

# BoDiffusion: Diffusing Sparse Observations for Full-Body Human Motion Synthesis

Angela Castillo<sup>\*1</sup> Maria Escobar<sup>\*1</sup> Guillaume Jeanneret<sup>2</sup> Albert Pumarola<sup>3</sup> Pablo Arbeláez<sup>1</sup>  
Ali Thabet<sup>3</sup> Artsiom Sanakoyeu<sup>3</sup>

<sup>1</sup>Center for Research and Formation in Artificial Intelligence, Universidad de los Andes

<sup>2</sup>University of Caen Normandie, ENSICAEN, CNRS, France

<sup>3</sup>Meta AI

## Abstract

*Mixed reality applications require tracking the user’s full-body motion to enable an immersive experience. However, typical head-mounted devices can only track head and hand movements, leading to a limited reconstruction of full-body motion due to variability in lower body configurations. We propose **BoDiffusion** – a generative diffusion model for motion synthesis to tackle this under-constrained reconstruction problem. We present a time and space conditioning scheme that allows BoDiffusion to leverage sparse tracking inputs while generating smooth and realistic full-body motion sequences. To the best of our knowledge, this is the first approach that uses the reverse diffusion process to model full-body tracking as a conditional sequence generation task. We conduct experiments on the large-scale motion-capture dataset AMASS and show that our approach outperforms the state-of-the-art approaches by a significant margin in terms of full-body motion realism and joint reconstruction error.*

## 1. Introduction

Full-body motion capture enables natural interactions between real and virtual worlds for immersive mixed-reality experiences [17, 38, 51]. Typical mixed-reality setups use a Head-Mounted Display (HMD) that captures visual streams with limited visibility of body parts and tracks the global location and orientation of the head and hands. Adding more wearable sensors [14, 16, 18] is expensive and less comfortable to use. Therefore, in this work, we tackle the challenge of enabling high-fidelity full-body motion tracking when only sparse tracking signals for the head and hands are available, as shown in Fig. 1.

Existing motion reconstruction approaches for 3-point input (head and hands) struggle to model the large variety

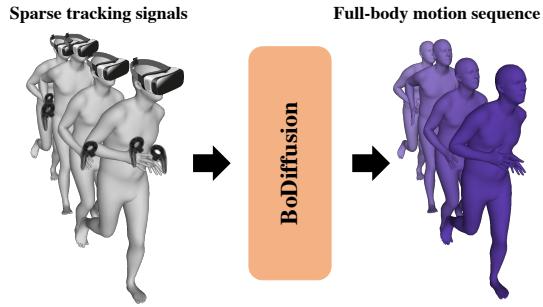
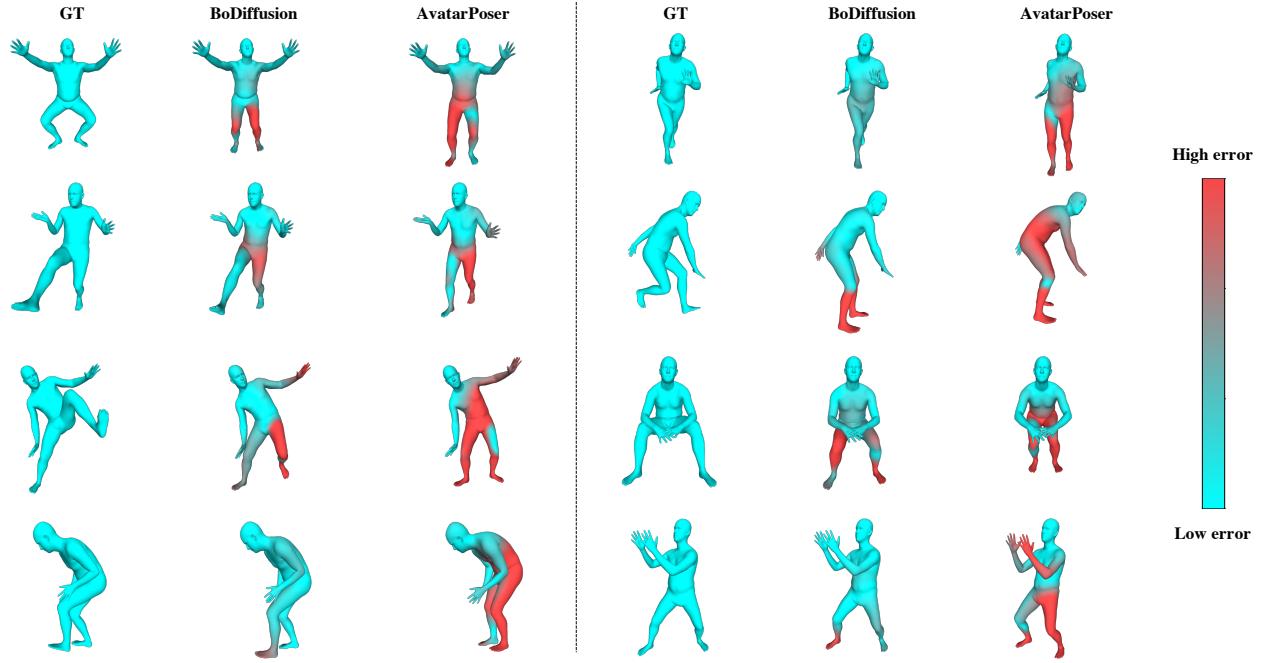


Figure 1. **BoDiffusion.** Head and wrist IMUs are the standard motion-capture sensors in current virtual-reality devices. BoDiffusion leverages the power of Transformer-based conditional Diffusion Models to synthesize fluid and accurate full-body motion from such sparse signals.

of possible lower-body motions and fail to produce smooth full-body movements because of their limited predictive nature [15]. A recent attempt [2] to address this problem uses a generative approach based on normalizing flows [41] falling short of incorporating temporal motion information and generating poses for every frame individually, thus resulting in unrealistic synthesized motions. Another approach [6] that integrates motion history information using a Variational Autoencoder (VAE) [21] takes limited advantage of the temporal history because VAEs often suffer from “posterior collapse” [8, 20]. Thus, there is a need for a scalable generative approach that can effectively model temporal dependencies between poses to address these limitations.

Recently, diffusion-based generative models [11, 46] have emerged as a potent approach for generating data across various domains such as images [42], audio [60], video [12], and language [9]. Compared to Generative Adversarial Networks (GANs), diffusion-based models have demonstrated to capture a much broader range of the target distribution [30]. They offer several advantages, including excellent log-likelihoods and high-quality samples, and em-

<sup>\*</sup> Equal contributions.



**Figure 2. Predicting Dense Full-Body Poses from Sparse Data.** Comparison of BoDiffusion and AvatarPoser [15] against the ground truth. Color gradient in the avatars indicates an absolute positional error, with a higher error corresponding to higher red intensity. BoDiffusion synthesizes substantially more accurate and plausible full-body poses, particularly in the lower body where no IMU data are captured.

ploy a solid, stationary training objective that scales effortlessly with training compute [30].

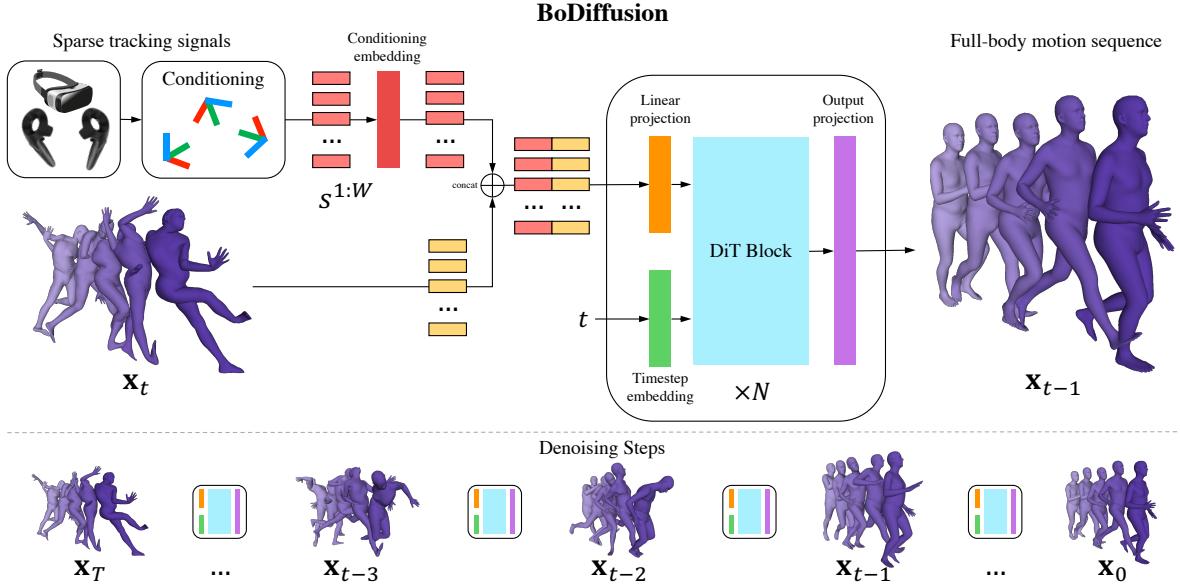
To leverage the powerful diffusion model framework, we propose **BoDiffusion (Body Diffusion)**, a new generative model for human motion synthesis. BoDiffusion directly learns the conditional data distribution of human motions, models temporal dependencies between poses, and generates full *motion sequences*, in contrast to previous methods that operate solely on static poses [2, 55]. Moreover, BoDiffusion does not suffer from the limitation of methods that require a known pelvis location and rotation during inference [2, 6, 55], and generates high-fidelity body motions relying solely on the head and hands tracking information.

Our main contributions can be summarized as follows. We propose BoDiffusion – the first diffusion-based generative model for full-body motion synthesis conditioned on the sparse tracking inputs obtained from HMDs. To build our diffusion model, we adopt a Transformer-based backbone [34], which has proven more efficient for image synthesis than the frequently used UNet backbone [5, 39, 42], and it is more naturally suited for modeling sequential motion data. To enable conditional motion synthesis in BoDiffusion, we introduce a novel time and space conditioning scheme, where global positions and rotations of tracked joints encode the control signal. Our extensive experiments

on AMASS [27] demonstrate that the proposed BoDiffusion synthesizes smoother and more realistic full-body pose sequences from sparse signals, outperforming the previous state-of-the-art methods (see Fig. 2 and 4). Find our full project on [bcv-uniandes.github.io/bodiffusion-wp/](https://bcv-uniandes.github.io/bodiffusion-wp/).

## 2. Related Work

**Pose Estimation from Sparse Observations.** Full-body pose estimation methods generally rely on inputs from body-attached sensors. Much prior work relies on 6 Inertial Measurement Units (IMUs) to predict a complete pose [14, 56, 57]. In [14], the authors train a bi-directional LSTM to predict body joints of a SMPL [25] model, given 6 IMU inputs (head, 2 arms, pelvis, and 2 legs). However, there is a high incentive to reduce the number of body-attached IMUs because depending on many body inputs creates friction in motion capture. LoBSTR [55] reduces this gap by working with 4 inputs (head, 2 arms, and pelvis). It takes past tracking signals of these body joints as input for a GRU network that predicts lower-body pose at the current frame. Furthermore, it estimates the upper body with an Inverse Kinematics (IK) solver. The methods in [2, 6] also require 4 joints as input since they leverage the pose of the pelvis to normalize the input data during training and inference.



**Figure 3. Overview of BoDiffusion.** BoDiffusion is a diffusion process synthesizing full-body motion using sparse tracking signals as conditioning. **Top:** At each denoising step, the model takes as input  $2W$  tokens, which correspond to local joint rotations with  $t$  steps of noise ( $\mathbf{x}_t = x_t^{1:W}$ ) and sparse tracking signals of the head and hands ( $s^{1:W}$ ) as conditioning. We concatenate the  $x_t^i$  tokens with the conditioning tokens  $s^i$  along the spatial axis to preserve the time information and ensure coherence between the conditioning signal and the synthesized motion. After that, we pass it through the Transformer backbone of  $N$  DiT blocks [34]. **Bottom:** During inference, we start from random Gaussian noise  $\mathbf{x}_T$  and perform  $T$  denoising steps until we reach a clean output motion  $\mathbf{x}_0$ .

In Mixed Reality (MR), obtaining user input from a headset and a pair of controllers is common. The authors of [15, 54] highlight the importance of a sensor-light approach and further reduce the amount of inputs to 3, a number that aligns well with scenarios in MR environments. AvatarPoser [15] combines a Transformer architecture and traditional IK to estimate full-body pose from HMD and controller poses. Similar to [15], our method uses only 3 inputs but provides much better lower-body prediction thanks to our diffusion model. Choutas *et al.* [4] propose an iterative neural optimizer for 3D body fitting from sparse HMD signals. However, they optimize poses frame-by-frame and do not consider motions. QuestSim [54] proposes to learn a policy network to predict joint torques and reconstruct full body pose using a physics simulator. Nevertheless, this approach is challenging to apply in a real-world scenario, especially when motion involves interaction with objects (*e.g.*, sitting on a chair). In such a case, one needs to simulate both the human body and all the objects, which have to be pre-scanned in advance and added to the simulation. In contrast, our approach is data-driven and does not require a costly physics simulation or object scanning.

**Human Motion Synthesis & Pose Priors.** A large body of work aims at generating accurate human motion given no past information [1, 23, 36, 37, 58]. Methods like

TEMOS [37] and OhMG [23] combine a VAE [21] and a Transformer network to generate human motion given text prompts. Recently, FLAG [2] argues against the reliability of using VAEs for body estimation and proposes to solve these disadvantages with a flow-based generative model. VPoser [32] learns a pose prior using VAE, and Humor [40] further improves it by learning a conditional prior using a previous pose. Recent work [31] proposes a more generic approach that learns a pose prior and approximates an IK solver using a neural network. Another line of work tackles motion synthesis using control signals provided by an artist or from game-pad input [10, 13, 24, 35, 50]. However, in contrast to our method, such approaches either focus on locomotion and rely on the known future root trajectory of the character or are limited to a predefined set of actions [35].

**Denoising Diffusion Probabilistic Models (DDPMs)** [11, 30] are a class of likelihood-based generative models inspired by Langevin dynamics [22] which map between a prior distribution and a target distribution using a gradual denoising process. Specifically, generation starts from a noise tensor and is iteratively denoised for a fixed number of steps until a clean data sample is reached. Recently, Ho *et al.* [11] have shown [11] that DDPMs are equivalent to the score-based generative models [48, 49]. Currently, DDPMs are showing impressive results in tasks like image genera-

tion and manipulation [5, 7, 29, 39, 42] due to their impressive ability to fit the training distribution at large scale and stable training objective. Moreover, concurrent to this work, Diffusion Models have also been used to synthesize human motion from text inputs [19, 52, 59].

UNet [44] architecture has been de-facto the main backbone for image synthesis with Diffusion Models [5, 39, 42] up until a recent work [34] that suggested a new class of DDPMs for image synthesis with Transformer-based backbones. Transformers are inherently more suitable than convolutional networks for modeling heterogeneous sequential data, such as motion, and we capitalize on this advantage in our work. In particular, we employ a Transformer-based Diffusion Model, based on the DiT backbone [34], to construct an architecture for conditional full-body pose estimation from 3 IMU tracking inputs.

### 3. BoDiffusion

In this section, we present our BoDiffusion model. We start with the DDPMs background in Sect. 3.1. Next, we define the problem statement and our probabilistic framework in Sect. 3.2. Then, in Sect. 3.3, we give an overview of the proposed BoDiffusion model for conditional full-body motion synthesis from sparse tracking signals, followed by the details of our model design. Please refer to Fig. 3 for an illustration of the entire pipeline of our method.

#### 3.1. Diffusion Process

We briefly summarize DDPMs [11] inner workings and formulate our conditional full-body motion synthesis task using the generative framework. Let  $x_0^{1:W} = \mathbf{x}_0 \sim q(\mathbf{x}_0)$  be our real motion data distribution, where  $W$  is the length of the sequence motion. The forward diffusion process  $q$  produces latent representations  $\mathbf{x}_1, \dots, \mathbf{x}_T$  by adding Gaussian noise at each timestep  $t$  with variances  $\beta_t \in (0, 1)$ . Hence, the data distribution is defined as follows:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}) \quad (1)$$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}), \quad (2)$$

where  $\mathbf{I}$  is the identity matrix. Due to the properties of Gaussian distributions, Ho *et al.* [11] showed that we can directly calculate  $\mathbf{x}_t$  from  $\mathbf{x}_0$  by sampling:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \quad (3)$$

where  $\alpha_t = 1 - \beta_t$ ,  $\bar{\alpha}_t = \prod_{i=1}^T \alpha_i$ , and  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ .

On the contrary, the reverse diffusion process  $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$  is the process of iterative denoising through steps  $t = T, \dots, 1$ . Ideally, we would like to perform this process in order to convert Gaussian noise  $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$  back to

the data distribution and generate real data points  $\mathbf{x}_0$ . However,  $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$  is intractable because it needs to use the entire data distribution. Therefore, we approximate it with a neural network  $p_\theta$  with parameters  $\theta$ :

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)). \quad (4)$$

We train to optimize the negative log-likelihood using the Variational Lower Bound (VLB) [11]:

$$\begin{aligned} -\log p_\theta(\mathbf{x}_0) &\leq -\log p_\theta(\mathbf{x}_0) + \\ &+ D_{\text{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0)\|p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)) = \mathcal{L}_{\text{vlb}}. \end{aligned} \quad (5)$$

Following [11], we parameterize  $\mu_\theta(\mathbf{x}_t, t)$  like this:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right). \quad (6)$$

After a couple simplifications, [11] ignores the weighting terms to rewrite  $\mathcal{L}_{\text{simple}}$  as follows:

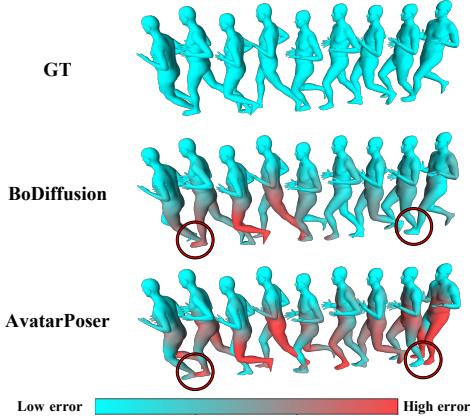
$$\mathcal{L}_{\text{simple}} = E_{\mathbf{x}_0 \sim q(\mathbf{x}_0), t \sim U[1, T]} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2. \quad (7)$$

Ho *et al.* [11] observed that optimizing  $\mathcal{L}_{\text{simple}}$  works better in practice than optimizing full VLB  $\mathcal{L}_{\text{vlb}}$ . During training, we follow Eq. 7, where we sample  $\mathbf{x}_0$  from the data distribution, the timestep as  $t \sim \mathcal{U}\{1, T\}$ , and compute  $\mathbf{x}_t$  using Eq. 3. Intuitively, we learn  $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$  by training neural network to predict the noise  $\epsilon$  that was used to compute the  $\mathbf{x}_t$  with Eq. 3. However, simple loss  $\mathcal{L}_{\text{simple}}$  assumes that we have a predefined variance  $\Sigma(\mathbf{x}_t, t) = \beta_t$ . Instead, we follow [30] and optimize the variance  $\Sigma_\theta(\mathbf{x}_t, t) = e^{v \log \beta_t + (1-v) \log \beta_t^{\frac{1-\bar{\alpha}_t-1}{1-\bar{\alpha}_1}}}$ , where  $v$  is a learnable scalar. Hereby, we use a combined objective:

$$\mathcal{L} = \mathcal{L}_{\text{simple}} + \lambda_{\text{vlb}} \mathcal{L}_{\text{vlb}}. \quad (8)$$

#### 3.2. Conditional Full-Body Motion Synthesis

**Problem Definition.** Human motion can be characterized by a sequence of body poses  $x^i$  ordered in time. We define a *pose* as a set of body joints arranged in the kinematic tree of the SMPL [25] model. Joint states are described by their local rotations relative to their parent joints, with the pelvis serving as the root joint and its rotation being defined with the global coordinate frame. We utilize the 6D representation of rotations [61] to ensure favorable continuity properties, making  $x^i \in \mathbb{R}^{22 \times 6}$ . The global translation of the pelvis is not modeled explicitly, as it can be calculated from the tracked head position by following the kinematic chain [15]. We consider a typical mixed reality system with HMD and two hand controllers that provides 3-point tracking information of head and hands in the form of their global positions  $p^i$  and rotations  $r^i$ . Furthermore, we additionally compute the linear



**Figure 4. Error Comparison.** Comparison of BoDiffusion and AvatarPoser [15] with color coding as previously explained. Motions generated by BoDiffusion exhibit greater similarity to the ground truth and display fewer foot skating artifacts, as highlighted in the red circles. Specifically, the leg in contact with the ground should not slide, and BoDiffusion produces motion sequences that adhere more closely to this requirement.

and angular velocities  $v^i, \omega^i$  of the head and wrists, making  $s^i = \{p^i, r^i, v^i, \omega^i\} \in \mathbb{R}^{3 \times (3+6+3+6)}$  to make the input signal more informative and robust [15]. The target task is to synthesize full-body human motion  $x^{1:W} = \{x^i\}_{i=1}^W$  using the limited tracking signals  $s^{1:W} = \{s^i\}_{i=1}^W$  as input.

**Probabilistic Framework.** We formally define our conditional full-body motion synthesis task by using the formulation of Diffusion Models outlined in Sect. 3.1. Let  $\mathbf{x}_t = x_t^{1:W}, \mathbf{s} = s^{1:W}$  for brevity. We want to learn a conditional distribution of the full-body human motion sequences  $\mathbf{x}_0$  defined as follows:

$$p_\theta(\mathbf{x}_0|\mathbf{s}) = \int p_\theta(\mathbf{x}_{0:T}|\mathbf{s}) d\mathbf{x}_{1:T}, \quad (9)$$

$$p_\theta(\mathbf{x}_{0:T}|\mathbf{s}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{s}), \quad (10)$$

where  $p(\mathbf{x}_T) \sim \mathcal{N}(0, \mathbf{I})$  is a Gaussian noise. In this case, we train a neural network  $\theta$  to predict the mean  $\mu_\theta(\mathbf{x}_t, t, \mathbf{s})$  and the variance  $\Sigma_\theta(\mathbf{x}_t, t, \mathbf{s})$ , similar to Eq. 4, but conditioned on sparse tracking signals  $\mathbf{s}$ . Thus, the simple loss from Eq. 7 then becomes:

$$\mathcal{L}_{\text{simple}} = E_{\mathbf{x}_0 \sim q(\mathbf{x}_0), t \sim U\{1, T\}} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{s})\|_2^2. \quad (11)$$

**Local Rotation Loss is Equivalent to the  $\mathcal{L}_{\text{simple}}$ .** In Human Motion Synthesis, it is widespread [2, 6, 15, 55] to use the local rotation loss that minimizes the difference between the local joint rotations of the estimated poses and

the ground truth. Because of this standard practice, one can hypothesize whether learning  $\epsilon_\theta$  (from Eq. 7) is helpful for synthetic motion sequences. However, we found that optimizing  $\epsilon_\theta$  is equivalent to directly minimizing the local rotation error.

**Lemma 1.** Let  $\mathcal{L}(x, x') = \|x - x'\|^2$  be the local rotation error loss between a motion sequence  $x$  and  $x'$  be an estimate of  $x$ . Then, optimizing the  $\mathcal{L}_{\text{simple}}$  loss is equivalent to optimizing  $\mathcal{L}$ .

We provide the proof of Lemma 1 in the Supplementary Material.

### 3.3. BoDiffusion Architecture

We draw inspiration from the diffusion models for image synthesis to design a model for learning the conditional distribution  $p_\theta(x_0^{1:W} | s^{1:W})$  of the full-body motion sequences (cf. Eq. 9). Specifically, we choose to leverage the novel Transformer backbone DiT [34] to build the BoDiffusion model because (i) it was shown to be superior for image synthesis task [34] compared to the frequently used UNet backbone [5, 39, 42], and (ii) it is more naturally suited for modeling heterogeneous motion data. Below, we provide a detailed description of our architecture and introduce a method that ensures the conditional generation of motion coherent with the provided sparse tracking signal  $s^{1:W}$ .

In order to leverage the Transformer’s ability to handle long-term dependencies while maintaining temporal consistency, we format the input  $x_t^{1:W}$ , which represents joint rotations over time, as a time-sequence tensor and split it along the time dimension into tokens. We treat each pose  $x_t^i$  as an individual token and combine the feature and joint dimensions into a  $d$ -dimensional vector, where  $d = 22 \times 6$  is the number of joints multiplied by the number of features. This strategy allows us to take advantage of the temporal information and efficiently process the motion sequence.

We implement our BoDiffusion model by extending the DiT architecture of Peebles *et al.* [34] with our novel conditioning scheme. The DiT backbone architecture consists of a stack of encoder transformer layers that use Adaptive Layer Normalization (AdaLN). The AdaLN layers produce the scale and shift parameters from the timestep embedding vector to perform the normalization depending on the timestep  $t$ . Peebles *et al.* [34] input the class labels along with the time embedding to the AdaLN layers to perform class-conditioned image synthesis. However, we empirically demonstrate (see Sect. 4.2) that using the conditioning tracking signal  $\mathbf{s}$  along with the time embedding  $t$  in the AdaLN layers harms the performance of our BoDiffusion model because in this case, we disregard the time information. Therefore, we propose a novel conditioning method that retains the temporal information and allows conditional synthesis coherent with the provided sparse tracking signal.

Method	Jitter	MPJVE	MPJPE	Hand PE	Upper PE	Lower PE	MPJRE	FCAcc ↑
Final IK*	-	59.24	18.09	-	-	-	16.77	-
LoBSTR*	-	44.97	9.02	-	-	-	10.69	-
VAE-HMD*	-	37.99	6.83	-	-	-	4.11	-
AvatarPoser [15]	1.53	28.23	4.20	2.34	1.88	8.06	3.08	79.60
AvatarPoser-Large [15]	1.17	23.98	3.71	2.20	1.68	7.09	<b>2.70</b>	82.30
BoDiffusion (Ours)	<b>0.49</b>	<b>14.39</b>	<b>3.63</b>	<b>1.32</b>	<b>1.53</b>	<b>7.07</b>	<b>2.70</b>	<b>87.28</b>

Table 1. **Comparison with State-of-the-art Methods for Full-Body Human Pose Estimation.** Results on a subset of the AMASS dataset (CMU, BMLrub, and HDM05) for Jitter [ $\text{km}/\text{s}^3$ ], MPJVE [cm/s], MPJPE [cm], Hand PE [cm], Upper PE [cm], Lower PE [cm], MPJRE [deg], and FCAcc [%] (balanced foot contact accuracy) metrics. AvatarPoser is retrained with 3 and 10 (Large) Transformer layers. The star (\*) denotes the results reported in [15].

**Conditioning on tracking signal.** We use the 3-point tracking information of head and hands from HMDs to compute an enriched input conditioning  $s^{1:W}$ . This conditioning  $s^{1:W}$  has the shape  $W \times d_s$ , where  $d_s = 18 \cdot 3$  is the number of features (18) per joint multiplied by the number of tracked joints (3). We treat it as a sequence of individual tokens  $s_i$  and apply a linear transformation (*conditioning embedding* layer in Fig. 3) to each of them, thus increasing the dimensionality of the tokens from  $d_s$  to  $d_{emb} = 18 \cdot 22$ . We observe that such higher-dimensional embedding enforces the model to pay more attention to the conditioning signal. Next, we concatenate the input sequence tokens  $x_t^i$  with the transformed conditioning tokens and input the result to the transformer backbone. By preserving the temporal structure of the tracking signal, we enable the model to efficiently learn the conditional distribution of motion where each pose in the synthesized sequence leverages the corresponding sparse tracking signal  $s^i$ .

## 4. Experiments

**Datasets.** We use the AMASS [27] dataset for training and evaluating our models. AMASS is a large-scale dataset that merges 15 optical-marker-based MoCap datasets into a common framework with SMPL [25] model parameters. For our first set of experiments, we use the CMU [3], BMLrub [53], and HDM05 [28] subsets for training and testing. We follow the same splits of AvatarPoser [15] to achieve a fair comparison. For our second set of experiments, we evaluate the Transitions [27] and HumanEVA [45] subsets of AMASS and train on the remaining datasets following the protocol described in [2].

**Evaluation Metrics.** We report four different types of metrics to evaluate our performance comprehensively. First, we report the velocity-related metrics Mean Per Joint Velocity Error [cm/s] (MPJVE), and Jitter error [ $\text{km}/\text{s}^3$ ] [57] that measure the temporal coherence and the smoothness of the generated sequences. Second, we report the position-related metrics Mean Per Joint Position Error [cm] (MPJPE), Hand Position Error [cm] (Hand PE), Upper Body Position Error

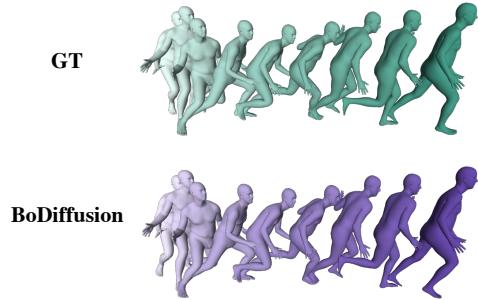


Figure 5. **Full-Sequence Generation.** Example prediction of BoDiffusion compared against the ground-truth sequence. Our method can generate realistic motions faithful to the ground truth. Color gradient represents time flow, whereas lighter colors denote the past.

[cm] (Upper PE), and Lower Body Position Error (Lower PE). The third set is rotation-related metrics, including the Mean Per Joint Rotation Error [deg] (MPJRE). Finally, we devise a metric based on Foot Contact (FC) to measure if the predicted body has a realistic movement of the feet. To calculate this metric for every pair of instances in a sequence, we determine if there is contact between the four joints of the feet and the ground by calculating the velocity of the joints and checking whether it is under a pre-defined threshold or not, following [52]. Afterward, we calculate the accuracy between the predicted and the ground-truth FC. Since the ratio of foot contact vs. foot in the air is meager, we calculate a balanced accuracy (FCAcc).

**Implementation Details.** Similar to [15], we set window size  $W = 41$ . Our Transformer backbone consists of 12 DiT blocks [34]. Before feeding to the backbone, the input tokens are projected to the hidden dimension  $emb = 384$ , as shown in Fig. 3. Finally, we project the output of the last DiT block back to the human body pose space of shape  $41 \times 6 \cdot 22$ , representing the 6D rotations for 22 body joints. During training, we use  $\lambda_{vib} = 1.0$ , and define  $t$  to vary between  $[1, T]$ , where  $T = 1000$  corresponds to a pure Gaussian distribution. At inference, we start from

Method	Jitter	MPJVE	MPJPE	Hand PE	Upper PE	Lower PE	MPJRE	FCAcc ↑
VAE-HMD (3p + pelvis)*	-	-	7.45	-	3.75	-	-	-
VPoser-HMD (3p + pelvis)*	-	-	6.74	-	1.69	-	-	-
HuMoR-HMD (3p + pelvis)*	-	-	5.50	-	1.52	-	-	-
ProHMR-HMD (3p + pelvis)*	-	-	5.22	-	1.64	-	-	-
FLAG [2] (3p + pelvis)*	-	-	<b>4.96</b>	-	<b>1.29</b>	-	-	-
AvatarPoser [15] (3p)	1.11	34.42	6.32	3.03	2.56	12.60	4.64	71.46
BoDiffusion (Ours) (3p)	<b>0.35</b>	<b>21.37</b>	5.78	<b>1.94</b>	2.27	<b>11.55</b>	<b>4.53</b>	<b>82.04</b>

Table 2. **Comparison against Generative-based Models.** Results reported on the held-out Transitions [27] and HumanEVA [45] subset of AMASS, following the protocol of FLAG [2], for Jitter [ $\text{km/s}^3$ ], MPJVE [cm/s], MPJPE [cm], Hand PE [cm], Upper PE [cm], Lower PE [cm], MPJRE [deg], and FCAcc [%] metrics. We report the results after retraining AvatarPoser, and report the same results as in [2] for methods with a star (\*).

pure Gaussian noise, and we use DDIM sampling [47] with 50 steps. We set the variance  $\Sigma_\theta$  of the reverse noise to zero. This configuration turns the model into a deterministic mapping from Gaussian noise to motions, allowing it to do much fewer denoising steps without degrading the quality of synthesized motions.

We use AdamW optimizer [26] with a learning rate of  $1e - 4$ , batch size of 256, without weight decay. Our model has 22M parameters and is trained for 1.5 days on four NVIDIA Quadro RTX 8000. More implementation details are in the Supplementary Material (Sec. A.1).

Our approach has no limitations concerning the length of the generated sequences. We can synthesize motions of arbitrary length by applying BoDiffusion in an autoregressive manner using a sliding window over the input data. We refer the reader to the Supplementary Material for more explanation of our inference-time protocol (see Sec. A.2).

## 4.1. Results

We compare BoDiffusion with AvatarPoser [15] and FLAG [2] following their experimental setups. For AvatarPoser in Table 1, we use the official source code to retrain the standard version with 3 Transformer layers. Furthermore, to ensure a fair comparison with BoDiffusion, we train a scaled-up version of AvatarPoser (AvatarPoser-Large) with 10 layers, 8 attention heads, and an embedding dimension of 384. Since the other state-of-the-art methods do not provide public source codes, we compare them against the results reported in each of the previous papers.

Table 1 shows that BoDiffusion outperforms the state-of-the-art approaches in all metrics on the test subset of the AMASS dataset (CMU, BMLrub, and HDM05). Since we enforce the temporal consistency in BoDiffusion by leveraging the novel conditioning scheme and learning to generate sequences of poses instead of individual poses, our method generates smoother and more accurate motions. This is demonstrated by our quantitative results in Tab. 1. We observe a significant improvement in the quality of generated motions by leveraging the BoDiffusion model. Thus, we are able to decrease the MPJVE by a margin of

9.59 cm/s and the Jitter error by 0.68  $\text{km/s}^3$ , compared to AvatarPoser-Large. Fig. 4 shows that motions generated by BoDiffusion exhibit more significant similarity to the ground truth across all the sequence frames and display fewer foot-skating artifacts compared to AvatarPoser, which struggles to maintain coherence throughout the sequence and severely suffers from foot skating. Furthermore, we empirically demonstrate that our method successfully learns a manifold of plausible human poses while maintaining temporal coherence. In practice, we are given the global position of the hands and head as the conditioning; thus, it is expected to have a lower error on these joints, while the conditioning does not uniquely define the configuration of legs and should be synthesized. However, Fig. 2, 4, 5 show that BoDiffusion produces plausible poses not only for the upper body but for the lower body as well, in contrast to the state-of-the-art Transformer-based AvatarPoser method.

Fig. 2 qualitatively shows the improvement of our method in positional errors. In particular, our method predicts lower body configurations that resemble the ground truth more than AvatarPoser. These results support the effectiveness of our conditioning scheme for guiding the generation towards realistic movements that are in close proximity to the ground-truth sequences.

Furthermore, our method achieves a better performance in the Foot Contact Accuracy metric (FCAcc), as shown in Table 1 and the feet movements in Fig. 5. Thus, the iterative nature of the DDPMs, along with our spatio-temporal conditioning scheme, allows us to generate sequences with high fidelity even at the feet, which are the furthest from the input sparse tracking signals.

Table 1 shows the performance of a larger version of AvatarPoser-Large compared to ours. In particular, we demonstrate that enlarging this model increases its motion capture capacity to the point where it reaches more competitive results. By definition, this experiment also demonstrates that using more complex methods leads to better performance. However, BoDiffusion depicts a better trade-off between the performance and computational complexity than state-of-the-art methods. Since BoDiffusion can take

Method	Jitter	MPJVE	MPJPE	MPJRE
BoDiffusion (Token input cond)	<b>0.49</b>	<b>14.39</b>	3.63	2.70
Timestep cond	1.38	52.78	7.19	4.00
Token input + Timestep cond with stochasticity	0.59 0.53	16.22 15.37	3.60 <b>3.53</b>	<b>2.60</b> 2.67
Window size W=1	19.71	174.9	4.77	3.13
Shuffled sequences	108.42	935.69	17.13	7.10

Table 3. **Design Ablations.** **Up:** We ablate our training scheme by varying the conditioning approach. At inference, we demonstrate that controlling the stochasticity smoothens our predictions. **Down:** We assess the importance of including temporal context.

Method	Jitter	MPJVE	MPJPE	MPJRE
UNet w/o diffusion	1.44	33.35	4.36	2.81
Transformer w/o diffusion	1.27	27.62	3.92	2.60
BoDiffusion-UNet	1.24	20.65	<b>3.63</b>	<b>2.48</b>
BoDiffusion-Transformer (Ours)	<b>0.49</b>	<b>14.39</b>	<b>3.63</b>	2.70

Table 4. **Architecture Ablations.** We evaluate the relevance of using DiT as our backbone. We also assess the effectiveness of the denoising power of our DDPM by comparing it against the backbones without diffusion.

DDIM steps	Jitter	MPJVE	MPJPE	MPJRE
10	0.56	16.16	3.89	2.84
20	0.52	15.05	3.72	2.75
30	0.51	14.75	3.66	2.73
40	0.49	14.55	3.64	2.71
50	0.49	14.39	3.63	2.70
100	0.48	14.12	3.44	2.59

Table 5. **Ablation of inference sampling steps.** During inference, we use DDIM sampling with 50 steps. Note that the performance improves when there are more sampling steps.

advantage of DiT, our approach will further improve in the measure that foundation models reach better results.

Table 2 shows the quantitative comparison between BoDiffusion and other generative-based state-of-the-art approaches for the Transitions [27] and HumanEVA [45] subsets of AMASS. AvatarPoser is included for reference. On the one hand, even though we only train with three sparse inputs, we have competitive results regarding an overall positional error (MPJPE) and upper body positional error (Upper PE) with the methods that also use the pelvis information. Our DDPM-based method outperforms the VAE-based approaches VAE-HMD and VPoser-HMD and has comparable results with the conditional flow-based models ProHMR-HMD and FLAG. On the other hand, our BoDiffusion has a better performance than AvatarPoser in all the metrics, with a significant improvement in the velocity-related metrics MPJVE and Jitter. Please refer to the Supplementary for additional qualitative results.

## 4.2. Ablation Experiments

We conduct ablation experiments to assess the effect of the different components of our method on the smoothness and temporal consistency of the generated sequences. In Table 3, we report the experiments corresponding to the conditioning scheme, the stochastic component at inference, and the relevance of temporal context. Firstly, we compare the effect of using different conditioning schemes. Our method receives the conditioning by concatenating the input tokens (Token input cond). Thus, the conditioning keeps time-dependent information, allowing us smoother predictions, as the low Jitter and MPJVE values show. In contrast, applying the condition through the timestep embedding (Timestep cond) results in a compression towards a time-agnostic vector embedding. Table 3 shows that using this time-agnostic embedding solely as conditioning results in detrimental performance for the method. Furthermore, using both the token input and the timestep conditioning still results in less smooth sequences and is less consistent than using only the token input conditioning scheme.

Secondly, we implement a purely stochastic inference scheme (w/ stochasticity), finding out that, even when the rotational and positional errors decrease slightly, having extra control over the randomness is beneficial, especially for the smoothness of the sequences, as shown by the decrease in MPJVE and Jitter. Thirdly, we evaluate the importance of having temporal consistency by using a sliding window of size one during training (Window size W=1) and randomly sorting the sequence at inference time (Unordered sequence). As expected, the MPJVE and Jitter errors increase significantly, and all the other metrics also increase by some proportion. Therefore, these experiments confirm the relevance of enforcing temporal consistency.

Table 4 presents the impact of different architectural choices on the performance of our proposed model. First, to validate the effectiveness of using DiT as our backbone (BoDiffusion-Transformer), we compare it against UNet (BoDiffusion-UNet), which has traditionally been used as a backbone for diffusion models [5, 42]. Table 4 indicates that the Transformer outperforms UNet in all the metrics, even when diffusion processes are not involved. Additionally, when incorporating our diffusion framework on top of both backbones, significant improvements are observed in the temporal consistency and quality of the generated sequences. It is important to note that while replacing the DiT backbone with UNet leads to a slight decrease ( $0.2^\circ$ ) in rotation error, it is accompanied by a significant increase in Jitter and Velocity errors. Thus, these ablation experiments demonstrate the complementarity of using a transformer-based backbone in a diffusion framework, resulting in smoother and more accurate predictions.

Table 5 shows the ablation experiment using different sampling steps for DDIM at inference time. Increasing the

sampling steps improves the performance of our method, proving the importance of the iterative nature of DDPMs. However, more steps require more computational capacity. Thus, we select 50 DDIM steps for an appropriate trade-off between performance and complexity.

## 5. Conclusion

In this work, we present BoDiffusion, a Diffusion model for conditional motion synthesis inspired by effective architectures from the image synthesis field. Our model leverages the stochastic nature of DDPMs to produce realistic avatars based on sparse tracking signals of the hands and head. BoDiffusion uses a novel spatio-temporal conditioning scheme and enables motion synthesis with significantly reduced jittering artifacts, especially on lower bodies. Our results outperform state-of-the-art methods on traditional metrics, and we propose a new evaluation metric to fully demonstrate BoDiffusion’s capabilities.

## Acknowledgements

Research reported in this publication was supported by the Agence Nationale pour la Recherche (ANR) under award number ANR-19-CHIA-0017.

## References

- [1] Hyemin Ahn, Timothy Ha, Yunho Choi, Hwiyeon Yoo, and Songhwai Oh. Text2action: Generative adversarial synthesis from language to action. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5915–5920. IEEE, 2018. [3](#)
- [2] Sadegh Aliakbarian, Pashmina Cameron, Federica Bogo, Andrew Fitzgibbon, and Thomas J Cashman. Flag: Flow-based 3d avatar generation from sparse observations. In *CVPR*, pages 13253–13262, 2022. [1, 2, 3, 5, 6, 7](#)
- [3] Carnegie Mellon University. CMU MoCap Dataset. [6, 14](#)
- [4] Vasileios Choutas, Federica Bogo, Jingjing Shen, and Julien Valentin. Learning to fit morphable models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*, pages 160–179. Springer, 2022. [3](#)
- [5] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. [2, 4, 5, 8, 13](#)
- [6] Andrea Dittadi, Sebastian Dziadzio, Darren Cosker, Ben Lundell, Thomas J Cashman, and Jamie Shotton. Full-body motion from a single head-mounted device: Generating smpl poses from partial observations. In *ICCV*, pages 11687–11697, 2021. [1, 2, 5](#)
- [7] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *ECCV*, 2022. [4](#)
- [8] Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. In *International Conference on Learning Representations*, 2018. [1](#)
- [9] Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. *arXiv preprint arXiv:2211.15029*, 2022. [1](#)
- [10] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. [3](#)
- [11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. [1, 3, 4, 14](#)
- [12] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022. [1](#)
- [13] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017. [3](#)
- [14] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM TOG*, 37(6):1–15, 2018. [1, 2](#)
- [15] Jiaxi Jiang, Paul Streli, Huajian Qiu, Andreas Fender, Larissa Laich, Patrick Snape, and Christian Holz. Avataposer: Articulated full-body pose tracking from sparse motion sensing. In *Proceedings of European Conference on Computer Vision*. Springer, 2022. [1, 2, 3, 4, 5, 6, 7, 12, 14, 15, 16, 17](#)
- [16] Yifeng Jiang, Yuting Ye, Deepak Gopinath, Jungdam Won, Alexander W Winkler, and C Karen Liu. Transformer inertial poser: Attention-based real-time human motion reconstruction from sparse imus. *arXiv preprint arXiv:2203.15720*, 2022. [1](#)
- [17] Brennan Jones, Yaying Zhang (yaying zhang), Priscilla N. Y. Wong, and Sean Rintel. Belonging there: Vroom-ing into the uncanny valley of xr telepresence. In *CSCW 2021*. ACM, April 2021. [1](#)
- [18] Manuel Kaufmann, Yi Zhao, Chengcheng Tang, Lingling Tao, Christopher Twigg, Jie Song, Robert Wang, and Otmar Hilliges. Em-pose: 3d human pose estimation from sparse electromagnetic trackers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11510–11520, 2021. [1](#)
- [19] Jihoon Kim, Jiseob Kim, and Sungjoon Choi. Flame: Free-form language-based motion synthesis & editing. *arXiv preprint arXiv:2209.00349*, 2022. [4](#)
- [20] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016. [1](#)
- [21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. [1, 3](#)
- [22] Paul Langevin. Sur la théorie du mouvement brownien. *Compt. Rendus*, 146:530–533, 1908. [3](#)

- [23] Junfan Lin, Jianlong Chang, Lingbo Liu, Guanbin Li, Liang Lin, Qi Tian, and Chang-wen Chen. Ohmg: Zero-shot open-vocabulary human motion generation. *arXiv preprint arXiv:2210.15929*, 2022. 3
- [24] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. Character controllers using motion vaes. *ACM Transactions on Graphics (TOG)*, 39(4):40–1, 2020. 3
- [25] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 2, 4, 6, 12
- [26] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 7
- [27] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, Oct. 2019. 2, 6, 7, 8, 14, 16, 17
- [28] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database HDM05. Technical Report CG-2007-2, Universität Bonn, June 2007. 6, 14
- [29] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 4
- [30] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 1, 2, 3, 4
- [31] Boris N Oreshkin, Florent Bocquelet, Felix G Harvey, Bay Raitt, and Dominic Laflamme. Protos: Proto-residual network for pose authoring via learned inverse kinematics. In *International Conference on Learning Representations*, 2021. 3
- [32] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019. 3
- [33] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 12
- [34] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 2, 3, 4, 5, 6, 13
- [35] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (TOG)*, 40(4):1–20, 2021. 3
- [36] Mathis Petrovich, Michael J Black, and Gü̈l Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021. 3
- [37] Mathis Petrovich, Michael J Black, and Gü̈l Varol. Temos: Generating diverse human motions from textual descriptions. *arXiv preprint arXiv:2204.14109*, 2022. 3
- [38] Thammathip Piumsomboon, Gun A Lee, Jonathon D Hart, Barrett Ens, Robert W Lindeman, Bruce H Thomas, and Mark Billinghurst. Mini-me: An adaptive avatar for mixed reality remote collaboration. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–13, 2018. 1
- [39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2, 4, 5
- [40] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. Humor: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11488–11499, 2021. 3
- [41] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015. 1
- [42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 2, 4, 5, 8
- [43] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017. 12
- [44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4
- [45] L. Sigal, A. Balan, and M. J. Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(4):4–27, Mar. 2010. 6, 7, 8, 14, 16, 17
- [46] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1
- [47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 7, 13
- [48] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 3
- [49] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020. 3, 13

- [50] Sebastian Starke, Ian Mason, and Taku Komura. Deepphase: periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 3
- [51] Franco Tecchia, Leila Alem, and Weidong Huang. 3d helping hands: a gesture based mr system for remote collaboration. In *Proceedings of the 11th ACM SIGGRAPH international conference on virtual-reality continuum and its applications in industry*, pages 323–328, 2012. 1
- [52] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 4, 6
- [53] Nikolaus F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision*, 2(5):2–2, Sept. 2002. 6, 14
- [54] Alexander Winkler, Jungdam Won, and Yuting Ye. Questsim: Human motion tracking from sparse sensors with simulated avatars. *ACM TOG*, 2022. 3
- [55] Dongseok Yang, Doyeon Kim, and Sung-Hee Lee. Lobstr: Real-time lower-body pose prediction from sparse upper-body tracking signals. In *Comput. Graph. Forum*, volume 40, pages 265–275. Wiley Online Library, 2021. 2, 5
- [56] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. Physical inertial poser (pip): Physics-aware real-time human motion tracking from sparse inertial sensors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 2
- [57] Xinyu Yi, Yuxiao Zhou, and Feng Xu. Transpose: Real-time 3d human translation and pose estimation with six inertial sensors. *ACM Transactions on Graphics*, 40(4), 08 2021. 2, 6
- [58] Andrei Zanfir, Eduard Gabriel Bazavan, Hongyi Xu, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Weakly supervised 3d human pose and shape reconstruction with normalizing flows. In *European Conference on Computer Vision*, pages 465–481. Springer, 2020. 3
- [59] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 4
- [60] Kong Zhifeng, Wei Ping, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *International Conference on Learning Representations.*, 2021. 1
- [61] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, pages 5745–5753, 2019. 4

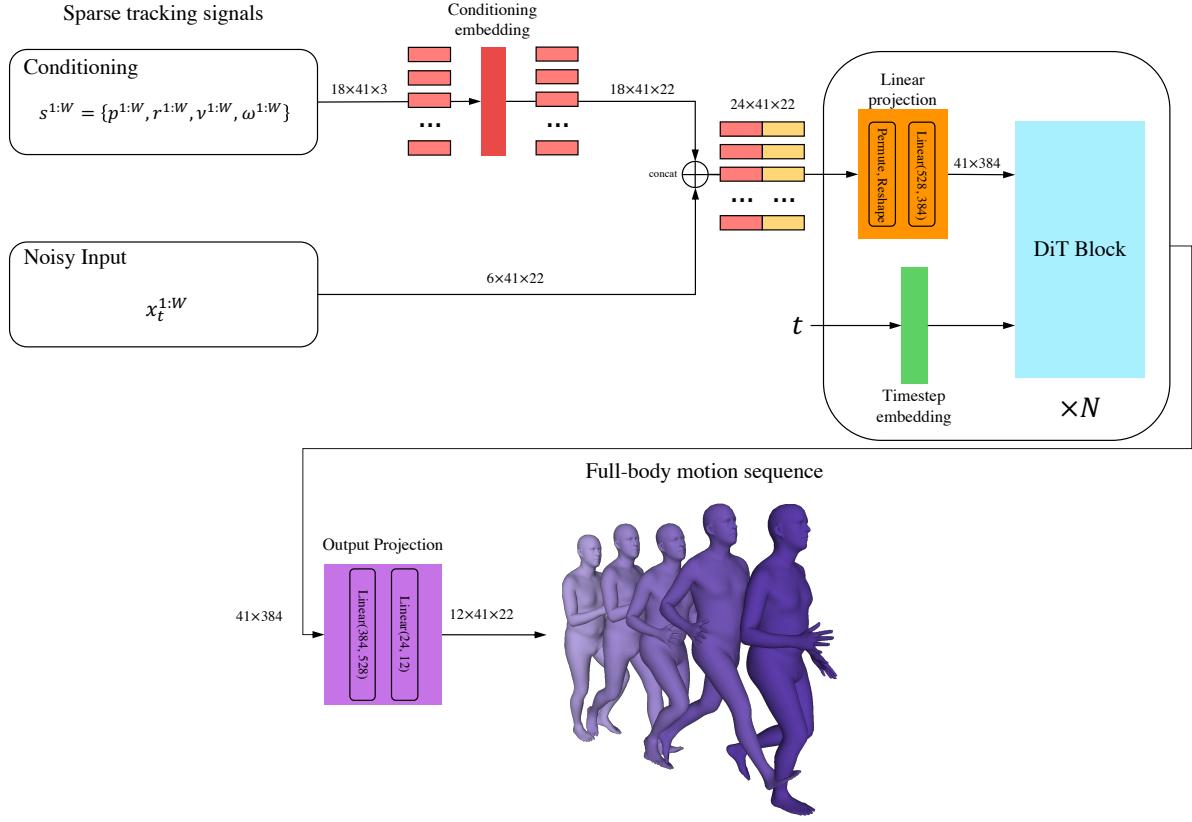


Figure 6. **Complete Overview of BoDiffusion.** Our conditional model takes full advantage of sparse information since we calculate relevant features in the conditioning pathway at the top (red block). In the case of the noisy input, we do not need any projection to match the sizes from the conditioning pathway. After concatenating both pathways, we organize the tensors’ dimensions and perform an additional projection. The linear projection changes the tensor dimensions to the embedding dimension for the DiT blocks. After denoising by the DiT, we perform a final projection to the original space of full body motions (purple block). The output estimates  $\epsilon_\theta$  and  $\Sigma_\theta$  that are used to compute the local rotations  $x_{t-1}^{1:W}$  by sampling from  $\mathcal{N}(x_{t-1}; \mu_\theta, \Sigma_\theta)$ . Here  $W = 41$  is the temporal window size, conditioning signal  $s^{1:W}$  contains 18D features for the three tracked joints (head and hands), and  $x_t^{1:W}$  is the noisy local 6D rotations for 22 body joints.  $\oplus$  is the operation of concatenation along the channels’ dimension. The numbers next to the arrows denote the input and output dimensions for the corresponding blocks.

## A. Implementation Details

We build upon the SMPL [25, 33, 43] parametric model that uses local rotations in axis-angle representation to produce a full-body pose. Our model predicts local rotations in 6D representation that are then converted to axis-angle representation to be used in the body model from SMPL. As in [15], we use a neutral body model corresponding to the average body model between women and men. We do not apply normalization to the conditioning signal  $s^{1:W}$  before inputting it into the model.

## A.1. Architecture

In Figures 6 and 7, we provide further information on the architecture of our model and technical details. In Figure 6, we show the projection of the input condition (red block), which corresponds to the joint positions  $p^{1:W}$ , rotations  $r^{1:W}$ , linear velocities  $v^{1:W}$ , and angular velocities  $\omega^{1:W}$  in the global coordinate frame. This projection aims to change the feature dimension of the conditioning input. It is worth saying that the input  $x_t$  corresponds to a noisy input at time  $t$ . After denoising with the DiT, we perform a final projection (Figure 6, in purple) to map back into the

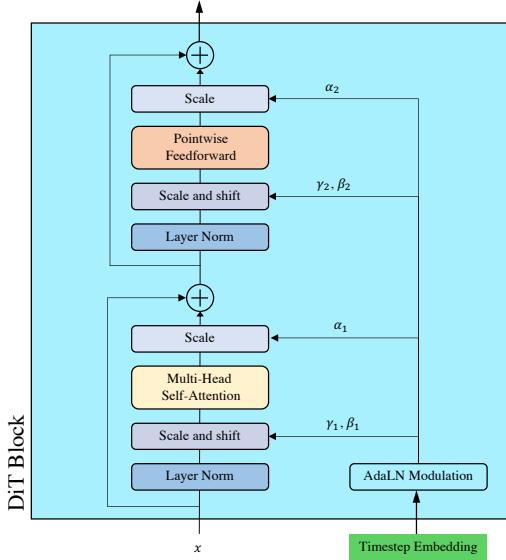


Figure 7. **BoDiffusion Architecture.** Our denoising model is built using multiple DiT blocks. Here we show the details of a single DiT block.

space of motions represented by 6D local rotations of joints. We return a 12-channel tensor which contains predictions of  $\epsilon_\theta$  and  $\Sigma_\theta$  (6 channels each) that are used to compute losses  $\mathcal{L}_{\text{simple}}$  and  $\mathcal{L}_{\text{vib}}$ .

Figure 7 presents a detailed scheme of our DiT architecture for the denoising process. The DiT network starts with a Layer Normalization followed by an adaptive normalization that uses the timestep embedding. This adaptive normalization consists of an MLP that learns regression values that come from the embedding vectors of the timestep instead of learning the modulation parameters  $\gamma$  and  $\beta$  parameters from the data. Afterward, we use six attention heads in the self-attention and perform one more scaling from which values come from the adaptive normalization. We apply a residual connection between the scaling’s input and output. Then, we repeat the normalization stages, but instead of having another attention mechanism, we use the typical point-wise feedforward. In the end, we finish with another residual connection, which is a summation. We follow [34, 49] to compute the timestep embedding.

## A.2. Inference

At inference time, we use DDIM [47] with 50 iterations and remove the stochasticity during sampling from the distribution by setting the variance  $\Sigma_\theta$  to zero. Using a sliding window, we use the temporal window size  $W = 41$  and apply BoDiffusion to the input tracking signal from HMD and hand controllers. While during online inference, one would apply our model using a sliding window with stride 1, for the sake of faster inference on AMASS dataset, we apply

our model using a stride of 20 frames. We did not observe the degradation of generations’ quality when we increased the stride.

## A.3. Inference Speed

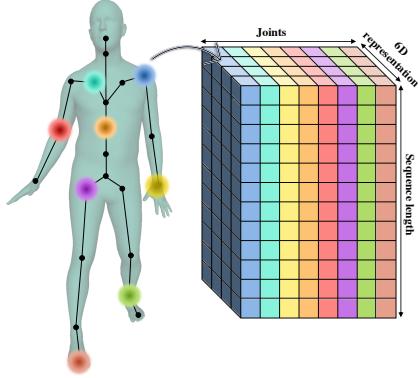
A forward pass with  $W = 41$  takes 0.021 secs for our method and 0.003 secs for AvatarPoser. At inference, we do 50 forward passes that amount to 1.046 secs. Our method is not optimized for speed yet because our goal was to prove that DDPMs can generate high-quality motions. Future work has a huge potential for making DDPM’s inference faster by more efficient sampling, reducing the number of layers and channels, and using quantization.

## A.4. UNet Architecture for Ablation

To ablate the architecture, we also implemented a version of BoDiffusion using the popular DDPM UNet backbone [5] designed for image data and not for motions, the overview of this architecture is shown in Fig. 12. We followed [5] for the architecture’s hyper-parameter selection. In our case, we modified the ImageNet 128-channel architecture but changed the base number of channels from 256 to 64. Furthermore, we kept the same hyper-parameters for the feature dimension multiplication. Considering the motion represented as a sequence of poses  $x^{1:W}$ , we can treat it as a “structured” image tensor (as shown in Fig. 8), such that spatial dimensions (height and width for image) are replaced by “time” and “joints” dimensions and channels are replaced by joint features (in our case it is 6D rotations). A “structured” image in this context means that each pixel of the image represents a single joint located in the kinematic tree along one axis and time along another axis of the tensor. Due to the sufficient depth of the network and a self-attention block in the middle, the effective receptive field of the deepest convolutional layers covers the entire “structured” image.

## B. Additional Ablation Experiment

Table 6 demonstrates that our window size is optimal for this task. First, we empirically show that removing the temporal information from the input leads to high jitter and velocity errors. In practice, using single frames is not enough to enforce temporal consistency, thus making it harder to understand the full-body movement. Therefore, even when the positional and rotational errors are not extremely high compared with our model, the jitter and velocity errors increase considerably, thus misspending the long-range analysis capacity of Transformers. Secondly, we vary the number of input sequences to demonstrate the importance of enforcing temporal consistency. Since our window size is 41, we choose half and double the number of input sequences to assess the benefit of increasing or decreasing the temporal information. As expected, increasing the window size to 81



**Figure 8. Input tensor representation for UNet network.** We represent the motion sequence as a 3D tensor in which the channels correspond to the 6D rotation, the height to the time sequence, and the width to the joints. This representation is analogous to the image data input tensors. In this way, we can reuse the convolutional architectures of the denoising UNet for 3D body pose estimation.

Method	Jitter	MPJVE	MPJPE	MPJRE
Window size $W = 1$	19.71	174.9	4.77	3.13
Window size $W = 21$	0.53	16.09	3.96	2.86
BoDiffusion ( $W = 41$ )	0.49	14.39	<b>3.63</b>	<b>2.70</b>
Window size $W = 81$	<b>0.46</b>	<b>13.69</b>	3.77	2.86

**Table 6. Window Size Ablation.** We evaluate the importance of including more or less temporal context. We report Jitter [km/s<sup>3</sup>], MPJVE [cm/s], MPJPE [cm], and MPJRE [deg].

results in having more temporal coherence, thus decreasing the jitter and velocity errors. However, increasing the input window size also increases the computational cost of training from 1.5 days to almost 3 days. In contrast, reducing the window size to 21 leads to harnessing the smoothness of the motion. It is worth mentioning that even when the jitter and velocity errors are affected by different window sizes, our method performs the best in terms of positional and rotational errors.

## C. Additional Qualitative Evaluation

Figure 9 shows additional qualitative results for BoDiffusion and AvatarPoser [15] on the CMU [3], BMLrub [53], and HDM05 [28] test sets. Notice that our method can generate poses close to the ground truth even when the actions are unusual. For instance, column 2 depicts poses of a person doing movements very close to the ground. Our method is able to use the sparse tracking input for such uncommon motions and predict plausible body configurations that are faithful to the ground truth. In contrast, AvatarPoser struggles with creating accurate poses when seeing an uncommon motion.

Figures 10 and 11 show qualitative results on the Transitions [27] and HumanEVA [45] test sets predicted with BoDiffusion and AvatarPoser [15]. First, note that BoDiffusion generates individual poses with a high fidelity in the upper-body configuration and a plausible lower-body configuration. Second, 11 shows that BoDiffusion captures more details of the position of the feet and avoids foot sliding, unlike AvatarPoser.

To fully appreciate the high quality of the motions generated by our approach, we suggest the reader watch the video attached to this supplementary material. The video demonstrates that BoDiffusion synthesizes more accurate motions with substantially less jitter than AvatarPoser [15].

## D. Local Rotation Loss

Due to the properties of Gaussian distributions, Ho *et al.* [11] showed that we can directly calculate  $\mathbf{x}_t$  from  $\mathbf{x}_0$  by sampling:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \quad (12)$$

and the following simple loss function can be used for network training:

$$\mathcal{L}_{\text{simple}} = E_{x_0 \sim q(x_0), t \sim U[1, T]} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2. \quad (13)$$

In Eq. 12,  $\alpha_t = 1 - \beta_t$ ,  $\bar{\alpha}_t = \prod_{i=1}^T \alpha_i$ ,  $\beta_t$  define the variance schedule for  $t \in \{1, \dots, T\}$ , and  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ .

We found that optimizing  $\epsilon_\theta$  to approximate the noise  $\epsilon$  (Eq. 13) is equivalent to directly minimizing the local rotation error.

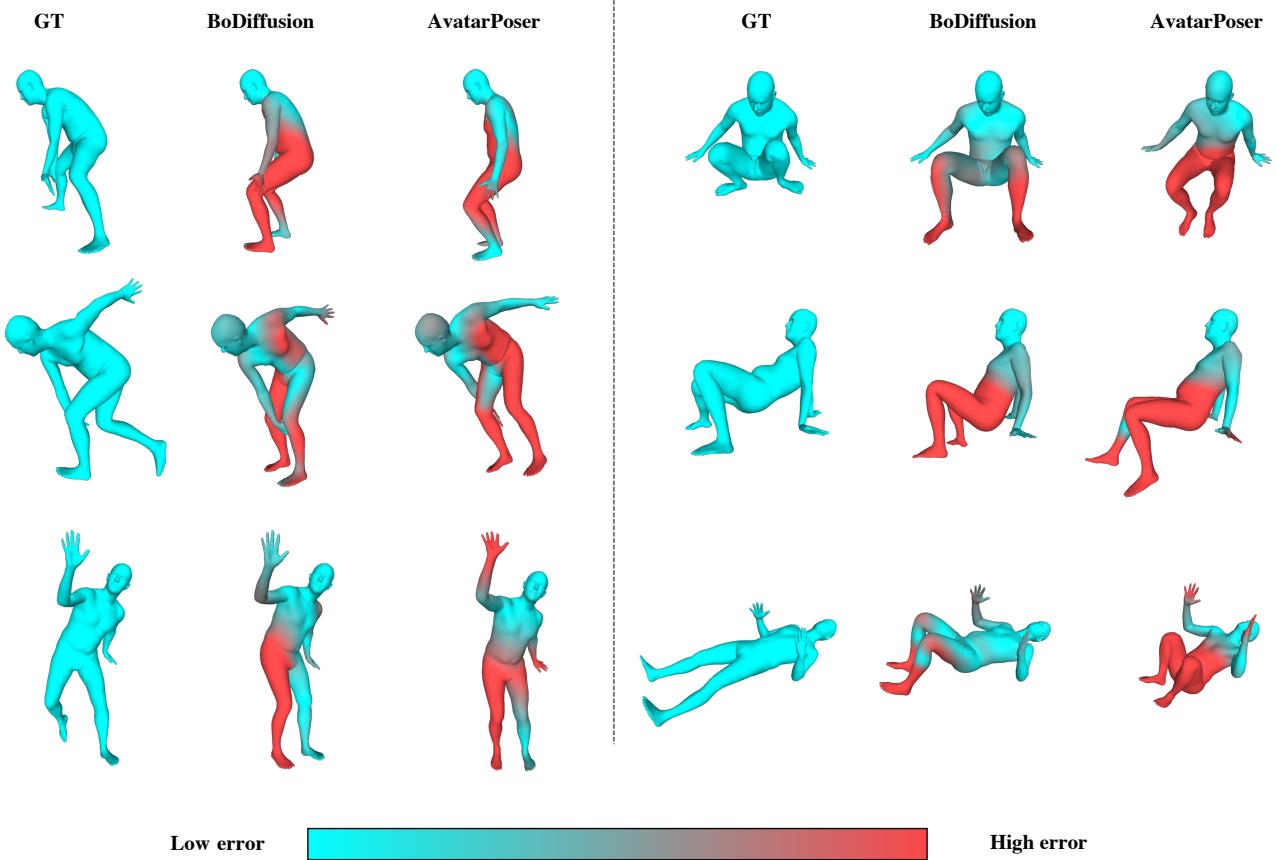
**Lemma 2.** Let  $\mathcal{L}(x, x') = \|x - x'\|^2$  be the local rotation error loss between motion sequences  $x$  and  $x'$ , where  $x'$  is an estimate of  $x$ . Then, optimizing the  $\mathcal{L}_{\text{simple}}$  loss is equivalent to optimizing  $\mathcal{L}$ .

*Proof.* Let the rotation loss be

$$\mathcal{L}(x, x') = \|x - x'\|^2. \quad (14)$$

Considering that  $x_t$  for any single step in the DDPM is generated with Eq. 12, we can solve for  $x$  from this equation. Similarly, since the DDPM model generates an estimate of  $\epsilon$ , we can generate the estimate  $x'$  by replacing  $\epsilon$  with  $\epsilon_\theta$ . Hence,

$$\begin{aligned} x &= \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon), \\ x' &= \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(\mathbf{x}_t)). \end{aligned} \quad (15)$$



**Figure 9. Performance on unconventional poses.** We compare single poses predicted by using BoDiffusion and AvatarPoser [15]. The poses were extracted from sequences of the CMU, BMLRub and HDM5 datasets. Mesh colors denote absolute positional error. Note how our method can predict plausible poses even for uncommon movements like crouching or lying down.

By combining Eq. 15 in Eq. 14, we compute

$$\begin{aligned}
 \mathcal{L}(x, x') &= \|x - x'\|^2 \\
 &= \left\| \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon) \right. \\
 &\quad \left. - \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(\mathbf{x}_t)) \right\|^2 \quad (16) \\
 &= \left\| \sqrt{\frac{1 - \bar{\alpha}_t}{\bar{\alpha}_t}} (\epsilon - \epsilon_\theta(\mathbf{x}_t)) \right\|^2 \\
 &= \frac{1 - \bar{\alpha}_t}{\bar{\alpha}_t} \|\epsilon - \epsilon_\theta(\mathbf{x}_t)\|^2
 \end{aligned}$$

Therefore,

$$\mathcal{L}(x, x') = \frac{1 - \bar{\alpha}_t}{\bar{\alpha}_t} \mathcal{L}_{\text{simple}}(\epsilon, \epsilon_\theta(\mathbf{x}_t)), \quad (17)$$

showing that minimizing the local rotation and the simple loss is equivalent to a scaling factor.  $\square$



Figure 10. **Error visualization on individual poses.** We compare BoDiffusion and AvatarPoser [15] on sequences from the Transitions [27] and HumanEVA [45] datasets. Note how our method can predict poses with higher fidelity to the ground truth. In contrast, AvatarPoser struggles to predict accurate lower-body configurations.

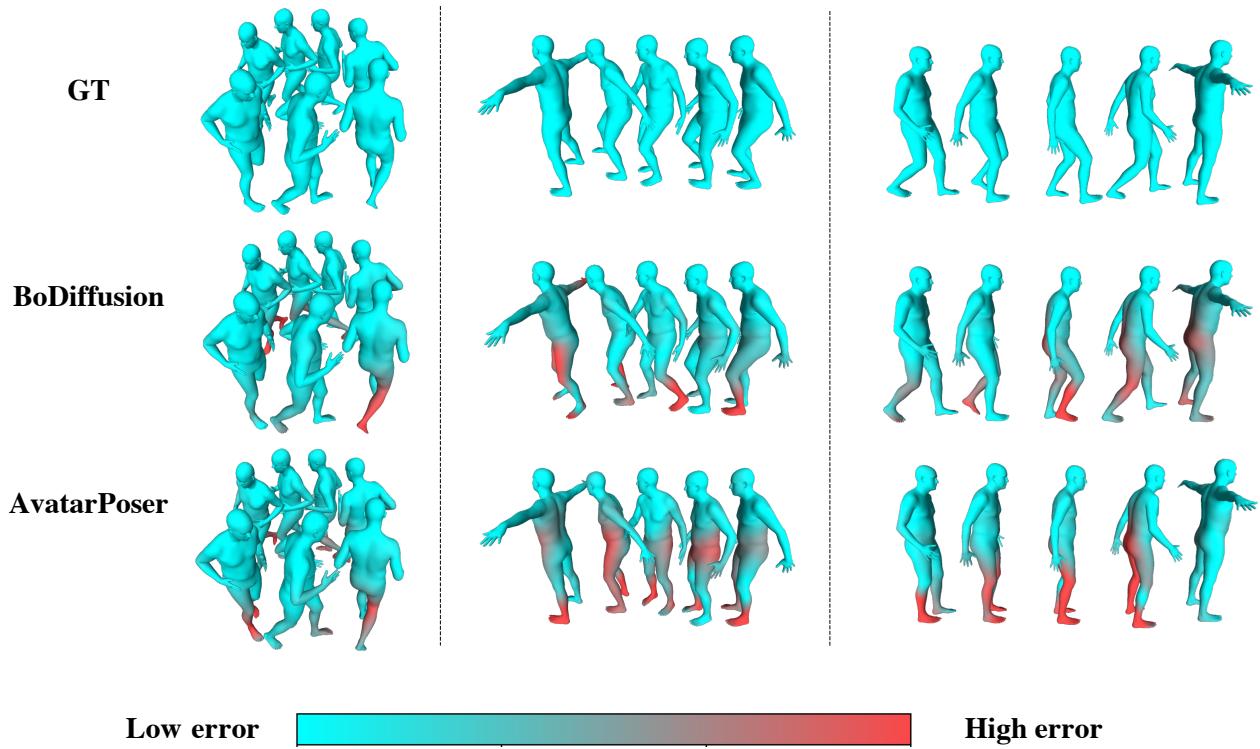


Figure 11. **Error visualization on sequences.** We compare predicted motions of BoDiffusion and AvatarPoser [15] on the test sequences from the Transitions [27] and HumanEVA [45] datasets. Notice that the motions generated by BoDiffusion look more natural and demonstrate better temporal consistency. On the contrary, methods like AvatarPoser struggle to maintain coherence throughout the frames regarding aspects like foot sliding (third sequence).

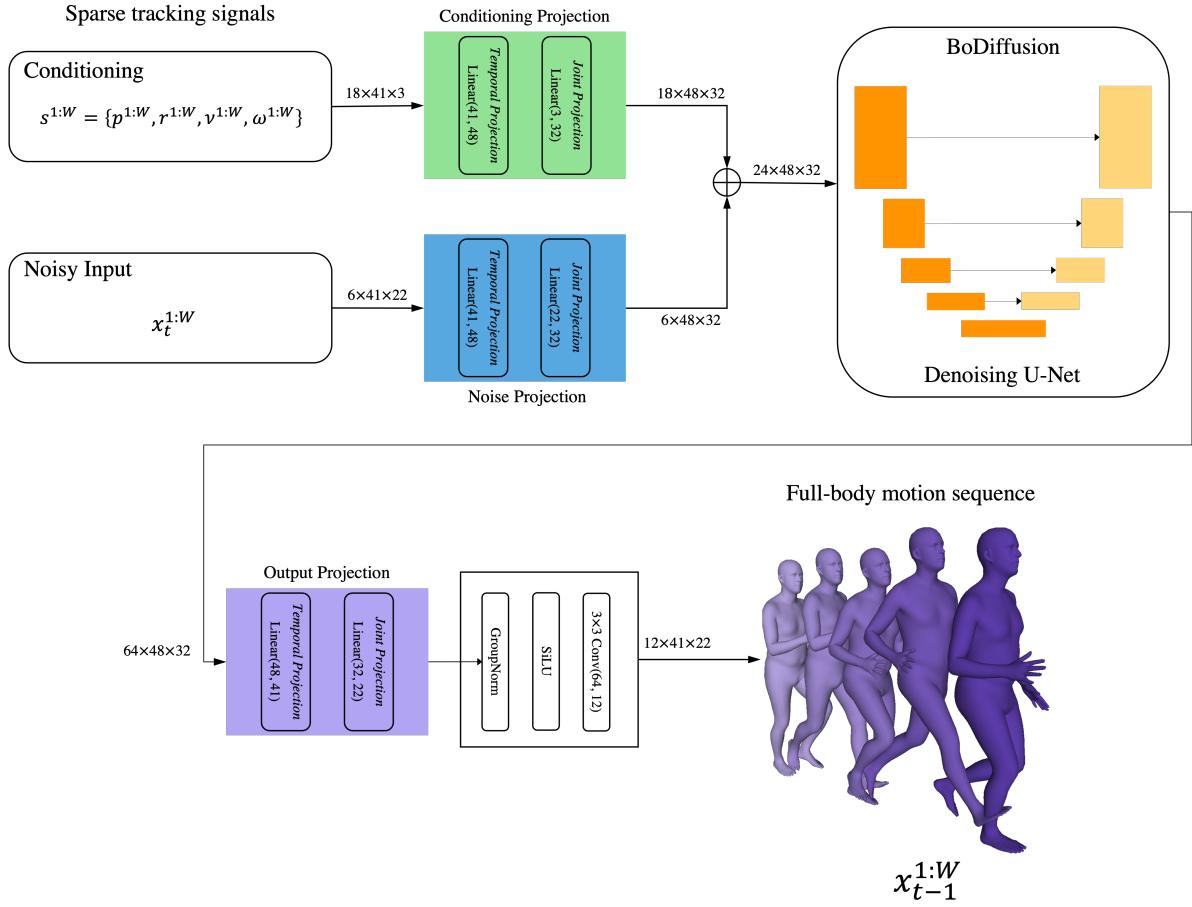


Figure 12. **Overview of BoDiffusion-UNet.** Here we show a version of BoDiffusion based on the U-Net architecture. We condition the input signals after a conditioning projection at the top (green block). Similarly, we project the noisy input local rotations (blue block) to match the sizes from the conditioning pathway. After denoising by the U-Net, we perform a final projection to the original space of full body motions (purple block). The output estimates  $\epsilon_\theta$  and  $\Sigma_\theta$  that are used to compute the local rotations  $x_{t-1}^{1:W}$  by sampling from  $\mathcal{N}(x_{t-1}; \mu_\theta, \Sigma_\theta)$ .  $\oplus$  is the operation of concatenation along the channels' dimension. The numbers next to the arrows denote the input and output dimensions for the corresponding blocks.