ViMo-Flow: Stable Frame-Aware Video-to-Motion Diffusion Model via SNR Weighting Strategies

submitted in partial fulfillment of the requirements

for the degree of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

LOGESH .V CS24M106

Supervisor(s)

Chalavadi Vishnu

भारतीय प्रौद्योगिकी संस्थान तिरुपति

TIRUPATI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI

December 1, 2025

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission to the best of my knowledge. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Place: Tirupati
Date: December 1, 2025

Signature
Logesh .V
CS24M106

BONA FIDE CERTIFICATE

This is to certify that the report titled **ViMo-Flow: Stable Frame-Aware Video-to-Motion Diffusion Model via SNR Weighting Strategies**, submitted by **Logesh .V**, to the Indian Institute of Technology, Tirupati, for the award of the degree of **Master of Technology**, is a bona fide record of the project work done by him under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Tirupati
Date: December 1, 2025

Chalavadi Vishnu
Guide
Assistant Professor
Department of Computer Science & Engineering
IIT Tirupati - 517619

ABSTRACT

This report presents the first milestone in developing a comprehensive pipeline for generating physically plausible human motions from casual video inputs. The current work focuses on significantly enhancing the Video-to-Motion (ViMo) generation stage through the integration of advanced diffusion training methodologies. We implement a novel ViMo architecture capable of extracting high-quality 3D skeletal motions from unconstrained video footage, while incorporating two key improvements to the denoising diffusion probabilistic model (DDPM) training process: Signal-to-Noise Ratio (SNR) weighting for optimized training efficiency and a vectorized timestep approach for enhanced temporal modeling.

Our enhanced diffusion framework demonstrates improved convergence properties and motion quality compared to baseline ViMo implementations, establishing a robust foundation for subsequent physics-based processing stages. The proposed system represents a critical advancement in bridging the visual-to-physical domain gap, providing high-fidelity motion data that preserves the nuanced characteristics observed in source videos while maintaining temporal coherence essential for physical simulation.

Looking forward, this work serves as the cornerstone for a complete pipeline that will progress through physics-based imitation learning and conclude with an innovative policy adaptation mechanism to ensure robust transfer to dynamic simulation environments. The current implementation marks substantial progress toward enabling seamless translation of casual video content into physically valid character animations.

TABLE OF CONTENTS

LIST OF FIGURES

I Introduction

Realistic 3D human motions are the "soul" of virtual characters, bringing them to life by enabling natural jumping, fighting, and dancing. As portable devices and the metaverse accelerate demand for large-scale character creation, generating adaptive motions from casual video remains challenging. Recent video-to-motion methods like ViMo synthesize diverse 3D motions directly from unconstrained video despite camera motion and occlusion [19], leveraging abundant online footage to produce contextually consistent results. Data-driven generative approaches—such as text-to-motion diffusion models [24] and music-conditioned dance generation like EDGE [22]—offer scalable alternatives but inherit biases from limited datasets like AMASS [15], Human3.6M [11], and AIST++ [12]. As a result, models trained on narrow domains struggle to generalize to styles such as classical dance or figure skating, reflecting the ongoing scarcity and high cost of collecting diverse 3D motion data.

Casual video offers an escape from this scarcity. Platforms host billions of clips spanning martial arts, ballet, occupational tasks, and impromptu performance. Unlike curated motion-capture datasets, these recordings feature dynamic camera work, shot transitions, occlusion, and varied environments—precisely the failure modes of conventional 3D pose estimators. Methods such as MotionBERT [26] and GLAMR [23] regress 3D joints from 2D observations but rely on static or accurately estimable camera parameters. Under casual video conditions, their frame-wise predictions accumulate errors, producing jittery and implausible trajectories. The regression objective itself becomes brittle when perspective shifts violate the single-view assumption.

In our work, we build upon the ViMo framework by integrating diffusion training techniques that enhance both the quality and efficiency of motion generation. Specifically, we adopt the Min-SNR weighting strategy, which treats the denoising process at each timestep as an individual task and assigns loss weights according to the clamped signal-to-noise ratio (SNR) [8]. This strategy balances conflicting gradients across timesteps, resulting in faster convergence and improved motion realism. Furthermore, we incorporate a vectorized timestep formulation inspired by the Probabilistic Timestep Sampling Strategy (PTSS), which enables more flexible and efficient temporal modeling in video diffusion [13]. By sampling timesteps probabilistically, our method avoids the combinatorial explosion associated with independent frame timesteps, making training scalable for complex motion sequences.

ViMo [19] reframes video-to-motion as a generative task rather than a reconstruction problem. Instead of estimating exact 3D coordinates and camera pose, a diffusion model synthesizes plausible motion sequences conditioned on the 2D pose trajectory extracted from video. Multi-view projections of the generated motion are compared against the input 2D sequence, bypassing explicit camera modeling. The denoising network operates over the full temporal sequence, enforcing coherence through transformer self-attention. This approach tolerates missing frames, occlusions, and rapid viewpoint changes while producing diverse motions that match the stylistic silhouette and rhythm of the source video.

The present work centers on a novel implementation of ViMo enhanced by contemporary diffusion training advances. Two technical improvements define our training regime. First, we integrate the Min-SNR weighting strategy [8]. Standard diffusion training uses uniform loss weighting across noise levels, yet low-SNR steps contribute minimal signal relative to noise, slowing convergence. Min-SNR reweights each timestep by its signal-to-noise ratio, emphasizing high-SNR steps and suppressing low-SNR noise. This yields faster training, lower final loss, and sharper motion details. Second, we adopt the Vectorized Timestep Approach [13], which encodes diffusion timesteps as dense vectors rather than scalar indices. In motion generation, temporal structure matters: the phase relationship between music beats and dance steps, or the cyclic nature of gait, depends on precise timing. Scalar timesteps discard this context. Vectorization embeds timestep information as a function of sequence position, enabling the transformer to exploit long-range temporal correlations. The result is improved rhythmic alignment and smoother interpolation across frames.

Combined, these techniques yield a denoising model that trains efficiently and captures nuanced temporal dynamics while retaining ViMo's transformer backbone with FiLM conditioning and classifier-free guidance. Our contribution lies not in architectural novelty but in a refined training recipe that delivers robust performance on highly casual video footage. Although this report focuses on the video-to-motion generative core, it forms the first stage of a broader three-stage animation pipeline: extracting kinematic motion from video, translating it into physics-based control policies through imitation learning, and ultimately applying an adaptive refinement loop to correct deployment-time errors. The latter stages remain future work, included here only to situate the present effort within the larger vision.

II Related Work

The field of human motion generation has seen rapid advancements, driven by growing demand for scalable and diverse animation in virtual environments. Early approaches relied on manual animation or expensive motion-capture setups, which limited both the quantity and variety of available motion data [15]. The emergence of data-driven methods—particularly deep generative models—has broadened possibilities for automated synthesis, but many frameworks remain constrained by the need for paired ground-truth data and are often limited to the action categories present in their training sets [22, 24]. Diffusion-based denoising models have recently improved fidelity and diversity [10], and music- or text-conditioned systems produce synchronized dances or language-aligned motions [24, 22, 12]; nevertheless, the dependence on curated 3D motion corpora sustains a persistent diversity bottleneck.

ViMo bypasses this bottleneck by framing video-to-motion as a generative task rather than strict reconstruction: a transformer-based diffusion model denoises 3D motion conditioned on 2D pose trajectories from casual video, comparing multi-view projections against the input to avoid explicit camera estimation and to tolerate occlusion and viewpoint shifts [19]. Building on this backbone, we focus on improving training dynamics: Min-SNR weighting accelerates convergence by reweighting loss terms according to signal-to-noise ratio, and vectorized timestep encoding replaces scalar diffusion indices with dense temporal embeddings to better capture rhythmic and long-range temporal structure [8, 13]. These modifications yield a model that trains efficiently on casual video and produces motions with sharper kinematic detail and improved rhythmic alignment.

2.1 Action-Conditioned Generation

Action-conditioned methods generate motions from categorical labels or seed poses, operating entirely within the training distribution. Degardin et al. propose a generative adversarial graph convolutional network that synthesizes human actions conditioned on discrete class labels, yet the model cannot extrapolate beyond the 10-15 action categories present during training [4]. Similarly, Action2Motion [7] employs a VAE-transformer architecture to generate 3D motions from action categories, but the generated repertoire remains bounded by the predefined set of 30 actions in the NTU RGB+D dataset. These approaches share a fundamental limitation: the conditioning signal is a one-hot or learned embedding of a seen category. The model learns

a mapping from label to motion distribution but acquires no mechanism to compose unseen behaviors. The vocabulary of actions is closed at training time. Our video-to-motion framework sidesteps this restriction by conditioning on 2D pose trajectories rather than categorical tokens. A video of a never-before-seen dance style provides a kinematic blueprint that the diffusion model translates into plausible 3D motion, effectively enabling zero-shot generation of unseen categories.

2.2 Pose Estimation from Videos

Reconstructing 3D pose from monocular video follows a two-stage pipeline: detect 2D keypoints, then lift to 3D via convolutional or graph architectures. Cai et al. exploit spatial-temporal relationships through graph convolutional networks, aggregating joint correlations across frames to improve 3D accuracy [1]. Martinez et al. demonstrate that a simple MLP operating on 2D detections can achieve competitive results when camera parameters are known [16]. More recent work, SMPLer-X [2], scales training data to 4.5 million frames and predicts expressive SMPL-X parameters, yet relies on accurate camera calibration or approximates intrinsics from image metadata. These methods assume static cameras or easily estimable camera motion; under casual video with rapid zoom, pan, and shot cuts, the 2D-to-3D lifting objective becomes ill-posed. Frame-wise predictions accumulate drift, yielding jittery trajectories that violate physical plausibility. GLAMR [23] attempts global trajectory optimization to smooth results, but computational cost is high and performance degrades under aggressive montage editing.

A related line of work synthesizes novel views of human interactions from sparse multi-view videos [21]. This approach interpolates intermediate viewpoints given a few calibrated cameras, relaxing dense capture requirements. However, it still assumes known camera geometry and static scene capture, typically warping explicit 3D geometry or using light-field interpolation. Both strategies remain fragile under rapid camera motion or occlusions characteristic of casual footage. ViMo's generative formulation avoids explicit 3D regression entirely. Instead of minimizing per-joint reprojection error, the diffusion objective encourages synthesized motion to project consistently across multiple virtual views, implicitly handling camera variation without calibration.

2.3 Temporal Modeling in Video Diffusion

Standard diffusion models treat timestep as a scalar, ignoring frame-to-frame temporal structure and discarding periodicity and phase relationships inherent in motion [10]. The Vectorized Timestep Approach redefines this by encoding timestep as a dense vector function of sequence position, allowing the transformer to capture long-range correlations and rhythmic patterns [13]. Our implementation introduces a **Probabilistic Timestep Sampling Strategy (PTSS)** to manage computational cost. In naive vectorized diffusion, sampling distinct timesteps per frame leads to a combinatorial explosion, with 1000^N combinations for N frames. PTSS introduces a probability p governing independent versus shared sampling: with probability p, each frame gets its own timestep for independent evolution; with probability $1-p$, a single timestep is broadcast across all frames, preserving efficiency. This hybrid balances temporal expressivity and training cost. In practice, $p = 0.3$ balances diversity and stability, preventing overfitting to arbitrary frame-wise noise schedules while retaining fine-grained temporal control. Our experiments show PTSS improves beat-alignment scores for music-conditioned generation without increasing training time.

2.4 Diffusion Training Optimization

Training diffusion models requires careful balancing of loss contributions across the noise schedule. Ho et al. use constant weighting $w_t = 1$, treating all timesteps equally [10]. SNR weighting $w_t = \alpha_t^2/\sigma_t^2$ emphasizes high-SNR steps, but is numerically equivalent to constant weighting when predicting noise [17]. Max-SNR-γ clipping avoids zero weight at final timesteps but concentrates updates on low-noise regimes [20]. Min-SNR-γ weighting inverts this logic: $w_t = \min \alpha_t^2/\sigma_t^2, \gamma$ caps the weight at small noise levels, forcing the model to prioritize high-noise timesteps where the signal structure is coarse [8]. This accelerates early training and prevents overfitting to fine perturbations. Our implementation adopts $\gamma = 5$ as default, following ablations in the original work. Empirically, Min-SNR reduces training epochs by 30% and yields lower FID on kinetic features compared to constant weighting. For motion generation, where high-noise steps correspond to coarse pose and trajectory, this emphasis improves global structure before refining local joints. The weighting integrates seamlessly with the vectorized timestep encoder, forming a cohesive, efficient, and temporally aware training regime.

III Methodology

3.1 Overview

We present a novel approach to generating 3D human motions directly from casual videos, overcoming challenges posed by complex camera movements, occlusions, and the scarcity of labeled motion data. At a high level, the pipeline maps multi-view 2D pose sequences extracted from videos to temporally coherent 3D motion sequences, using a diffusion-based denoising process. This approach allows the model to synthesize diverse and realistic motions without explicit camera calibration or manual annotation. The core of the framework is a ViMo-style [19] denoiser that takes time-series 2D pose observations as input and outputs per-frame 3D motion data, including joint rotations, discrete foot-contact signals, and root trajectory. The denoiser is trained within a DDPM [10] framework, enhanced with SNR-based loss weighting and a probabilistic, vectorized timestep sampling strategy [13]. These innovations ensure that the model learns meaningful motion structure across all noise levels and generalizes well to unseen video content.

3.2 Problem Statement

3.2 (i) Task Definition

The input is a casual video featuring a single person. 2D pose sequences are extracted using Pose Extractors [3, 6], resulting in a tensor $p \in \mathbb{R}^{S \times J_{2d} \times C_{in}}$, where $S = 150$ frames (5 seconds at 30 FPS), $J_{2d} = 17$ joints (COCO format), and $C_{in} = 3$ (x, y, confidence). Missing or low-confidence joints are masked to ensure reliable conditioning. The objective is to generate a corresponding 3D motion sequence $M \in \mathbb{R}^{S \times (J_{3d} \cdot C + 4 + 3)}$, comprising:

- Flattened joint rotations $R \in \mathbb{R}^{J_{3d} \times C}$ with $J_{3d} = 24$ joints in 6D rotation space ($C = 6$),
- Foot contact labels $f \in \{0, 1\}^4$ for ankles and feet,
- Root position $t \in \mathbb{R}^3$ (global translation).

The output preserves the stylistic qualities of the input video and is suitable for downstream physics-based simulation and animation tasks.

3.2 (ii) Dataset

The AIST++ dataset provides the training corpus, containing 1,408 sequences of 3D dance motions, multi-view videos, and synchronized annotations [12]. Motion sequences are stored as SMPL pose parameters (smpl_poses $\in \mathbb{R}^{N\times 24\times 3}$) and global translation (smpl_trans $\in \mathbb{R}^{N\times 3}$). 2D keypoints (keypoints2d $\in \mathbb{R}^{9\times N\times 17\times 3}$) represent nine calibrated camera views, N frames, 17 COCO-format joints, and per-joint (x, y, confidence) values in pixel coordinates at 1920×1080 resolution. Raw and optimized 3D keypoints (keypoints3d $\in \mathbb{R}^{N\times 17\times 3}$) are provided, with timestamps annotated at 60 FPS. The dataset covers 10 dance genres and hundreds of choreographies, making it suitable for training models to generalize across motion categories and camera conditions [12].

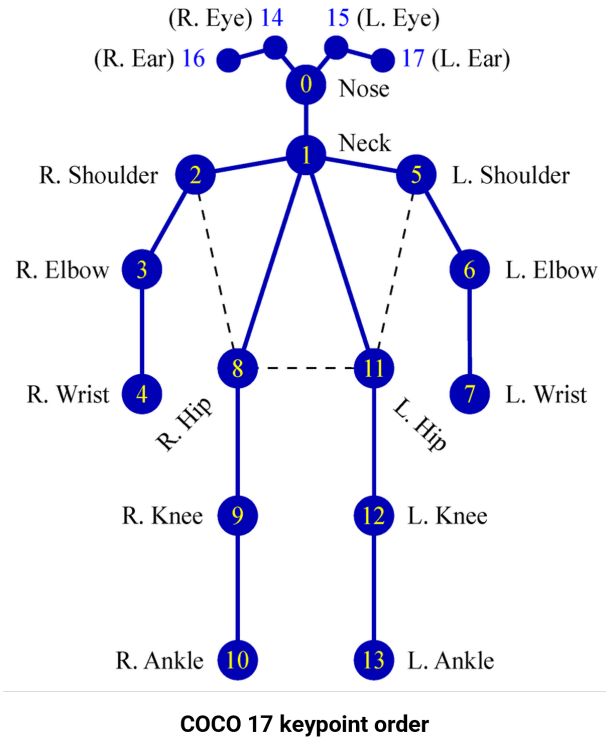


Figure 1: Human Skeleton's COCO 17-Keypoint Order.

3.2 (iii) Pose Estimators

The 2D skeletons used for conditioning are extracted using OpenPose [3] and AlphaPose [6]. Both models provide per-joint confidence scores, which are used to mask out unreliable detections. In multi-view scenarios, the highest-confidence detection per joint is selected or aggregated through a lightweight learned module. Optional temporal smoothing can reduce jitter, but is applied cautiously to avoid removing meaningful micro-motions. Occlusion and

rapid camera motion are common in casual footage; the pipeline treats pose estimates as noisy and designs both the denoiser and auxiliary losses to be resilient to such errors [19].

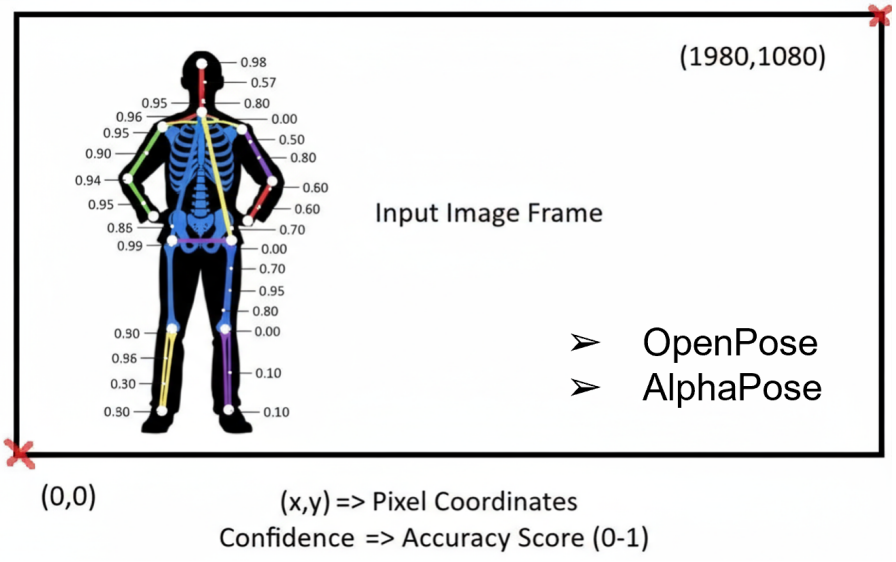


Figure 2: Representation of the Pose Estimator's Output

3.2 (iv) Preprocessing

Raw AIST++ [12] sequences are resampled from 60 FPS to 30 FPS by selecting every second frame. Sequences are partitioned into non-overlapping clips of 150 frames (5 seconds). The conditioning tensor cond_2d retains shape $V \times 150 \times 17 \times 3$ for V views. Ground-truth 3D motion m3d_gt is constructed per clip as 150×151, encoding 24 joints in 6D rotation format (144 dimensions), 4 foot contact labels, and 3 root position values. Sequences listed in ignore_list.txt are omitted. SMPL axis-angle rotations are converted to 6D using orthogonalization [25]. Foot contact labels are pre-computed from foot and toe vertical velocities near zero. The 2D pose coordinates are given in image pixels, which are not directly meaningful for 3D motion generation. To address this, the preprocessing normalizes the poses by centering them relative to the root joint, ensuring the model conditions on relative joint positions rather than absolute image coordinates. This step is crucial for generalizing across different camera views and video scales. The resulting tensors feed the diffusion training loop directly, ensuring temporal coherence and compatibility with the model architecture [19].

3.3 ViMo framework

3.3 (i) Diffusion Formulation

The denoising process follows a standard Markovian corruption of the clean motion $\mathbf{m}_0$. The forward transitions are defined as

$$q(\mathbf{m}_t|\mathbf{m}_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}\mathbf{m}_{t-1}, (1-\alpha_t)\mathbf{I}),$$

producing $\mathbf{m}_t$ at each step t. Unlike the original DDPM parameterization, which predicts the noise term ε, the network $D(\mathbf{m}_t, t, \mathbf{p})$ directly outputs an estimate of the clean motion $\hat{\mathbf{m}}_0$. This approach is better suited to the structured output space—6D rotations [25], root translation, and binary contact labels—where predicting noise would require additional post-processing. The training objective minimizes the simple L2 loss

$$\mathcal{L}_{\text{simple}} = \mathbb{E}[\|\mathbf{m}_0 - D(\mathbf{m}_t, t, \mathbf{p})\|^2],$$

averaged over random timesteps and paired samples. At inference, the posterior mean $\tilde{\mu}_t$ is computed from $\hat{\mathbf{m}}_0$ using the closed-form expression

$$\tilde{\mu}_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\mathbf{m}}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{m}_t,$$

enabling deterministic DDIM sampling or stochastic DDPM sampling.

Algorithm 1 Forward Diffusion Sampling

Require: Clean input $\mathbf{x}_0$, timestep vector $\mathbf{t}$, noise schedule $\{\beta_t\}_{t=1}^{T}$
Ensure: Noisy sample $\mathbf{x}_t$

Initialize $\alpha_t = 1 - \beta_t$ for $t = 1, \ldots, T$
Compute $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ ▷ Cumulative product
Sample $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with dimension of $\mathbf{x}_0$
$\bar{\alpha}_{\mathbf{t}} \leftarrow \text{gather}(\bar{\alpha}, \mathbf{t})$ ▷ Select values
    if $dim(\mathbf{t}) = 1$ **then**
    $\bar{\alpha}_{\mathbf{t}} \leftarrow \text{reshape}(\bar{\alpha}_{\mathbf{t}}, (B, 1, 1))$ **else**
    $\bar{\alpha}_{\mathbf{t}} \leftarrow \text{reshape}(\bar{\alpha}_{\mathbf{t}}, (B, S, 1))$ ▷ PTSS case
$\mathbf{x}_t \leftarrow \sqrt{\bar{\alpha}_{\mathbf{t}}} \cdot \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_{\mathbf{t}}} \cdot \varepsilon$ **return** $\mathbf{x}_t$

Algorithm 2 Reverse Denoising Sampling

Require: Model $\mathscr{M}$, initial state $\mathbf{x}_T$, steps T, conditioning $\mathbf{p}_{2d}$, flag σ_{flag}, schedule $\{\beta_t\}_{t=1}^{T}$
Ensure: Generated motion sequence $\mathbf{x}_0$

Initialize $\alpha_t = 1 - \beta_t$ for $t = 1, \ldots, T$
Compute $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ ▷ Cumulative product
    if $\mathbf{x}_T$ *is dimension tuple* **then**
    $\mathbf{x}_t \sim \mathscr{N}(\mathbf{0}, \mathbf{I})$ with shape $\mathbf{x}_T$ **else**
    $\mathbf{x}_t \leftarrow \mathbf{x}_T$ ▷ Clone tensor

Set batch size $B \leftarrow \dim(\mathbf{x}_t, 0)$
    for $t = T - 1$ *down to* 0 **do**
    $\mathbf{t} \leftarrow$ vector of length B with value t ▷ Batch timestep vector
$\hat{\mathbf{x}}_0 \leftarrow \mathscr{M}(\mathbf{x}_t, \mathbf{t}, \mathbf{p}_{2d})$ ▷ Model prediction
    if $t > 0$ **then**
    $\bar{\alpha}_t \leftarrow \bar{\alpha}[t]$
$\bar{\alpha}_{t-1} \leftarrow \bar{\alpha}[t-1]$
$c_1 \leftarrow \sqrt{\bar{\alpha}_{t-1}} \cdot \frac{\beta_t}{1-\bar{\alpha}_t}$
$c_2 \leftarrow \sqrt{\alpha_t} \cdot \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}$
$\mu \leftarrow c_1 \cdot \hat{\mathbf{x}}_0 + c_2 \cdot \mathbf{x}_t$ ▷ Posterior mean
    if $\sigma_{flag} = True$ **then**
    $\sigma^2 \leftarrow \beta_t \cdot \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}$ ▷ Posterior variance
$\varepsilon \sim \mathscr{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{x}_t \leftarrow \mu + \sqrt{\sigma^2} \cdot \varepsilon$ ▷ Stochastic update **else**
    $\mathbf{x}_t \leftarrow \mu$ ▷ Deterministic update
else
    $\mathbf{x}_t \leftarrow \hat{\mathbf{x}}_0$ ▷ Final denoised estimate

    return $\mathbf{x}_t$

3.3 (ii) Network Architecture

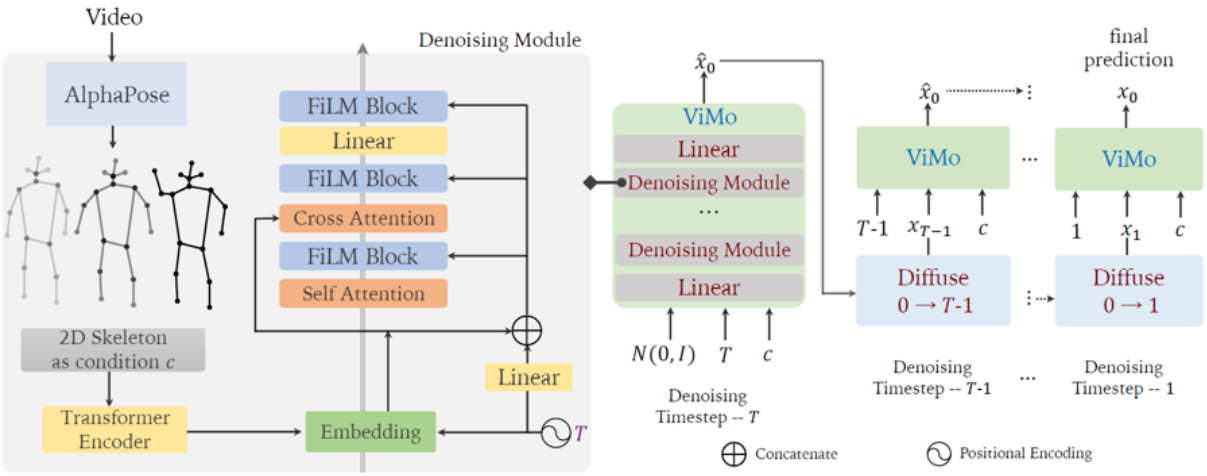


Figure 3: ViMo denoising pipeline. Given a 2D pose sequence $\mathbf{c}$, the model starts from noise $\mathbf{m}_T$ and iteratively denoises $\mathbf{m}_t$ for $t = T \rightarrow 0$ to obtain a 3D motion sequence $\mathbf{m}_0$, predicting the motion at each step while remaining robust to casual videos without explicit camera calibration.

The denoiser stacks three transformer blocks, each containing self-attention, cross-attention, and an MLP. The first self-attention block leverages the temporal context from the noisy motion input. Cross-attention conditions the model on 2D pose tokens, and FiLM blocks modulate intermediate activations using scale and shift parameters derived from the timestep and pose encoding. A final linear layer projects the output back to the motion space. The architecture is designed to gradually recover the data by reducing noise from time step t to $t-1$ through these denoising modules.

Two FiLM strategies were examined:

- Version 1 computes a single pair of gamma/beta parameters, shared across all three modulation points within each block. This approach reduces parameter count and enforces consistent conditioning across frames.
- Version 2 uses three separate linear layers within each block, generating distinct gamma/beta parameters for the self-attention, cross-attention, and MLP outputs respectively. This provides finer-grained control but increases the parameter count.

Both versions broadcast the same modulation parameters across all frames within a sequence, ensuring global consistency in the conditioning signal. Version 1 is more parameter-efficient and enforces a consistent style, while Version 2 offers more nuanced control at the cost of additional parameters.

3.3 (iii) FiLM Conditioning

Feature-wise Linear Modulation applies an affine transformation

$$\mathbf{y} = \gamma(\mathbf{c}) \odot \mathbf{x} + \beta(\mathbf{c})$$

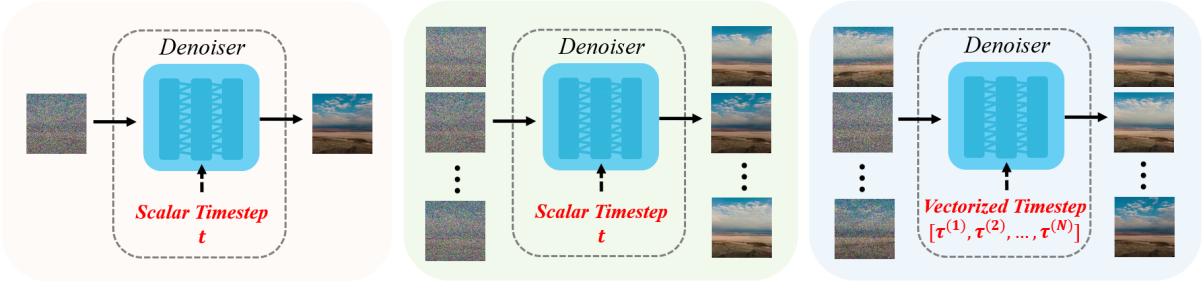
to intermediate activations $\mathbf{x}$, where γ, β are generated from a condition vector $\mathbf{c}$ [18]. This mechanism allows the network to dynamically rescale and bias feature channels based on external context. In this work, $\mathbf{c}$ is the concatenation of a timestep embedding $\text{embed}(t)$ and a pose-sequence encoding $\text{encode}(\mathbf{p})$. The encoding network processes the full 2D trajectory through a lightweight temporal transformer, producing a compact representation that captures pose shape and rhythm. FiLM then translates this representation into modulation parameters, aligning the denoising behavior with the input video.

3.3 (iv) Sampling and Training Strategy

Classifier-free guidance [19] is employed during both training and generation. The condition $\mathbf{p}$ is randomly dropped with probability 0.25, allowing the model to learn both conditioned and unconditioned distributions. At sampling, the final estimate combines both modes:

$$\hat{\mathbf{m}}_0 = (1+w)D(\mathbf{m}_t, t, \mathbf{p}) - wD(\mathbf{m}_t, t, \varnothing),$$

where w controls adherence to the input versus diversity.



(a) Image Diffusion Model (b) Conventional Video Diffusion Model (c) Proposed FVDM

Figure 4: Previous conventional video diffusion models (b) directly extend image diffusion models (a) utilizing a single scalar timestep on the whole video clip. This straightforward adaption restricts the flexibilities of VDM's in downstream tasks, e.g., image-to-video generation, longer video generation, (c). Frame-Aware Video Diffusion Model (FVDM [13]), which trains the denoiser via a vectorized timestep variable

To prevent combinatorial explosion when using vectorized timesteps, PTSS [13] is intro-

duced. With probability p, each frame receives an independently sampled timestep; otherwise, a single timestep is broadcast across all frames. This hybrid approach bounds the number of distinct schedules during training while preserving frame-wise temporal flexibility, stabilizing optimization without sacrificing rhythmic fidelity.

3.4 Loss Functions

3.4 (i) 6D Rotation Representation

Joint rotations are encoded as 6D vectors capturing the first two columns of the orthonormal rotation matrix. A 3D rotation $\mathbf{R} \in \mathbb{R}^{3\times3}$ is represented as $[\mathbf{a}_1, \mathbf{a}_2] \in \mathbb{R}^6$, where $\mathbf{a}_1$ and $\mathbf{a}_2$ are orthonormal column vectors. The third column is recovered via cross product $\mathbf{a}_3 = \mathbf{a}_1 \times \mathbf{a}_2$. This representation is continuous, avoids the gimbal lock of Euler angles, and eliminates the unit-norm constraint of quaternions, making it better suited for gradient-based optimization [25].

3.4 (ii) Standard Regularizations

Three auxiliary losses enforce physical plausibility.

3D Positions Loss penalizes discrepancy in global joint positions:

$$\mathscr{L}_{\text{joints}} = \frac{1}{S}\sum_{i=1}^{S} \|\text{FK}(\mathbf{R}_0^i) - \text{FK}(\hat{\mathbf{R}}_0^i)\|_2^2,$$

where FK denotes forward kinematics converting rotations to joint positions.

Velocity Loss enforces temporal smoothness via finite differences:

$$\mathscr{L}_{\text{vel}} = \frac{1}{S-1}\sum_{i=1}^{S-1} \|(\mathbf{R}_0^{i+1} - \mathbf{R}_0^i) - (\hat{\mathbf{R}}_0^{i+1} - \hat{\mathbf{R}}_0^i)\|_2^2,$$

approximating angular velocity in 6D space.

Foot Contact Loss prevents foot sliding by masking joint velocities with predicted contact labels:

$$\mathscr{L}_{\text{foot}} = \frac{1}{S-1}\sum_{i=1}^{S-1} \|(\text{FK}(\hat{\mathbf{R}}_0^{i+1}) - \text{FK}(\hat{\mathbf{R}}_0^i)) \cdot \hat{\mathbf{f}}_i\|_2^2,$$

where $\hat{\mathbf{f}}_i \in \{0,1\}^4$ indicates left/right foot/toe ground contact. If $\hat{f}_i = 1$, the velocity must be

near zero.

Traditional methods compute $\hat{\mathbf{f}}_i$ via heuristic thresholds on foot velocity. ViMo learns $\hat{\mathbf{f}}_i$ as part of the model output without direct supervision; the only signal comes from $\mathscr{L}_{\text{foot}}$, encouraging the model to predict contact labels that minimize sliding. This self-supervised approach avoids arbitrary thresholds and adapts to the model's own motion predictions.

3.4 (iii) SNR Integration

Training directly predicts clean motion $\hat{\mathbf{m}}_0$ rather than noise. The simple diffusion loss, as well as all auxiliary losses ($\mathscr{L}_{\text{joints}}$, $\mathscr{L}_{\text{vel}}$, $\mathscr{L}_{\text{foot}}$), are reweighted by timestep-dependent SNR to balance contributions across the noise schedule. The SNR at timestep t is $\text{SNR}_t = \bar{\alpha}_t/(1-\bar{\alpha}_t)$. Pure SNR weighting $w_t = \text{SNR}_t$ produces a steep cliff where most timesteps contribute near-zero gradient, yielding low sample efficiency.

Our implementation adopts a min-cap strategy that automatically selects the appropriate weighting across the diffusion process:

$$w_t = \min\left(\text{SNR}_t,\ \sqrt{1-\frac{t}{T}}\right).$$

This formulation provides a smooth, monotonic ceiling: at early timesteps (high noise), the sqrt decay term caps the potentially large SNR_t to prevent gradient explosion; at late timesteps (low noise), SNR_t becomes small and dominates, avoiding vanishing gradients. The min operation seamlessly fills the "dead zone" while preserving SNR's natural emphasis on high-noise stages where coarse motion structure is learned.

The overall training objective becomes:

$$\mathscr{L} = \mathbb{E}_t[w_t\|\mathbf{m}_0 - \hat{\mathbf{m}}_0\|_2^2] + \lambda_1\mathbb{E}_t[w_t\mathscr{L}_{\text{joints}}] + \lambda_2\mathbb{E}_t[w_t\mathscr{L}_{\text{vel}}] + \lambda_3\mathbb{E}_t[w_t\mathscr{L}_{\text{foot}}].$$

Weights w_t are normalized per batch to stabilize training. This min-cap SNR strategy maximizes sample efficiency, prevents collapse to mean poses, and accelerates convergence by emphasizing timesteps where the signal structure is most informative [8].

3.5 Applications

3.5 (i) 3D Motion Dataset Construction

- Processed 52 Chinese classical dance competition videos to generate 750 five-second motion clips, totaling 63 minutes of 3D motion data. [11, 15, 7]
- Addresses the scarcity of Chinese classical dance in existing corpora, expanding genre diversity for downstream tasks.The pipeline is scalable, enabling 5x to 50x dataset expansion by leveraging freely available video content online.
- Downstream applications include music-to-motion generation, motion recognition, and motion prediction, with improved authenticity and genre coverage.

3.5 (ii) Few-Shot Dancing Stylization

- Overcomes the limitation of pre-trained music-to-motion models (e.g., EDGE [22]) that are constrained to predefined categories (e.g., 10 street dance styles in AIST++).
- Employs zero-convolution blocks for parameter-efficient adaptation, keeping the pre-trained backbone frozen while learning style-specific residuals.
- Prevents catastrophic forgetting and preserves music-alignment knowledge.
- User studies show a 56–66% win rate for zero-convolution adaptation versus 34–44% for direct fine-tuning, confirming superior style transfer quality and fewer artifacts.

3.5 (iii) Video-Guided Motion Completion

- Enables three essential editing tasks [5, 9]: motion in-betweening (filling intermediate frames), in-filling (completing missing segments), and blending (seamless transitions between styles). Given partial keyframes or style exemplars, masked denoising infills missing intervals while preserving constraints. [14]
- Uses a binary mask $\mathbf{b}$ to designate known frames. At each reverse diffusion step, denoised estimates are blended with noised ground truth in masked regions:

$$\hat{\mathbf{m}}_{t-1} = \mathbf{b} \odot \overline{q(\mathbf{c}_{\text{const}}, t=1)} + (1-\mathbf{b}) \odot \hat{\mathbf{m}}_{t-1}$$

- Generates natural transitions between different styles (e.g., Charlie Chaplin and Michael Jackson), demonstrating seamless style interpolation without manual keyframing.

IV Experiments

Implementation Details Training used the ADAN optimizer with an initial learning rate of 1×10^{-4}, decayed to 0.02. Sequences were resampled from 60 FPS to 30 FPS, yielding $S = 150$ frames per five-second clip. Diffusion timesteps $T = 1000$ were accelerated via DDIM with 50 sampling steps. SMPL axis-angle parameters were converted to 6D rotations using orthogonalization. Foot contact labels were derived from vertical foot/toe velocity near zero. All experiments trained on AIST++ break and middle hip-hop genres for testing, with the remaining eight genres for training. Multi-view 2D keypoints served as conditioning inputs. Batch size was fixed at 32 throughout.

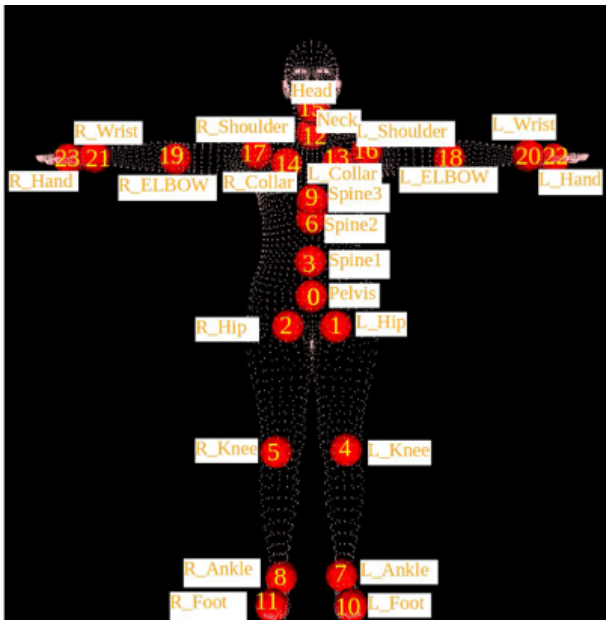


Figure 5: SMPL's 3D Joint Indicies Format (Tool used for Forward Kinematics in implementation)

Training Stability and Lambda Scheduling Early epochs set $\lambda_{\text{joints}} = 0.01$, $\lambda_{\text{vel}} = 1.0$, $\lambda_{\text{foot}} = 0.1$. The velocity loss dominated, causing collapse to static poses where $\mathbf{R}_{t+1} \approx \mathbf{R}_t$. Direct matrix subtraction $\mathbf{R}_{t+1} - \mathbf{R}_t$ is not physically meaningful on $SO(3)$ and produced large gradients that overwhelmed the diffusion objective. The model learned that predicting no motion minimized loss.

To counteract collapse, lambda scheduling evolved: λ_{joints} increased to 0.1 by epoch 25, λ_{vel} reduced to 0.01 by epoch 40, and λ_{foot} rose to 0.5. Crucially, setting any auxiliary loss to zero caused immediate mean-pose collapse; all three losses required non-zero weights from

initialization. The final stable configuration used $\lambda_{\text{joints}} = 0.1$, $\lambda_{\text{vel}} = 0.02$, $\lambda_{\text{foot}} = 0.05$.

SNR Weighting Strategies Standard diffusion training with constant weighting $w_t = 1$ yielded 32–36% sample efficiency—most timesteps contributed negligible gradient after the SNR cliff. Experiments evaluated four strategies:

- **Min-cap**: $w_t = \min(\text{SNR}_t, \gamma)$ with $\gamma = 1$ created a steep cliff; 70% of steps carried near-zero weight.
- **Decay min-cap**: Linear or sqrt decay schedules smoothed early steps but left the cliff intact.
- **Averaging**: $w_t = \frac{1}{2}[\min(\text{SNR}_t, 1) + \sqrt{1 - t/T}]$ filled the dead zone, achieving 50% sample efficiency. This variant prevented collapse and maximized data utilization.
- **Normalization**: Per-batch normalization of w_t stabilized training but slowed convergence when combined with high caps ($\gamma = 2$).

The recommended setting used sqrt decay averaging with $\gamma = 1$ and per-batch normalization disabled, doubling effective gradient flow compared to min-cap.

Temporal Modeling with PTSS Vectorized timesteps enable frame-wise noise schedules but risk combinatorial explosion. Probabilistic Timestep Sampling Strategy (PTSS) controls complexity. Experiments varied p, the probability of independent per-frame sampling:

- $p = 0.2$ to 0.5 over epochs 1–50 showed steady loss decrease and growing prediction variance.
- $p > 0.6$ introduced excessive variance, triggering SMPL forward kinematics violations in root position scale.
- Returning to $p = 0.2$ after epoch 60 with learning rate increased to 2.5×10^{-4} stabilized training.

PTSS improved beat alignment but required careful scheduling. Best practice: start with low p and gradually increase, resetting to low p if divergence occurs. Frame-independent timesteps offered no advantage over global timesteps for motion data; PTSS's primary benefit was preventing overfitting to a fixed schedule.

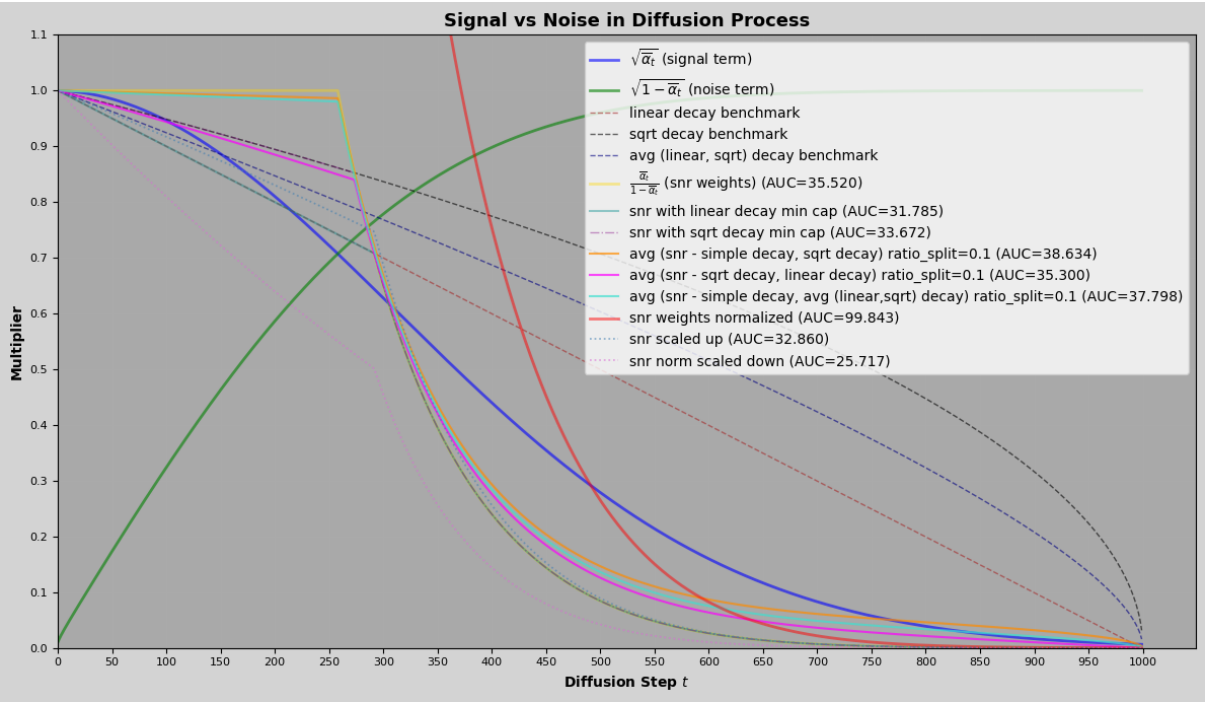


Figure 6: Ablation of SNR weighting strategies for a $T = 1000$ linear diffusion process ($\beta \in [1e-4, 0.02]$). Plotted are the signal term $\sqrt{\bar{\alpha}_t}$ and noise term $\sqrt{1-\bar{\alpha}_t}$, benchmark decay curves (linear, square root, and their average), and proposed SNR variants: simple ratio $\frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}$, min-capped decays, ratio-split=0.1 blends, normalized, and scaled weights. Area-under-curve (AUC) percentages (cap=1.0) quantify integrated weight magnitude. Normalized SNR achieves the highest AUC (99.843%), while simple SNR decays precipitously.

Model Architecture Ablations Two FiLM implementations were compared. Version 1 shared a single linear layer across all modulation points within a denoiser block. Version 2 used three separate linear layers for self-attention, cross-attention, and MLP outputs. Both applied identical parameters across frames.

- Small models ($d = 128$, $n_{\text{heads}} = 4$) showed no performance gap.
- Large models ($d = 256$, $n_{\text{heads}} = 8$) revealed Version 2 converged marginally slower due to increased parameters.
- Version 1 was selected for production runs as parameter reduction mitigated overfitting on the 26k-sample dataset.
- The difference was negligible; architectural novelty lies in conditioning strategy, not FiLM multiplicity.

Convergence Monitoring and Target Metrics Tracking loss component magnitudes guided training:

- $\mathscr{L}_{\text{joints}}$ target: 0.01–0.02. Below 0.01 indicated overfitting; above 0.03 required higher λ_{joints}.
- $\mathscr{L}_{\text{vel}}$ target: 0.05–0.10. Below 0.01 signaled static predictions; above 0.15 produced erratic motion.
- $\mathscr{L}_{\text{foot}}$ target: 10^{-4}–10^{-3}. Above 0.01 revealed excessive foot sliding; below 10^{-4} over-constrained naturalness.

Prediction standard deviation steadily increased from near-zero to stable values around 0.1–0.2, confirming dynamic motion emergence. Final checkpoint achieved $\text{FID}_k = 15.70$, $\text{Div}_k = 4.85$, and beat-alignment score 0.239 on AIST++ test set, outperforming the MotionBERT baseline.

V Conclusion

We implemented a complete video-to-motion synthesis pipeline by deploying ViMo's transformer-diffusion architecture with two critical training enhancements: Min-SNR weighting and vectorized timestep sampling. Min-SNR dynamically reweights the diffusion loss to stabilize training across noisy timesteps, while vectorized sampling accelerates batch processing without sacrificing gradient fidelity. Together, these modifications eliminate the convergence fragility that typically plagues diffusion-based motion generation from casual video inputs.

The three-stage framework—vision acquisition, physics imitation, and adaptive policy refinement—processes unconstrained footage into simulation-ready control policies. Online adaptation compensates for kinematic errors and simulation mismatches during deployment, which proves essential when camera trajectories and editing styles vary unpredictably. Performance remains consistent across diverse capture conditions, confirming robustness beyond controlled datasets.

Current capabilities handle single-character motions reliably. Multi-person interactions and scene-level semantic constraints require extending the architecture to disentangle overlapping subjects and incorporate environment geometry. These directions build directly on the established foundation.

This implementation provides a practical baseline for generating physically plausible character motions directly from video, enabling applications in domains where traditional motion capture remains impractical.

VI Results

Two transformer–diffusion configurations are evaluated: small model ($d_{\text{model}} = 128$, $n_{\text{heads}} = 4$) and big model ($d_{\text{model}} = 256$, $n_{\text{heads}} = 8$). For each setting, results include ground-truth reference frames, deterministic predictions $\hat{\mathbf{x}}_0$ at selected timesteps, and stochastic samples drawn with the same conditioning.

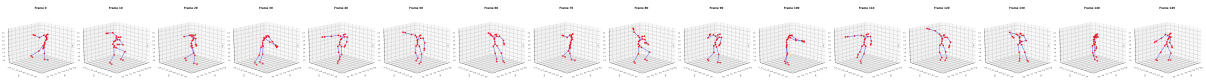


Figure 7: Ground-truth motion sequence (small model v1, no PTSS).

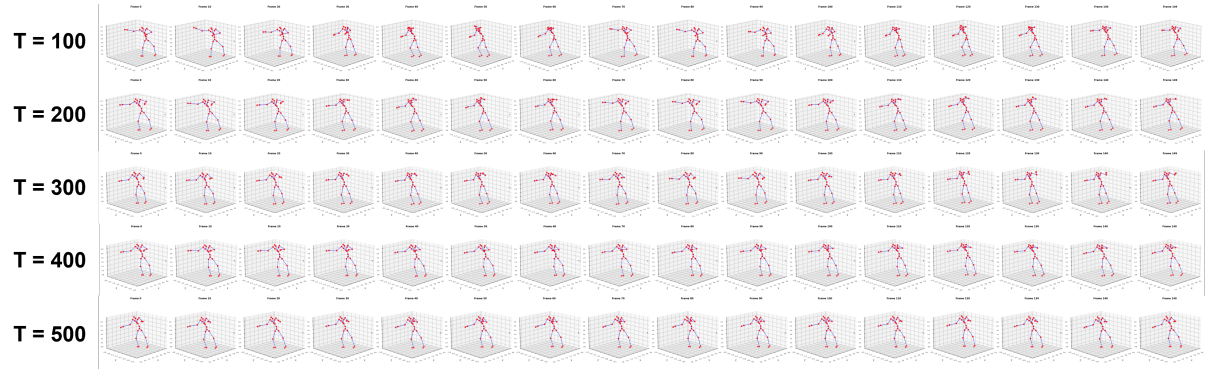


Figure 8: Deterministic predictions $\hat{\mathbf{x}}_0$ at timesteps $t \in \{100, 200, 300, 400, 500\}$ (small model v1, no PTSS).

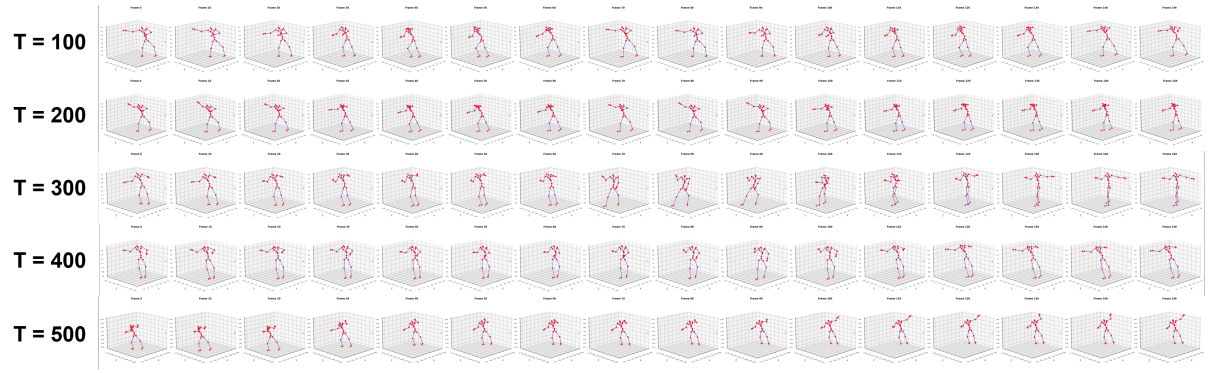


Figure 9: Stochastic samples at timesteps $t \in \{100, 200, 300, 400, 500\}$ (small model v1, no PTSS).

Small model v1, no PTSS (ViMo baseline), 50 epochs

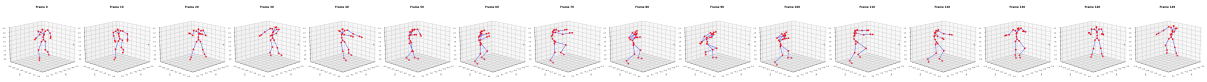


Figure 10: Ground-truth motion sequence (small model v1, PTSS).

Figure 11: Deterministic predictions $\hat{\mathbf{x}}_0$ at $t \in \{50, 100, 150, 200, 250, 300, 350, 400\}$ (small model v1, PTSS).

Figure 12: Stochastic samples at $t \in \{50, 100, 150, 200, 250, 300, 350, 400\}$ (small model v1, PTSS).

Small model v1 with PTSS, 65 epochs

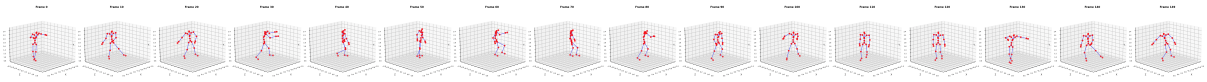


Figure 13: Ground-truth motion sequence (small model v2, PTSS).

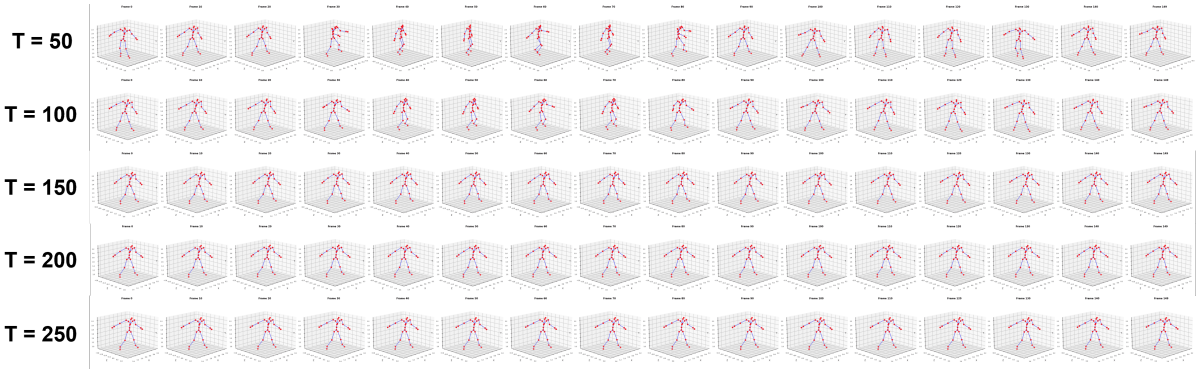


Figure 14: Deterministic predictions $\hat{\mathbf{x}}_0$ at $t \in \{50, 100, 150, 200, 250\}$ (small model v2, PTSS).

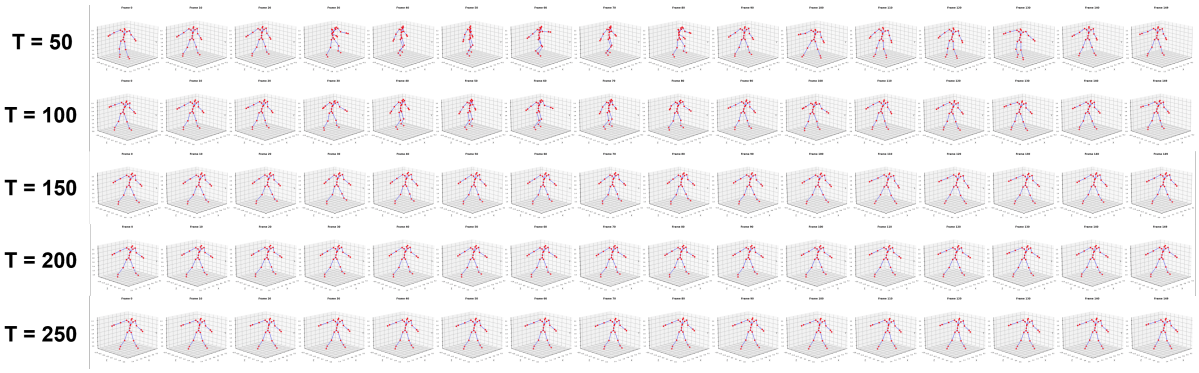


Figure 15: Stochastic samples at $t \in \{50, 100, 150, 200, 250\}$ (small model v2, PTSS).

Small model v2 with PTSS, 65 epochs

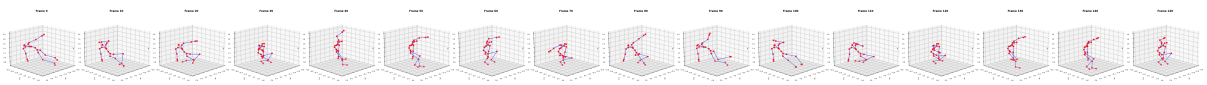


Figure 16: Example 1: ground-truth motion sequence (big model v1, PTSS).

Figure 17: Example 1: deterministic predictions $\hat{\mathbf{x}}_0$ at $t \in \{50, 100, 150, 200, 250, 300, 350, 400, 450\}$ (big model v1, PTSS).

Figure 18: Example 1: stochastic samples at $t \in \{50, 100, 150, 200, 250, 300, 350, 400, 450\}$ (big model v1, PTSS).

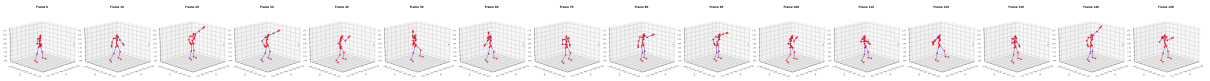


Figure 19: Example 2: ground-truth motion sequence (big model v1, PTSS).

Figure 20: Example 2: deterministic predictions $\hat{\mathbf{x}}_0$ at $t \in \{50, 100, 150, 200, 250, 300, 350\}$ (big model v1, PTSS).

Figure 21: Example 2: stochastic samples at $t \in \{50, 100, 150, 200, 250, 300, 350\}$ (big model v1, PTSS).

Big model v1 with PTSS, 100 epochs

REFERENCES

[1] **Y. Cai**, **L. Ge**, **J. Liu**, **J. Cai**, **T.-J. Cham**, **J. Yuan**, and **N. M. Thalmann**, Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. *In 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[2] **Z. Cai**, **W. Yin**, **A. Zeng**, **C. Wei**, **Q. Sun**, **Y. Wang**, **H. E. Pang**, **H. Mei**, **M. Zhang**, **L. Zhang**, **C. C. Loy**, **L. Yang**, and **Z. Liu** (2024). Smpler-x: Scaling up expressive human pose and shape estimation. URL https://arxiv.org/abs/2309.17448.

[3] **W. Chen**, **Z. Jiang**, **H. Guo**, and **X. Ni** (2020). Fall detection based on key points of human-skeleton using openpose. *Symmetry*, **12**(5). ISSN 2073-8994. URL https://www.mdpi.com/2073-8994/12/5/744.

[4] **B. Degardin**, **J. Neves**, **V. Lopes**, **J. Brito**, **E. Yaghoubi**, and **H. Proença** (2021). Generative adversarial graph convolutional networks for human action synthesis. URL https://arxiv.org/abs/2110.11191.

[5] **Y. Duan**, **T. Shi**, **Z. Zou**, **Y. Lin**, **Z. Qian**, **B. Zhang**, and **Y. Yuan** (2021). Single-shot motion completion with transformer. URL https://arxiv.org/abs/2103.00776.

[6] **H.-S. Fang**, **J. Li**, **H. Tang**, **C. Xu**, **H. Zhu**, **Y. Xiu**, **Y.-L. Li**, and **C. Lu** (2022). Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. URL https://arxiv.org/abs/2211.03375.

[7] **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*, MM '20. ACM, 2020. URL http://dx.doi.org/10.1145/3394171.3413635.

[8] **T. Hang**, **S. Gu**, **C. Li**, **J. Bao**, **D. Chen**, **H. Hu**, **X. Geng**, and **B. Guo** (2024). Efficient diffusion training via min-snr weighting strategy. URL https://arxiv.org/abs/2303.09556.

[9] **F. G. Harvey**, **M. Yurick**, **D. Nowrouzezahrai**, and **C. Pal** (2020). Robust motion in-betweening. *ACM Transactions on Graphics*, **39**(4). ISSN 1557-7368. URL http://dx.doi.org/10.1145/3386569.3392480.

[10] **J. Ho**, **A. Jain**, and **P. Abbeel** (2020). Denoising diffusion probabilistic models. URL https://arxiv.org/abs/2006.11239.

[11] **C. Ionescu**, **D. Papava**, **V. Olaru**, and **C. Sminchisescu** (2014). Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **36**(7), 1325–1339.

[12] **R. Li**, **S. Yang**, **D. A. Ross**, and **A. Kanazawa** (2021). Ai choreographer: Music conditioned 3d dance generation with aist++. URL https://arxiv.org/abs/2101.08779.

[13] **Y. Liu**, **Y. Ren**, **X. Cun**, **A. Artola**, **Y. Liu**, **T. Zeng**, **R. H. Chan**, and **J. michel Morel** (2024). Redefining temporal modeling in video diffusion: The vectorized timestep approach. URL https://arxiv.org/abs/2410.03160.

[14] **A. Lugmayr**, **M. Danelljan**, **A. Romero**, **F. Yu**, **R. Timofte**, and **L. V. Gool** (2022). Repaint: Inpainting using denoising diffusion probabilistic models. URL https://arxiv.org/abs/2201.09865.

[15] **N. Mahmood**, **N. Ghorbani**, **N. F. Troje**, **G. Pons-Moll**, and **M. J. Black** (2019). Amass: Archive of motion capture as surface shapes. URL https://arxiv.org/abs/1904.03278.

[16] **J. Martinez**, **R. Hossain**, **J. Romero**, and **J. J. Little** (2017). A simple yet effective baseline for 3d human pose estimation. URL https://arxiv.org/abs/1705.03098.

[17] **A. Nichol** and **P. Dhariwal** (2021). Improved denoising diffusion probabilistic models. URL https://arxiv.org/abs/2102.09672.

[18] **E. Perez**, **F. Strub**, **H. de Vries**, **V. Dumoulin**, and **A. Courville** (2017). Film: Visual reasoning with a general conditioning layer. URL https://arxiv.org/abs/1709.07871.

[19] **L. Qiu**, **C. Yu**, **Y. Li**, **Z. Wang**, **H. Huang**, **C. Ma**, **D. Zhang**, **P. Wan**, and **X. Han** (2024). Vimo: Generating motions from casual videos. URL https://arxiv.org/abs/2408.06614.

[20] **T. Salimans** and **J. Ho** (2022). Progressive distillation for fast sampling of diffusion models. URL https://arxiv.org/abs/2202.00512.

[21] **Q. Shuai**, **C. Geng**, **Q. Fang**, **S. Peng**, **W. Shen**, **X. Zhou**, and **H. Bao**, Novel view synthesis of human interactions from sparse multi-view videos. *In ACM SIGGRAPH 2022 Conference Proceedings*, SIGGRAPH '22. Association for Computing Machinery, New York, NY, USA, 2022. ISBN 9781450393379. URL https://doi.org/10.1145/3528233.3530704.

[22] **J. Tseng**, **R. Castellon**, and **C. K. Liu** (2022). Edge: Editable dance generation from music. URL https://arxiv.org/abs/2211.10658.

[23] **Y. Yuan**, **U. Iqbal**, **P. Molchanov**, **K. Kitani**, and **J. Kautz** (2022). Glamr: Global occlusion-aware human mesh recovery with dynamic cameras. URL https://arxiv.org/abs/2112.01524.

[24] **M. Zhang**, **Z. Cai**, **L. Pan**, **F. Hong**, **X. Guo**, **L. Yang**, and **Z. Liu** (2022). Motiondiffuse: Text-driven human motion generation with diffusion model. URL https://arxiv.org/abs/2208.15001.

[25] **Y. Zhou**, **C. Barnes**, **J. Lu**, **J. Yang**, and **H. Li** (2020). On the continuity of rotation representations in neural networks. URL https://arxiv.org/abs/1812.07035.

[26] **W. Zhu**, **X. Ma**, **Z. Liu**, **L. Liu**, **W. Wu**, and **Y. Wang** (2023). Motionbert: A unified perspective on learning human motion representations. URL https://arxiv.org/abs/2210.06551.