray proj* replaces the camera-ray projection used in $\mathcal{L}_{\text{repr}}$ with a direct comparison $\mathcal{L}_{\text{repr}} = \|\hat{r} - \Gamma(\hat{x}_0)\|$. Multistep denoising and $\mathcal{L}_{\text{repr}}$ are the two most impactful components. Replacing the ray projection in $\mathcal{L}_{\text{repr}}$ with a naive 3D comparison roughly triples FID, confirming that our 2D-consistent pseudo-targets are essential to the representation alignment. \mathcal{L}_{vel} and distance weighting contribute smaller but non-negligible improvements on FID, with negligible differences on the remaining metrics.

5 Conclusion, Limitations and Future Work

We have presented VideoMDM, a training method for 3D human motion diffusion using only 2D supervision, and demonstrated its ability to generate high-quality motions that in some settings nearly match the performance of fully 3D-supervised methods. Our cross-modality diffusion owes its success to a set of stabilization techniques and natural motion regularizations formulated in 2D. VideoMDM makes a significant stride towards training motion diffusion models directly from abundant monocular videos. Such capability opens up new possibilities for learning generative priors of real-world motion distributions such as multi-person behaviors and human-object interactions which are otherwise difficult to acquire.

Limitations. VideoMDM achieves its strongest results with ground-truth camera parameters, which are unavailable for most in-the-wild videos. PnP-estimated cameras recover most of the gap on synthetic HumanML3D but leave a larger drop on Fit3D, where camera estimation noise compounds with lifter and pose-extraction noise; better camera estimators would directly translate into better

priors. Our method also depends on a pretrained 2D-to-3D lifter as a noisy teacher: although the learned prior generalizes substantially beyond the lifter’s distribution (Section 4.2), domains where no reasonable lifter is available (e.g. non-human motion) remain out of reach. All settings we evaluate contain no or only minimal occlusions; extending to occluded settings such as those in [11] is necessary for fully in-the-wild deployment.

Acknowledgments and Disclosure of Funding

Or Litany acknowledges support from the Israel Science Foundation (grant 624/25) and the Azrieli Foundation Early Career Faculty Fellowship. This research was also supported in part by an academic gift from Meta. The authors gratefully acknowledge this support. This research was supported by the Council for Higher Education in Israel under the Moonshot Project.

References

- [1] A. Alliegro, Y. Siddiqui, T. Tommasi, and M. Nießner. Polydiff: Generating 3d polygonal meshes with diffusion models, 2023. URL <https://arxiv.org/abs/2312.11417>.
- [2] X. Bie, W. Guo, S. Leglaive, L. Girin, F. Moreno-Noguer, and X. Alameda-Pineda. Hit-dvae: Human motion generation via hierarchical transformer dynamical vae, 2022. URL <https://arxiv.org/abs/2204.01565>.
- [3] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=r11U0zWCW>.
- [4] G. Bradski. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*, 2000.
- [5] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields, 2019. URL <https://arxiv.org/abs/1812.08008>.
- [6] 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.
- [7] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, E. VanderBilt, A. Kembhavi, C. Vondrick, G. Gkioxari, K. Ehsani, L. Schmidt, and A. Farhadi. Objaverse-xl: a universe of 10m+ 3d objects. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS ’23*, Red Hook, NY, USA, 2023. Curran Associates Inc.
- [8] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(6):7157–7173, June 2023. ISSN 0162-8828. doi: 10.1109/TPAMI.2022.3222784. URL <https://doi.org/10.1109/TPAMI.2022.3222784>.
- [9] M. Fieraru, M. Zanfir, S.-C. Pirlea, V. Olaru, and C. Sminchisescu. Aifit: Automatic 3d human-interpretable feedback models for fitness training. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.
- [10] R. Gao, A. Holynski, P. Henzler, A. Brussee, R. Martin-Brualla, P. Srinivasan, J. T. Barron, and B. Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024.
- [11] K. Grauman, A. Westbury, L. Torresani, K. Kitani, J. Malik, T. Afouras, K. Ashutosh, V. Baiyya, S. Bansal, B. Boote, E. Byrne, Z. Chavis, J. Chen, F. Cheng, F.-J. Chu, S. Crane, A. Dasgupta, J. Dong, M. Escobar, C. Forigua, A. Gebreselasie, S. Haresh, J. Huang, M. M. Islam, S. Jain, R. Khirodkar, D. Kukreja, K. J. Liang, J.-W. Liu, S. Majumder, Y. Mao, M. Martin, E. Mavroudi, T. Nagarajan, F. Ragusa, S. K. Ramakrishnan, L. Seminara, A. Somayazulu, Y. Song, S. Su, Z. Xue, E. Zhang, J. Zhang, A. Castillo, C. Chen, X. Fu, R. Furuta, C. Gonzalez, P. Gupta, J. Hu, Y. Huang, Y. Huang, W. Khoo, A. Kumar, R. Kuo, S. Lakhavani, M. Liu, M. Luo, Z. Luo,

- B. Meredith, A. Miller, O. Oguntola, X. Pan, P. Peng, S. Pramanick, M. Ramazanov, F. Ryan, W. Shan, K. Somasundaram, C. Song, A. Southerland, M. Tateno, H. Wang, Y. Wang, T. Yagi, M. Yan, X. Yang, Z. Yu, S. C. Zha, C. Zhao, Z. Zhao, Z. Zhu, J. Zhuo, P. Arbelaez, G. Bertasius, D. Crandall, D. Damen, J. Engel, G. M. Farinella, A. Furnari, B. Ghanem, J. Hoffman, C. V. Jawahar, R. Newcombe, H. S. Park, J. M. Rehg, Y. Sato, M. Savva, J. Shi, M. Z. Shou, and M. Wray. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives, 2024. URL <https://arxiv.org/abs/2311.18259>.
- [12] C. Guo, X. Zuo, S. Wang, S. Zou, Q. Sun, A. Deng, M. Gong, and L. Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020.
- [13] 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 (CVPR)*, pages 5152–5161, 2022.
- [14] C. Guo, Y. Mu, M. G. Javed, S. Wang, and L. Cheng. Momask: Generative masked modeling of 3d human motions, 2023. URL <https://arxiv.org/abs/2312.00063>.
- [15] R. Guo, H. Pi, Z. Shen, Q. Shuai, Z. Hu, Z. Wang, Y. Dong, R. Hu, T. Komura, S. Peng, and X. Zhou. Motion-2-to-3: Leveraging 2d motion data for 3d motion generations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14305–14316, October 2025.
- [16] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models, 2020. URL <https://arxiv.org/abs/2006.11239>.
- [17] S.-E. Hong, S. Lim, J. Hwang, M. Chang, and H. Kang. Bipo: Bidirectional partial occlusion network for text-to-motion synthesis, 2025. URL <https://arxiv.org/abs/2412.00112>.
- [18] T. Jiang, P. Lu, L. Zhang, N. Ma, R. Han, C. Lyu, Y. Li, and K. Chen. Rtmpose: Real-time multi-person pose estimation based on mmpose, 2023. URL <https://arxiv.org/abs/2303.07399>.
- [19] R. Kapon, G. Tevet, D. Cohen-Or, and A. H. Bermano. Mas: Multi-view ancestral sampling for 3d motion generation using 2d diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1965–1974, 2024.
- [20] K. Karunratanakul, K. Preechakul, S. Suwajanakorn, and S. Tang. Guided motion diffusion for controllable human motion synthesis, 2023. URL <https://arxiv.org/abs/2305.12577>.
- [21] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), July 2023. ISSN 0730-0301. doi: 10.1145/3592433. URL <https://doi.org/10.1145/3592433>.
- [22] T. Kynkäänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila. Improved precision and recall metric for assessing generative models. *CoRR*, abs/1904.06991, 2019.
- [23] J. Li, Y. Yuan, D. Rempe, H. Zhang, C. Lu, J. Kautz, and U. Iqbal. Coin: Control-inpainting diffusion prior for human and camera motion estimation. In *European Conference on Computer Vision (ECCV)*, 2024.
- [24] J. Li, J. Cao, H. Zhang, D. Rempe, J. Kautz, U. Iqbal, and Y. Yuan. Genmo: Generative models for human motion synthesis. *arXiv preprint arXiv:2505.01425*, 2025.
- [25] J. Li, C. K. Liu, and J. Wu. Lifting motion to the 3d world via 2d diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17518–17528, 2025.
- [26] M. Liu, C. Xu, H. Jin, L. Chen, M. Varma T, Z. Xu, and H. Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36:22226–22246, 2023.

- [27] M. Liu, R. Shi, L. Chen, Z. Zhang, C. Xu, X. Wei, H. Chen, C. Zeng, J. Gu, and H. Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10072–10083, 2024.
- [28] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023.
- [29] Z. Liu, Y. Feng, M. J. Black, D. Nowrouzezahrai, L. Paull, and W. Liu. Meshdiffusion: Score-based generative 3d mesh modeling, 2023. URL <https://arxiv.org/abs/2303.08133>.
- [30] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel. A comprehensive survey of visual slam algorithms. *Robotics*, 11(1), 2022. ISSN 2218-6581. doi: 10.3390/robotics11010024. URL <https://www.mdpi.com/2218-6581/11/1/24>.
- [31] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes, 2019. URL <https://arxiv.org/abs/1904.03278>.
- [32] Y. Mu, X. Zuo, C. Guo, Y. Wang, J. Lu, X. Wu, S. Xu, P. Dai, Y. Yan, and L. Cheng. Gsd: View-guided gaussian splatting diffusion for 3d reconstruction, 2024. URL <https://arxiv.org/abs/2407.04237>.
- [33] H. Nam, G. Kwon, G. Y. Park, and J. C. Ye. Contrastive denoising score for text-guided latent diffusion image editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9192–9201, 2024.
- [34] C. Peng, I. Sobol, M. Tomizuka, K. Keutzer, C. Xu, and O. Litany. A lesson in splats: Teacher-guided diffusion for 3d gaussian splats generation with 2d supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.
- [35] E. Pinyoanuntapong, M. U. Saleem, P. Wang, M. Lee, S. Das, and C. Chen. Bamm: Bidirectional autoregressive motion model, 2024. URL <https://arxiv.org/abs/2403.19435>.
- [36] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- [37] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents, 2022. URL <https://arxiv.org/abs/2204.06125>.
- [38] B. Roessle, N. Müller, L. Porzi, S. Rota Bulò, P. Kotschieder, A. Dai, and M. Nießner. L3dg: Latent 3d gaussian diffusion. In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400711312. doi: 10.1145/3680528.3687699. URL <https://doi.org/10.1145/3680528.3687699>.
- [39] M. Shi, K. Aberman, A. Aristidou, T. Komura, D. Lischinski, D. Cohen-Or, and B. Chen. Motionet: 3d human motion reconstruction from monocular video with skeleton consistency. *ACM Transactions on Graphics*, 40(1):1–15, Sept. 2020. ISSN 1557-7368. doi: 10.1145/3407659. URL <http://dx.doi.org/10.1145/3407659>.
- [40] R. Shi, H. Chen, Z. Zhang, M. Liu, C. Xu, X. Wei, L. Chen, C. Zeng, and H. Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023.
- [41] S. Shin, J. Kim, E. Halilaj, and M. J. Black. WHAM: Reconstructing world-grounded humans with accurate 3D motion. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2024.
- [42] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, 5(4):56–68, 1986. doi: 10.1177/027836498600500404. URL <https://doi.org/10.1177/027836498600500404>.
- [43] I. Sobol, C. Xu, and O. Litany. Zero-to-hero: Enhancing zero-shot novel view synthesis via attention map filtering, 2024. URL <https://arxiv.org/abs/2405.18677>.

- [44] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [45] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations, 2021. URL <https://arxiv.org/abs/2011.13456>.
- [46] 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>.
- [47] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, page 6309–6318, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.
- [48] B. Wandt, J. J. Little, and H. Rhodin. ElePose: Unsupervised 3D Human Pose Estimation by Predicting Camera Elevation and Learning Normalizing Flows on 2D Poses . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6625–6635, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.00652. URL <https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.00652>.
- [49] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep high-resolution representation learning for visual recognition, 2020. URL <https://arxiv.org/abs/1908.07919>.
- [50] Y. Wang, Z. Wang, L. Liu, and K. Daniilidis. Tram: Global trajectory and motion of 3d humans from in-the-wild videos. In *European Conference on Computer Vision*, pages 467–487. Springer, 2024.
- [51] Z. Wang, C. Lu, Y. Wang, F. Bao, C. LI, H. Su, and J. Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 8406–8441. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1a87980b9853e84dfb295855b425c262-Paper-Conference.pdf.
- [52] K. Xie, J. Lorraine, T. Cao, J. Gao, J. Lucas, A. Torralba, S. Fidler, and X. Zeng. Latte3d: Large-scale amortized text-to-enhanced3d synthesis. In *European Conference on Computer Vision*, pages 305–322. Springer, 2024.
- [53] X. Zeng, A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, and K. Kreis. Lion: Latent point diffusion models for 3d shape generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [54] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu. Motiondiffuse: Text-driven human motion generation with diffusion model, 2022. URL <https://arxiv.org/abs/2208.15001>.
- [55] M. Zhang, D. Jin, C. Gu, F. Hong, Z. Cai, J. Huang, C. Zhang, X. Guo, L. Yang, Y. He, and Z. Liu. Large motion model for unified multi-modal motion generation. *arXiv preprint arXiv:2404.01284*, 2024.
- [56] S. Zhang, B. L. Bhatnagar, Y. Xu, A. Winkler, P. Kadlecik, S. Tang, and F. Bogo. Rohm: Robust human motion reconstruction via diffusion, 2024. URL <https://arxiv.org/abs/2401.08570>.
- [57] W. Zhu, X. Ma, Z. Liu, L. Liu, W. Wu, and Y. Wang. Motionbert: A unified perspective on learning human motion representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.
- [58] Q. Zou, S. Yuan, S. Du, Y. Wang, C. Liu, Y. Xu, J. Chen, and X. Ji. Parco: Part-coordinating text-to-motion synthesis, 2024. URL <https://arxiv.org/abs/2403.18512>.

A Weights for 3D to 2D Loss Equivalence

In standard DDPM and DDIM training, given a sample $\mathbf{x} \sim p$ and denoiser output $\hat{\mathbf{x}}$, the reconstruction loss is the mean squared error:

$$\mathbb{L}_{3d} = \mathbb{E}_{\mathbf{x} \sim p} [\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2].$$

Since the loss decomposes over coordinates, we ignore any additional structure (e.g., joint J or frame F) and focus on a single 3D point:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \hat{\mathbf{x}} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix}.$$

For 2D projection, let ψ denote the camera elevation angle and θ the azimuth angle. We denote by $\mathcal{P}(\mathbf{x}, \psi, \theta)$ the perspective projection of \mathbf{x} onto the camera image plane:

$$\mathcal{P}(\mathbf{x}, \psi, \theta) = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{\cos \theta \cos \psi x + \sin \theta \cos \psi z}{d(\mathbf{x}, \theta, \psi)} \\ \frac{\cos \psi y + \cos \theta \sin \psi x + \sin \theta \sin \psi z}{d(\mathbf{x}, \theta, \psi)} \end{bmatrix},$$

with

$$d(\mathbf{x}, \theta, \psi) = \sin \psi y + \cos \theta \cos \psi x + \sin \theta \cos \psi z.$$

We define the 2D MSE loss as

$$\mathbb{L}_{2d} = \mathbb{E}_{\mathbf{x} \sim p, \theta \sim \mathcal{U}[0, 2\pi]} [\|W \odot (\mathcal{P}(\hat{\mathbf{x}}, \psi, \theta) - \mathcal{P}(\mathbf{x}, \psi, \theta))\|_2^2],$$

where $W = [W_u, W_v]^\top$ contains per-axis weights and \odot denotes element-wise multiplication.

Depth assumption. We approximate the projection denominator as constant between \mathbf{x} and $\hat{\mathbf{x}}$:

$$d \triangleq d(\mathbf{x}, \theta, \psi) = d(\hat{\mathbf{x}}, \theta, \psi)$$

We now show that there exist weights W_u, W_v such that $\mathbb{L}_{2d} = \mathbb{L}_{3d}$, and that they are both proportional to d , the depth of the point in camera coordinates.

Proof.

Consider the normalization weights

$$W_u = \frac{d}{\Phi}, \quad W_v = \frac{d}{\cos \psi}, \quad \Phi = \frac{\cos \psi}{\sqrt{2 - \tan^2 \psi}}.$$

Image v -axis contribution.

$$\begin{aligned} & \frac{1}{2\pi} \int_0^{2\pi} \left(\frac{d}{\cos \psi} \left(\frac{\cos \psi y + \cos \theta \sin \psi x + \sin \theta \sin \psi z}{d} - \frac{\cos \psi \hat{y} + \cos \theta \sin \psi \hat{x} + \sin \theta \sin \psi \hat{z}}{d} \right) \right)^2 d\theta \\ &= \frac{1}{2\pi} \int_0^{2\pi} (y + \cos \theta \tan \psi x + \sin \theta \tan \psi z - \hat{y} - \cos \theta \tan \psi \hat{x} - \sin \theta \tan \psi \hat{z})^2 d\theta \\ &= \frac{1}{2\pi} \int_0^{2\pi} \left(2 \cos \theta \tan \psi (yx - \hat{y}x - y\hat{x} + \hat{y}\hat{x}) + 2 \sin \theta \tan \psi (yz - \hat{y}z - y\hat{z} + \hat{y}\hat{z}) \right. \\ &\quad \left. + 2 \sin \theta \cos \theta \tan^2 \psi (xz - \hat{x}z - x\hat{z} + \hat{x}\hat{z}) + (y^2 - 2y\hat{y} + \hat{y}^2) \right. \\ &\quad \left. + \cos^2 \theta \tan^2 \psi (x^2 - 2x\hat{x} + \hat{x}^2) + \sin^2 \theta \tan^2 \psi (z^2 - 2z\hat{z} + \hat{z}^2) \right) d\theta \\ &\stackrel{(1)}{=} (y - \hat{y})^2 + \frac{1}{2} \tan^2 \psi ((x - \hat{x})^2 + (z - \hat{z})^2), \end{aligned}$$

where step (1) uses linearity of integration and the identities

$$\begin{aligned} \int_0^{2\pi} \cos \theta d\theta &= \int_0^{2\pi} \sin \theta d\theta = \int_0^{2\pi} \cos \theta \sin \theta d\theta = 0, \\ \int_0^{2\pi} \cos^2 \theta d\theta &= \int_0^{2\pi} \sin^2 \theta d\theta = \pi. \end{aligned}$$

Image u -axis contribution.

$$\begin{aligned}
& \frac{1}{2\pi} \int_0^{2\pi} \left(\frac{d}{\Phi} \left(\frac{\cos \theta \cos \psi x + \sin \theta \cos \psi z}{d} - \frac{\cos \theta \cos \psi \hat{x} + \sin \theta \cos \psi \hat{z}}{d} \right) \right)^2 d\theta \\
&= \frac{1}{2\pi} \int_0^{2\pi} \left(\cos \theta \frac{\cos \psi}{\Phi} (x - \hat{x}) + \sin \theta \frac{\cos \psi}{\Phi} (z - \hat{z}) \right)^2 d\theta \\
&= \frac{\cos^2 \psi}{2\pi \Phi^2} \int_0^{2\pi} \left(\cos^2 \theta (x - \hat{x})^2 + 2 \sin \theta \cos \theta (x - \hat{x})(z - \hat{z}) + \sin^2 \theta (z - \hat{z})^2 \right) d\theta \\
&\stackrel{(1)}{=} \frac{\cos^2 \psi}{2\Phi^2} ((x - \hat{x})^2 + (z - \hat{z})^2).
\end{aligned}$$

Summing the coefficients on $(x - \hat{x})^2 + (z - \hat{z})^2$ from the u and v contributions:

$$\frac{1}{2} \tan^2 \psi + \frac{\cos^2 \psi}{2\Phi^2} = \frac{1}{2} \tan^2 \psi + \frac{\cos^2 \psi}{2 \left(\frac{\cos \psi}{\sqrt{2 - \tan^2 \psi}} \right)^2} = \frac{1}{2} \tan^2 \psi + \frac{2 - \tan^2 \psi}{2} = 1.$$

Thus, for every sample,

$$\mathbb{E}_\theta \left[\|W \odot (\mathcal{P}(\hat{\mathbf{x}}, \psi, \theta) - \mathcal{P}(\mathbf{x}, \psi, \theta))\|_2^2 \right] = \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2,$$

and therefore $\mathbb{L}_{2d} = \mathbb{L}_{3d}$.

B Projection of a 3D Point onto a 2D Camera Ray

We consider a calibrated pinhole camera Π with extrinsics $[\mathbf{R} | \mathbf{t}] \in \mathbb{R}^{3 \times 4}$ that map world points to the camera frame as

$$\mathbf{x}_{\text{cam}} = \mathbf{R} \mathbf{x}_{\text{world}} + \mathbf{t}, \quad \mathbf{R} \in \text{SO}(3), \mathbf{t} \in \mathbb{R}^3.$$

Let $y = (u, v)^T \in \mathbb{R}^2$ denote image coordinates in normalized units (after applying the inverse of the intrinsics, K^{-1}), and define the camera-frame ray direction

$$\mathbf{y}_{\mathbf{H}} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \in \mathbb{R}^3.$$

World ray. The camera center in world coordinates and the corresponding world-frame ray direction are

$$\mathbf{C} = -\mathbf{R}^\top \mathbf{t}, \quad \mathbf{r} = \mathbf{R}^\top \mathbf{y}_{\mathbf{H}}. \quad (5)$$

Thus, the camera ray in world coordinates is the line

$$\ell(\lambda) = \mathbf{C} + \lambda \mathbf{r}, \quad \lambda \in \mathbb{R}.$$

Orthogonal projection onto the ray. Given a world point $\mathbf{x} \in \mathbb{R}^3$, its closest point \mathbf{P}_Π on the ray ℓ is obtained by minimizing $\|\mathbf{C} + \lambda \mathbf{r} - \mathbf{x}\|_2^2$ with respect to λ . The optimal scalar is

$$\lambda^* = \frac{\mathbf{r}^\top (\mathbf{x} - \mathbf{C})}{\mathbf{r}^\top \mathbf{r}} = \frac{\mathbf{r}^\top \mathbf{x} + \mathbf{y}_{\mathbf{H}}^\top \mathbf{t}}{\mathbf{y}_{\mathbf{H}}^\top \mathbf{y}_{\mathbf{H}}}, \quad (6)$$

where the second equality uses $\mathbf{C} = -\mathbf{R}^\top \mathbf{t}$ and the orthogonality of \mathbf{R} : $\mathbf{r}^\top \mathbf{r} = (\mathbf{R}^\top \mathbf{d})^\top (\mathbf{R}^\top \mathbf{d}) = \mathbf{d}^\top \mathbf{d}$. The projected point is then

$$\mathbf{P}_\Pi(\mathbf{x}, \mathbf{y}) = \mathbf{C} + \lambda^* \mathbf{r}. \quad (7)$$

In the paper, we use $P_\Pi(\hat{\mathbf{x}}_0, \mathbf{y})$ to denote the above operation, applied separately per frame and per joint.

C Experiments Technical Details

C.1 HumanML3D Data Processing

We construct a 2D-only version of HumanML3D by sampling, for each motion, a camera with azimuth $\sim \mathcal{U}[-\pi, \pi]$, elevation $\sim \mathcal{U}[0, \pi/8]$, and position constrained so that the closest joint is at least 3 units from the camera. We project the 3D motion to 2D and discard the 3D ground truth from the training set. To obtain the noisy 3D teacher required by our method, we lift the 2D motions back to 3D using either MotionBERT [57] or MVLift [25]. MotionBERT produces centered poses, so we recover an uncentered trajectory by estimating root depth from observed limb lengths (Section D); MVLift predicts global root trajectories directly and requires no such recovery.

Skeleton Conversion Both motionBERT and MVLift require a different skeleton than HumanML3D, in order to be as fair to the baselines as possible, we used HumanML3D data to render SMPL meshes, for which vertices conversion to the skeletons format is known, and convert the joints in 3D, and then rendered them to 2D for the lifter to lift using the same camera.

C.2 HumanML3D Evaluation Metrics

We report FID (statistical similarity between real and generated motion features), Diversity (average distance between random pairs), R-Precision Top-3 (text–motion alignment in a shared embedding), Multimodal Distance (distance to ground-truth motions), and Multimodality (variance across samples generated from the same prompt). All results are averaged over 20 generations and reported with 95% confidence intervals.

C.3 Fit3D Data Processing

For each 2D sequence extracted by RTMPose [18] BodyWithFeet performance model, we further smooth it using weighted temporal mean (with weights (0.25, 0.5, 0.25)), and convert to the SMPL joint format using manually picked weight. Sequences are split into clips matching the HumanML3D length range using the dynamic-programming procedure detailed in Section C.4. Text descriptions are generated from the exercise names in HumanML3D’s natural-language style.

For WHAM we use the official implementation and checkpoint, cut motions to clip length, and align them to the dataset frame on the first frame of each clip to avoid penalizing WHAM for not using the provided cameras at lifting evaluation time.

Skeleton Conversion WHAM requires no skeleton conversion as it uses the raw RGB image and outputs in SMPL format. MVLift, however, requires 17-keypoint skeletons. Since all work on fit3D that relies on 2D pose estimation uses RTMPose as pseudo-GT 2D poses, we needed to convert the joints directly in 2D to the MVLift skeleton format. To do so, we trained a 2D joint regressor using the accurate paired data created on HumanML3D, converting from the 22-joint format (HumanML) to the 17-joint format (MVLift), and applied it to all 2D poses.

C.4 Clip Extraction for Fit3D

The raw sequences in the dataset [9] vary significantly in duration, so we split each motion into shorter, natural clips using a lightweight dynamic-programming procedure. The goal is to produce segments that (i) fall within a desired length range, (ii) avoid splitting at high-motion frames, and (iii) prefer frames whose pose is close to the rest pose.

Split Cost. For each candidate split frame i , we compute a frame cost

$$\ell_{\text{frame}}(i) = \lambda_{\text{pose}} \ell_{\text{pose}}(i) + \lambda_{\text{vel}} \ell_{\text{vel}}(i),$$

where $\ell_{\text{pose}}(i)$ measures the MSE between the local pose at frame i and the rest local pose, and $\ell_{\text{vel}}(i)$ is the average local velocity magnitude around frame i . We use $\lambda_{\text{pose}} = 30000$ and $\lambda_{\text{vel}} = 15000$.

We also apply a clip-length penalty

$$\ell_{\text{len}}(\Delta) = (\Delta - L_{\text{target}})^2,$$

with $L_{\text{min}} = 60$, $L_{\text{target}} = 150$, and $L_{\text{max}} = 300$.

Dynamic Programming. Let $\text{DP}[i]$ store the best cumulative cost for segmenting frames $[0, i]$ and the index of the previous split. For each frame i , we consider all valid previous splits j such that

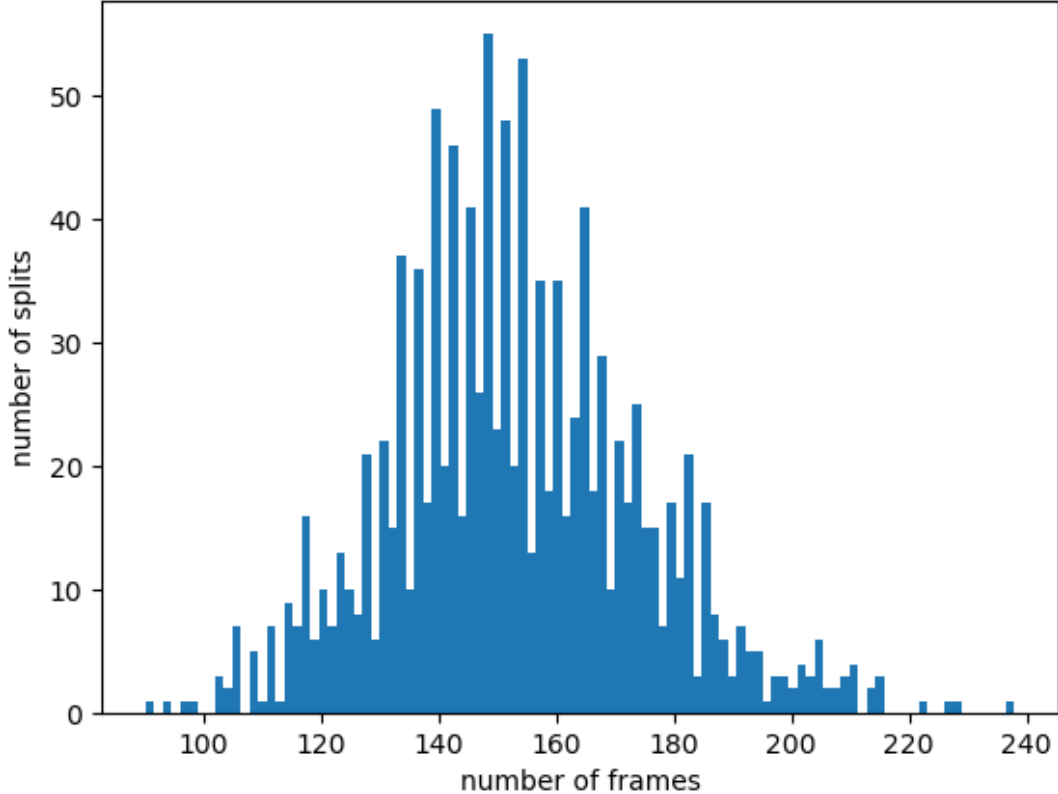


Figure 7: Histogram of resulting clip lengths after dynamic-programming segmentation.

$L_{\min} \leq i - j \leq L_{\max}$, and select the j minimizing

$$DP[j] + \ell_{\text{len}}(i - j) + \ell_{\text{frame}}(i).$$

After the forward pass, we choose the best terminal split and backtrack through the stored indices to recover all segment boundaries.

Outcome. This produces stable, naturally aligned clips that avoid erratic split points while keeping durations consistent across the dataset. Running the procedure over all sequences yields 1,161 clips in total (including subject *s09*). See Figure 7.

C.5 NBA Data Processing

Because the motions are scaled and centered, recovering a real pinhole camera from the data is infeasible. We therefore adopt MAS’s convention: a camera 7 units along the Z -axis at $\pi/16$ elevation. ElePose serves as the noisy teacher.

C.6 Compute Cost Estimates

All experiments were run on a single NVIDIA RTX 4090 GPU (24 GB VRAM). Table 4 summarises the compute required for each method. Lifter preprocessing (running MotionBERT or MVLift on the full dataset) is a one-time cost shared by any lifter-based approach; the resulting pseudo-3D labels are reused across all training runs.

Hyperparameters Search The sweep described in Appendix F comprised approximately 56 runs of ~ 20 hours each ($\sim 1,120$ GPU-hours total), which is not reflected in the per-method figures above.

Earlier Exploration Besides the compute presented here, earlier versions of the paper with contributions which were removed because of their lack of support from the ablations included other hyperparameter searches, one small and one larger, as well as early exploration training and the blender rendering of all motions for the survey / supplementary material. We do not have an exact number for total compute used but it could be estimated as around $2.5\times$ the values reported for the hyperparameter search and lifting / training models for experiments.

Table 4: Compute costs on a single RTX 4090. Inference is reported in seconds per 100 generated samples.

Stage	Ours	MDM (on lifter)	MAS
Lifter preprocessing (hr)	~200	~200	—
Training (hr)	~46	~36	~6
Inference (sec / 100)	~5	~5	~12

D Naive Root Depth Estimation

We denote by $\{\mathbf{x}_i^{3D}\}_{i=1}^{22}$ the 3D joint locations of the articulated skeleton and by $\{\mathbf{x}_i^{2D}\}_{i=1}^{22}$ their image projections under a calibrated pinhole camera with focal length f . The skeleton connectivity forms a tree whose edges (bones) have known rest lengths $L_i^{3D} = \|\mathbf{x}_{p(i)}^{3D} - \mathbf{x}_i^{3D}\|$, where $p(i)$ is the parent joint of node i . For each frame, we observe only the 2D joint coordinates and the corresponding 2D bone lengths $L_i^{2D} = \|\mathbf{x}_{p(i)}^{2D} - \mathbf{x}_i^{2D}\|$.

Assumptions. We (falsely) assume that the global 3D orientation of the skeleton is uniformly random, and that the three bones incident to the root joint (pelvis) are independently and uniformly oriented in $SO(3)$. Under this assumption, the direction of each bone is uniformly distributed over the unit sphere, so that $\cos \theta \sim \text{Unif}[-1, 1]$, where θ is the angle between the bone and the camera optical axis.

For a bone of true length L^{3D} viewed at mean depth z and at an angle θ to the optical axis, its projected 2D length satisfies approximately

$$L^{2D} \approx \frac{f}{z} L^{3D} \sin \theta, \quad (8)$$

which rearranges to

$$\frac{L^{3D}}{L^{2D}} \approx \frac{z}{f \sin \theta}. \quad (9)$$

Estimator. Because the orientations are random, with high probability at least one of the three root-adjacent bones will have $\sin \theta$ close to 1. We therefore define a simple, ‘naive’ depth estimator as

$$\hat{z}_{\text{root}} = f C_3 \min_{i \in \mathcal{N}_{\text{root}}} \frac{L_i^{3D}}{L_i^{2D}}, \quad (10)$$

where $\mathcal{N}_{\text{root}}$ denotes the three edges connected to the root.

Constant. The constant C_3 corrects the expectation bias due to the non-uniform distribution of $\sin \theta$ under uniform random orientations. By integrating over this distribution for three independent samples, we obtain

$$C_3 = \frac{1}{\mathbb{E}[1/S_{\text{max}}]} \approx 0.9358, \quad (11)$$

where $S_{\text{max}} = \max(\sin \theta_1, \sin \theta_2, \sin \theta_3)$. Thus, in normalized camera coordinates ($f = 1$),

$$\hat{z}_{\text{root}} \approx 0.9358 \min_{i \in \mathcal{N}_{\text{root}}} \frac{L_i^{3D}}{L_i^{2D}}.$$

This estimator provides a simple and numerically stable baseline for root-depth inference from 2D skeletons, assuming uniformly random joint orientations and known bone lengths.

E Formal Definition of Recall[†]

Recall Alternative. [22] define the notion of ‘close enough’ as follows: for each sample ϕ' , find its K -th (usually taken as 3) nearest neighbor within its own manifold Φ , and say that sample ϕ is

"close enough" if $|\phi - \phi'|_2 \leq |\phi' - NN_K(\phi', \Phi)|_2$. Thus, for real and generated features Φ_r, Φ_g , we can define:

$$P = \frac{1}{|\Phi_g|} \sum_{\phi \in \Phi_g} \mathbf{1}\{\exists \phi' \in \Phi_r : |\phi - \phi'|_2 \leq |\phi' - NN_K(\phi', \Phi_r)|_2\}$$

$$R = \frac{1}{|\Phi_r|} \sum_{\phi \in \Phi_r} \mathbf{1}\{\exists \phi' \in \Phi_g : |\phi - \phi'|_2 \leq |\phi' - NN_K(\phi', \Phi_g)|_2\}$$

Note that to estimate recall, we open hyperspheres around generated samples based on *generated* nearest neighbors. This biases the metric towards favoring exaggerated diversity in the generated distribution. We therefore suggest a variant of the recall metric that respects the spread of the real distribution by computing the NN in the real distribution.

$$R^\dagger = \frac{1}{|\Phi_r|} \sum_{\phi \in \Phi_r} \mathbf{1}\{\exists \phi' \in \Phi_g : |\phi - \phi'|_2 \leq |\phi - NN_K(\phi, \Phi_r)|_2\}$$

F Hyper Parameter Choices

Overview. The only hyperparameters shown to affect method quality were $\lambda_{vel} t^*$ and number of steps for multistep under t^* . The hyper parameter search resulted in $\lambda_{vel} = 287$ on HumanML3D and $\lambda_{vel} = 187$ on NBA. $t^* = 30$ on HumanML3D and $t^* = 23$ on NBA and $\#steps = 3$ on HumanML3D and $\#steps = 4$ on NBA.

MDM Hyperparameters. For MDM, we follow the original configuration from [46], with two exceptions: (i) we use $T = 50$ diffusion steps, which has become standard practice over the past year, and (ii) for the NBA experiment, we modify the noise schedule by setting $\tau = 2$ to place greater emphasis on high-noise denoising stages.

G Mathematical Formulations of Diffusion Sampling and Guidance

We assume the DDIM notation of [44], where for a noise schedule $t \in \{1, \dots, T\}$ with noising weights β_1, \dots, β_T , the cumulative product $\alpha_t = \prod_{j=1}^t (1 - \beta_j)$ defines the forward noising process of data $\mathbf{x} \sim p$:

$$\mathbf{x} \sim p$$

$$q(\mathbf{x}_t | \mathbf{x}) \sim \mathcal{N}(\sqrt{1 - \alpha_t} \mathbf{x}, \alpha_t \mathbf{I})$$

$$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$$

DDPM [16] provides a method to sample from p using a denoiser that estimates $p(\mathbf{x} | \mathbf{x}_t)$, as summarized in Section G.2. DDIM generalizes this formulation and enables more efficient sampling, described (in part) in Section G.1.

G.1 DDIM Used in Training

Formal Multistep Training. During training, to optimize the model over intermediate steps $t < t^*$, we use DDIM sampling to skip from some $t_b \geq t^*$ down to t_{b-1}, \dots, t_1 , where b is the number of buckets defined by LIS [34]. Each t_i is sampled uniformly as $t_i \sim \mathcal{U}((i-1)B, (i-1)B + B - 1)$, with bucket size $B = \lceil t^*/(b-1) \rceil$. For our hyperparameters $t^* = 12$ and $b = 3$, this yields $t_3 \sim \mathcal{U}(12, 17)$, $t_2 \sim \mathcal{U}(6, 11)$, and $t_1 \sim \mathcal{U}(0, 5)$.

We randomly mix batches where $t \geq t^*$ and the standard diffusion loss is used, with batches where $t < t^*$. For the latter, we first sample $\mathbf{x}_{t_3} = \sqrt{\alpha_{t_3}} \tilde{\mathbf{x}} + \sqrt{1 - \alpha_{t_3}} \epsilon$ with $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. We then iteratively compute, for $i = 3, 2, 1$:

$$\hat{\mathbf{x}}_{i \rightarrow 0} = D_\theta(\text{stop_gradient}(\mathbf{x}_{t_i}), t_i) \quad (12)$$

$$\mathcal{L}_i = \mathcal{L}_{\text{total}}(\hat{\mathbf{x}}_{i \rightarrow 0}) \quad (13)$$

$$\hat{\epsilon}_i = \frac{\mathbf{x}_{t_i} - \sqrt{\alpha_{t_i}} \hat{\mathbf{x}}_{i \rightarrow 0}}{\sqrt{1 - \alpha_{t_i}}} \quad (14)$$

$$\mathbf{x}_{t_{i-1}} = \sqrt{\alpha_{t_{i-1}}} \hat{\mathbf{x}}_{i \rightarrow 0} + \sqrt{1 - \alpha_{t_{i-1}}} \hat{\epsilon}_i \quad (15)$$

The batch loss is the average $\mathcal{L} = \frac{1}{3}(\mathcal{L}_3 + \mathcal{L}_2 + \mathcal{L}_1)$.

G.2 DDPM Used in Lifting

DDPM Sampling. Full-sampling quality is best with DDPM, so for lifting we apply guidance within the DDPM inference procedure:

$$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I}) \quad (16)$$

$$\text{for } t = T, T-1, \dots, 2, 1 : \quad (17)$$

$$\hat{\mathbf{x}}_0 = D_\theta(\mathbf{x}_t, t) \quad (18)$$

$$\epsilon_t \sim \mathcal{N}(0, \mathbf{I}) \quad (19)$$

$$w_{\hat{\mathbf{x}}_0} = \frac{\sqrt{\alpha_{t-1}\beta_t}}{1 - \alpha_t} \quad (20)$$

$$w_{\mathbf{x}_t} = \frac{\sqrt{1 - \beta_t}(1 - \alpha_{t-1})}{1 - \alpha_t} \quad (21)$$

$$w_{\epsilon_t} = \sqrt{\frac{1 - \alpha_{t-1}}{1 - \alpha_t}} \beta_t \quad (22)$$

$$\mathbf{x}_{t-1} = w_{\hat{\mathbf{x}}_0} \hat{\mathbf{x}}_0 + w_{\mathbf{x}_t} \mathbf{x}_t + w_{\epsilon_t} \epsilon_t \quad (23)$$

Applying 2D Guidance. To apply ray-projection guidance during DDPM sampling, we insert:

$$\hat{\mathbf{x}}_0 = P_\Pi(\hat{\mathbf{x}}_0, \mathbf{y}) \quad (24)$$

after equation (19). This procedure is a specific instance of guided diffusion introduced by [45] and a relaxation of [23], which handles the harder case of unknown camera.

H Human Preference Survey

The human surveys were conducted through a web-based interface with anonymous crowdworkers recruited via Prolific. Participants were unaware of which method generated which motion; clips were labeled only as *Video A* and *Video B* with randomized left/right assignment. Both clips played automatically on page load, and choice buttons remained disabled until both had finished playing, ensuring each motion was watched in full.

The **NBA survey** (20 participants) compared our method against MAS [19] for unconditional basketball motion generation. Each participant evaluated 10 pairs of stick-figure animations and selected the one that looked more realistic and natural. No text prompt or reference motion was provided. A representative screenshot from the survey interface is shown in Figure 8.

The **Fit3D survey** (52 participants) evaluated six methods for text-conditioned motion generation. Each participant evaluated 20 pairs, one per text prompt, where the two methods shown were drawn at random from the six candidates. Each page displayed the text prompt used for sampling and a reference video from the Fit3D dataset [9] to ground participants’ understanding of the described action, alongside the two candidate clips. A representative screenshot from the survey interface is shown in Figure 9.

For both surveys the following text instructions were introduced:

Participant Consent Form

Study title: Human Evaluation of AI-Generated Human Motion

Purpose: This study investigates the perceptual quality of computer-generated human motion animations.

Participation: You will watch 20 pairs of short video clips and indicate which appears more realistic (2–4 minutes).

Risks: There are no known risks.

Confidentiality: No personally identifiable information is collected. All responses are anonymous.

Voluntary: You may stop at any time without penalty.

Compensation: Upon completion you will receive a unique code to claim your compensation.

Motion Generation Survey

You will evaluate **20 pairs of motion clips**. Each pair shows two animations for the same text prompt; choose whichever looks more **realistic and natural**.

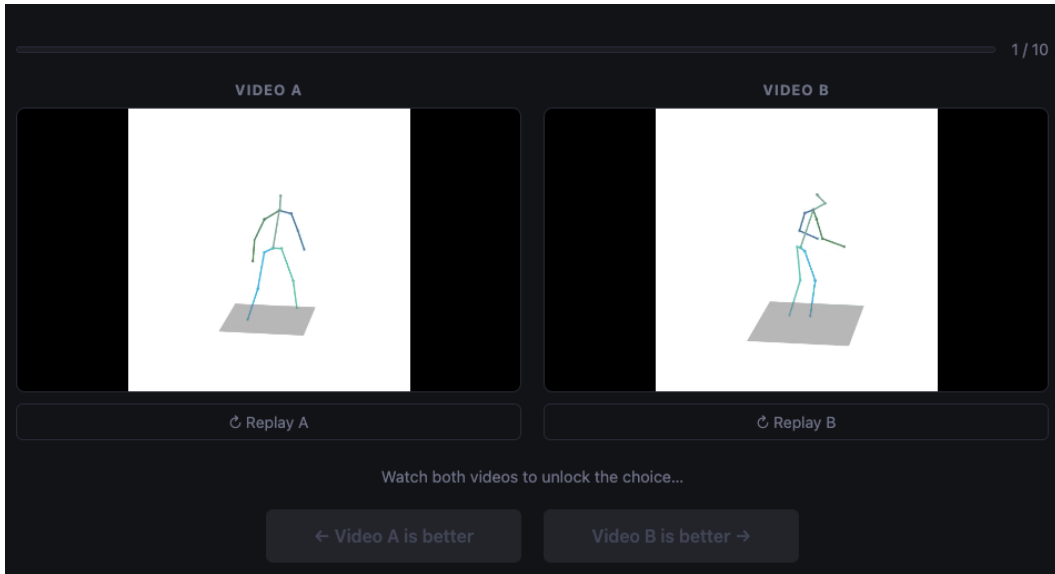


Figure 8: A representative screenshot from the NBA Human Preference Survey interface.

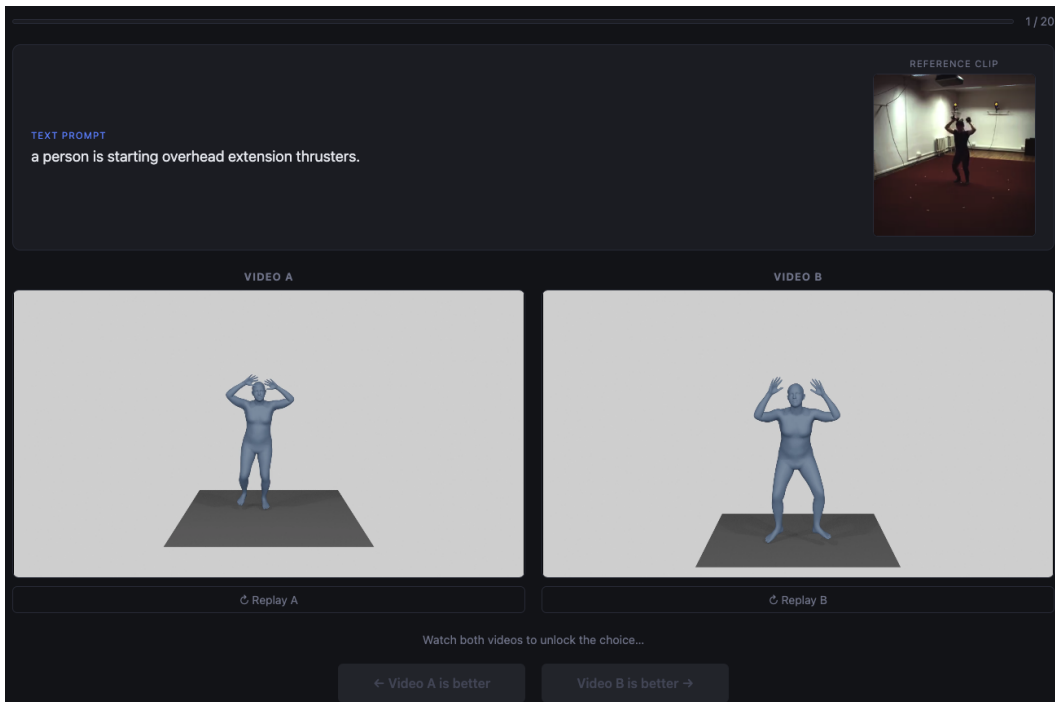


Figure 9: A representative screenshot from the Fit3D Human Preference Survey interface.

- Read the text prompt at the top of each page.
- Both videos play automatically; choice buttons unlock after both finish.
- You may replay clips as many times as you like before choosing.
- The survey takes approximately 2–4 minutes.

Participants were compensated according to the survey duration, with an average effective rate of £15.53/hour across both surveys.

I Camera Parameters Estimation

When camera parameters are unavailable, we estimate them by solving EPnP between the first 24 frames (minimal motion length used by MDM [46]) of the 3D estimated pose (MVLift [25] results on HumanML3D [13] and WHAM [41] on Fit3D [9]) and the 2D poses. To do so we use OpenCV [4] EPnP solver and Levenberg-Marquardt pose refinement, both with default parameters.

J Explicit HumanML Channel Partitioning

HumanML3D’s representation [13] is composed of:

1. 1 channel for angular velocity around the y-axis, 2 channels for root velocity in the XZ plane, 1 channel for root height.
2. 3 channels per non-root joint, representing X (root coordinate frame) Y (global) and Z (root coordinate frame).
3. 6 channels per non-root joint, representing the 6D continuous rotations of the joints in relation to the rest pose angle (T-shape human), each joint rotation is calculated as the normalized displacement from its ancestor.
4. 3 channels per joint (including root) representing the per-joint velocity.
5. 4 channels representing the 4 foot contact flags. For the NBA dataset with only 2 foot joints we replicate these flags per foot.

So in total our x is composed of $A_J = 4 + (J - 1) \times 3$ channels and \mathbf{r} of $B_J = (J - 1) \times 6 + J \times 3 + 4$ channels.