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

Angela Castillo*[1] Maria Escobar*[1] Guillaume Jeanneret[2] Albert Pumarola[3] Pablo Arbeláez[1]
Ali Thabet[3] Artsiom Sanakoyeu[3]
[1]Center for Research and Formation in Artificial Intelligence, Universidad de los Andes
[2]University of Caen Normandie, ENSICAEN, CNRS, France
[3]Meta AI

A. Implementation Details

We build upon the SMPL [6, 11, 9] 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 [5], 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 1 and 2, we provide further information on the architecture of our model and technical details. In Figure 1, 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 1, in purple) to map back into the space of motions represented by 6D local rotations of joints. We return a 12-channel tensor which contains predictions of ϵ_θ and Σ_θ (6 channels each) that are used to compute losses $\mathcal{L}_{\text{simple}}$ and $\mathcal{L}_{\text{vlb}}$.

Figure 2 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 γ and β 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 [10, 14] to compute the timestep embedding.

* Equal contributions.

Method	Jitter	MPJVE	MPJPE	MPJRE	FID
BoDiffusion	**0.49**	**14.39**	**3.63**	**2.70**	**0.056**
AvatarPoser– single frames	1.53	28.23	4.20	3.08	0.075
AvatarPoser– predict sequence	2.02	65.22	12.07	4.37	0.107

Table 1. **Smoothness Evaluation.** We retrain AvatarPoser to generate sequences instead of single frames. We report Jitter [km/s^3], MPJVE [cm/s], MPJPE [cm], MPJRE [deg], and FID.

A.2. Inference

At inference time, we use DDIM [13] with 50 iterations and remove the stochasticity during sampling from the distribution by setting the variance Σ_θ to zero. To process the input tracking signal from HMD and hand controllers, we use a sliding window with a temporal window size of $W = 41$. While during online inference, one would typically use a sliding window with a stride of 1, for the sake of faster inference on AMASS dataset, we apply our model using a stride of 20 frames, without observing any degradation in the quality of generated sequences.

Once we have produced all output values for the sequence at inference, we average the overlapping regions to produce a final pose estimate. Notably, although we employ averaging for faster inference, we found that the model's smoothing power does not solely rely on this process. Thus, to ensure a fair comparison, we retrained AvatarPoser to

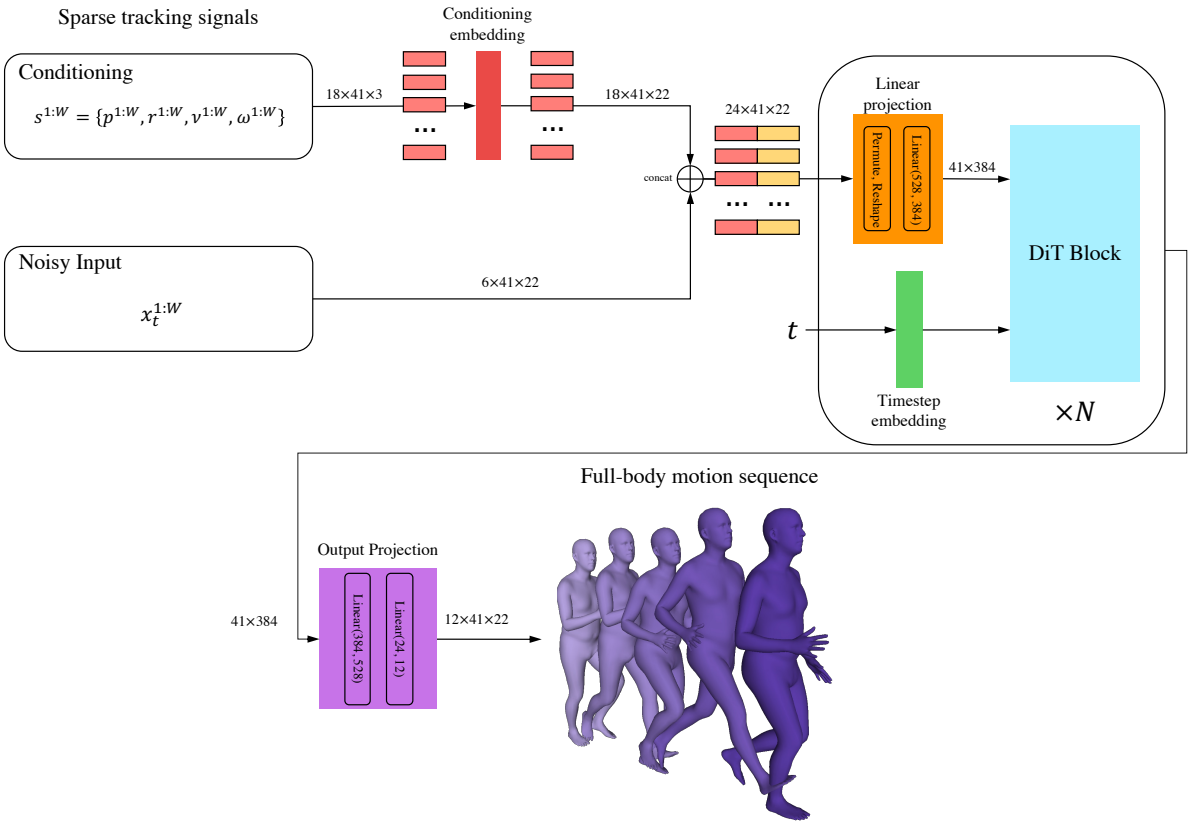

Figure 1. **Complete Overview of BoDiffusion.** Our conditional model takes full advantage of sparse information since we calculate relevant features in the conditioning pathway at the top (red block). In the case of the noisy input, we do not need any projection to match the sizes from the conditioning 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 ϵ_θ and Σ_θ 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.

generate sequences instead of single frames. The results in Table 1 demonstrate that this straightforward extension of AvatarPoser does not provide adequate temporal context to ensure smooth motion estimation. Specifically, the Jitter increases from 1.53 to 2.02, the MPJVE from 28.23 to 65.23, and the MPJPE from 4.10 to 12.07. In contrast, BoDiffusion effectively leverages temporal information to generate smooth sequences, as explained in Sect. 4.1.

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. Model size and computational cost

While the regular AvatarPoser model has 4M parameters, our approach has 22M parameters. Thus, to ensure a fair comparison with AvatarPoser, we rescaled it to 22M parameters (AvatarPoser-Large), as shown in Table 1. Note that BoDiffusion outperforms AvatarPoser-Large in all metrics, creating smoother and more accurate sequences. For one forward pass, AvatarPoser takes 0.16 GFLOPs, while AvatarPoser-Large takes 0.91 GFLOPs. In contrast, BoDiffusion takes 0.48 GFLOPs per forward pass, which amounts to 24 GFLOPs for 50 iterations of DDIM. Nonetheless, our method can be employed with fewer sampling DDIM steps to improve its efficiency, as shown in Table 5 of the main paper.

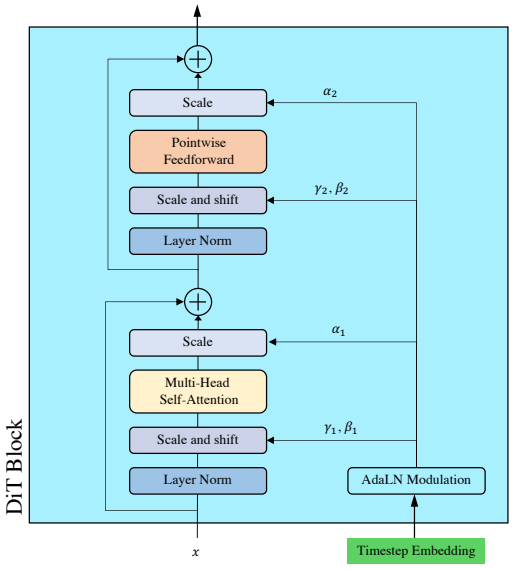

Figure 2. **BoDiffusion Architecture.** Our denoising model is built using multiple DiT blocks. Here we show the details of a single DiT block.

A.5. UNet Architecture for Ablation

To ablate the architecture, we also implemented a version of BoDiffusion using the popular DDPM UNet backbone [2] designed for image data and not for motions, the overview of this architecture is shown in Fig. 7. We followed [2] 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. 3), 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 2 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

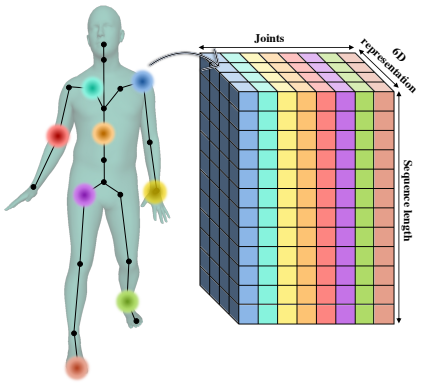

Figure 3. **Input tensor representation for U-Net network.** We represent the motion sequence as a 3D tensor in which the channels correspond to the 6D rotation, the height to the time sequence, and the width to the joints. This representation is analogous to the image data input tensors. In this way, we can reuse the convolutional architectures of the denoising U-Net 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	**3.63**	**2.70**
Window size $W = 81$	**0.46**	**13.69**	3.77	2.86

Table 2. **Window Size Ablation.** We evaluate the importance of including more or less temporal context. We report Jitter [km/s^3], MPJVE [cm/s], MPJPE [cm], and MPJRE [deg].

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 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 4 shows additional qualitative results for BoDiffusion and AvatarPoser [5] on the CMU [1], BMLrub [15], and HDM05 [8] test sets. Notice that our method can generate poses close to the ground truth even when the actions

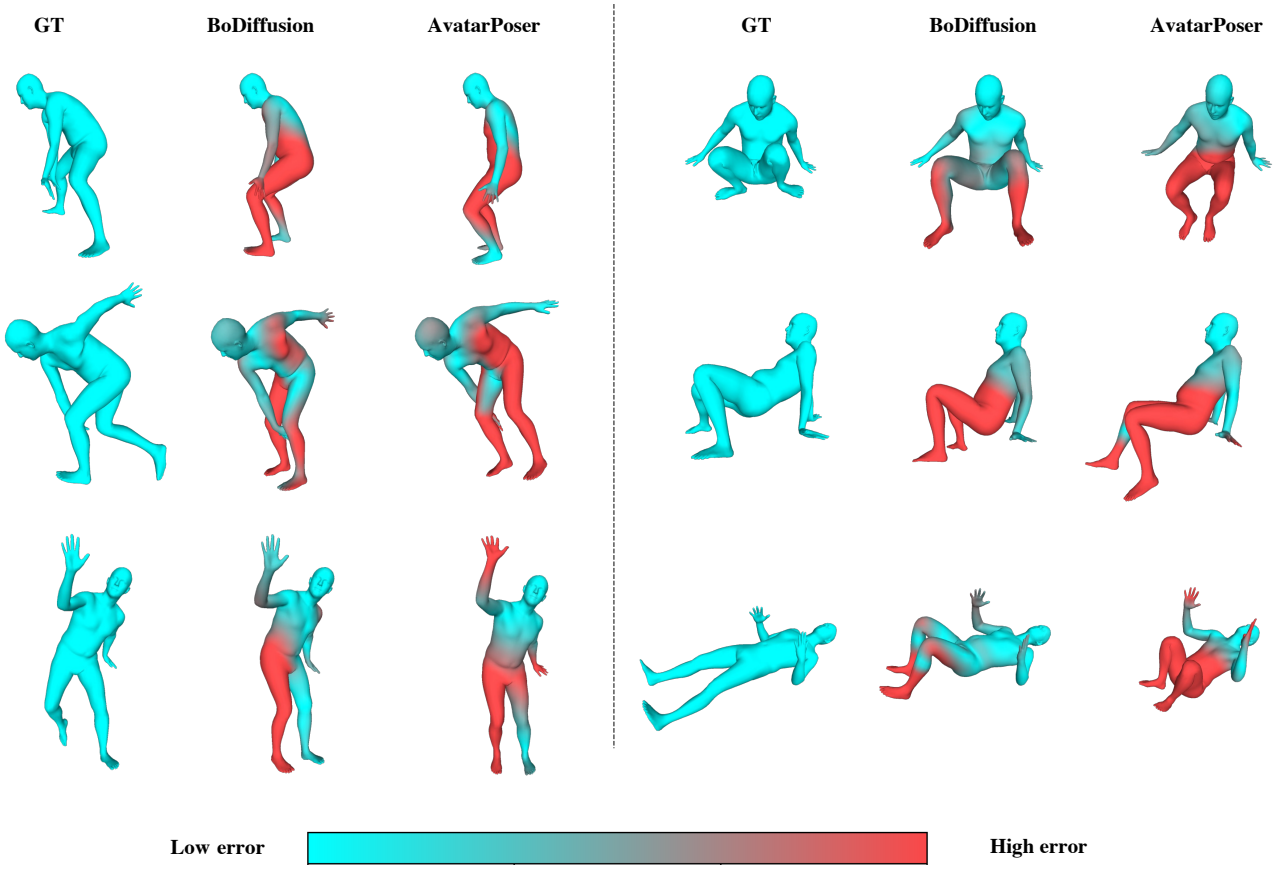

Figure 4. **Performance on unconventional poses.** We compare single poses predicted by using BoDiffusion and AvatarPoser [5]. The poses were extracted from sequences of the CMU, BMLRub and HDM5 datasets. Mesh colors denote absolute positional error. Note how our method can predict plausible poses even for uncommon movements like crouching or lying down.

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 5 and 6 show qualitative results on the Transitions [7] and HumanEVA [12] test sets predicted with BoDiffusion and AvatarPoser [5]. First, note that BoDiffusion generates individual poses with a high fidelity in the upper-body configuration and a plausible lower-body configuration. Second, 6 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 [5] on sequences from the BMLrub [15], Transitions [7] and HumanEVA [12] test sets.

D. Perceptual Quality Evaluation

In Table 1, we calculate the Frechet Inception Distance (FID) between the feature distributions of our generated motions and the real motions. FID is a widely used metric to evaluate perceptual quality across different generation domains. We adapted the implementation designed by [3] specifically for human motion generation. The FID for the standard version of AvatarPoser is 0.075, while for BoDiffusion is 0.056. Thus, this metric further confirms the improvement of our method in generating high-quality motion sequences.

E. Local Rotation Loss

Due to the properties of Gaussian distributions, Ho *et al*. [4] showed that we can directly calculate $\mathbf{x}_t$ from $\mathbf{x}_0$ by sampling:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{1}$$

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

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

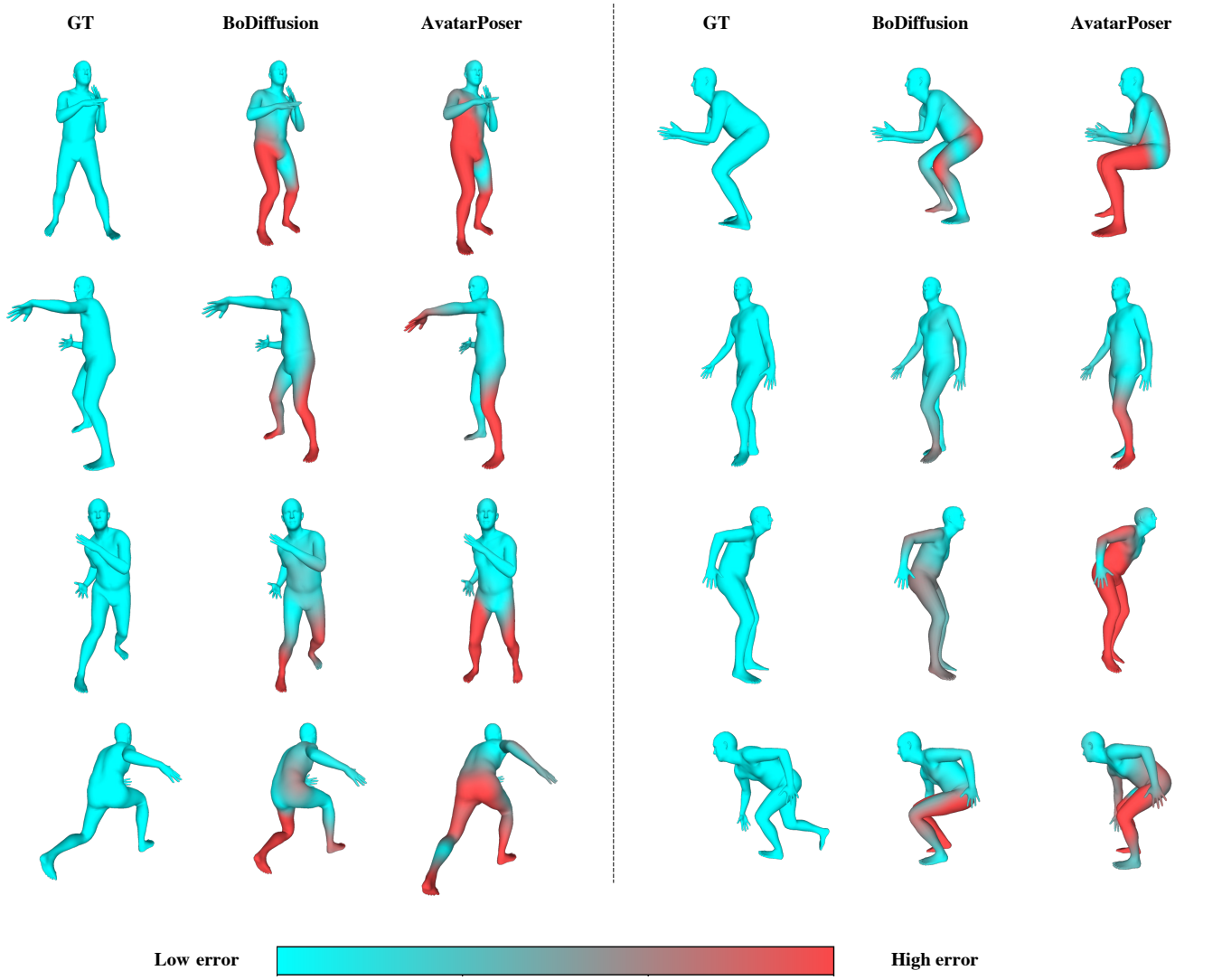

Figure 5. **Error visualization on individual poses.** We compare BoDiffusion and AvatarPoser [5] on sequences from the Transitions [7] and HumanEVA [12] datasets. Note how our method can predict poses with higher fidelity to the ground truth. In contrast, AvatarPoser struggles to predict accurate lower-body configurations.

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

We found that optimizing ϵ_θ to approximate the noise ϵ (Eq. 2) 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 motion sequences x and x', where x' is an estimate of x. Then, optimizing the $\mathcal{L}_{simple}$ loss is equivalent to optimizing $\mathcal{L}$.*

Proof. Let the rotation loss be

$$\mathcal{L}(x, x') = ||x - x'||^2. \tag{3}$$

Considering that x_t for any single step in the DDPM is generated with Eq. 1, we can solve for x from this equation. Similarly, since the DDPM model generates an estimate of ϵ, we can generate the estimate x' by replacing ϵ with ϵ_θ. Hence,

$$\begin{aligned} x &= \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\epsilon), \\ x' &= \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t)). \end{aligned} \tag{4}$$

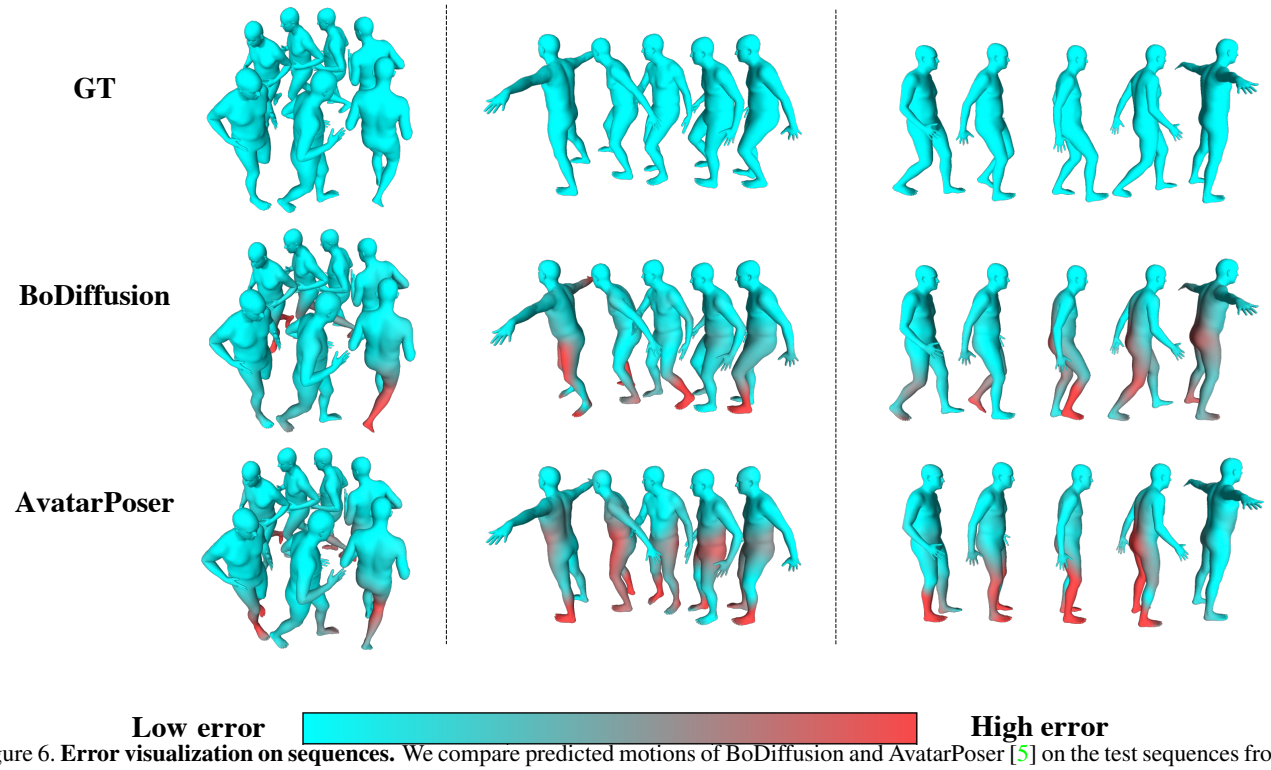

Figure 6. **Error visualization on sequences.** We compare predicted motions of BoDiffusion and AvatarPoser [5] on the test sequences from the Transitions [7] and HumanEVA [12] datasets. Notice that the motions generated by BoDiffusion look more natural and demonstrate better temporal consistency. On the contrary, methods like AvatarPoser struggle to maintain coherence throughout the frames regarding aspects like foot sliding (third sequence).

By combining Eq. 4 in Eq. 3, we compute

$$\begin{aligned}\mathcal{L}(x, x') &= ||x - x'||^2 \\ &= \left|\left| \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon) \right.\right. \\ &\left.\left. - \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t)) \right|\right|^2 \\ &= \left|\left| \sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}}(\epsilon - \epsilon_\theta(\mathbf{x}_t)) \right|\right|^2 \\ &= \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t} ||\epsilon - \epsilon_\theta(\mathbf{x}_t)||^2\end{aligned} \tag{5}$$

Therefore,

$$\mathcal{L}(x, x') = \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}\mathcal{L}_{\text{simple}}(\epsilon, \epsilon_\theta(\mathbf{x}_t)), \tag{6}$$

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

References

[1] Carnegie Mellon University. CMU MoCap Dataset. 3

[2] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 3

[3] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *CVPR*, 2022. 4

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 4

[5] Jiaxi Jiang, Paul Streli, Huajian Qiu, Andreas Fender, Larissa Laich, Patrick Snape, and Christian Holz. Avatarposer: Articulated full-body pose tracking from sparse motion sensing. In *Proceedings of European Conference on Computer Vision*. Springer, 2022. 1, 3, 4, 5, 6

[6] 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. 1

[7] 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. 4, 5, 6

[8] 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. 3

[9] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face,

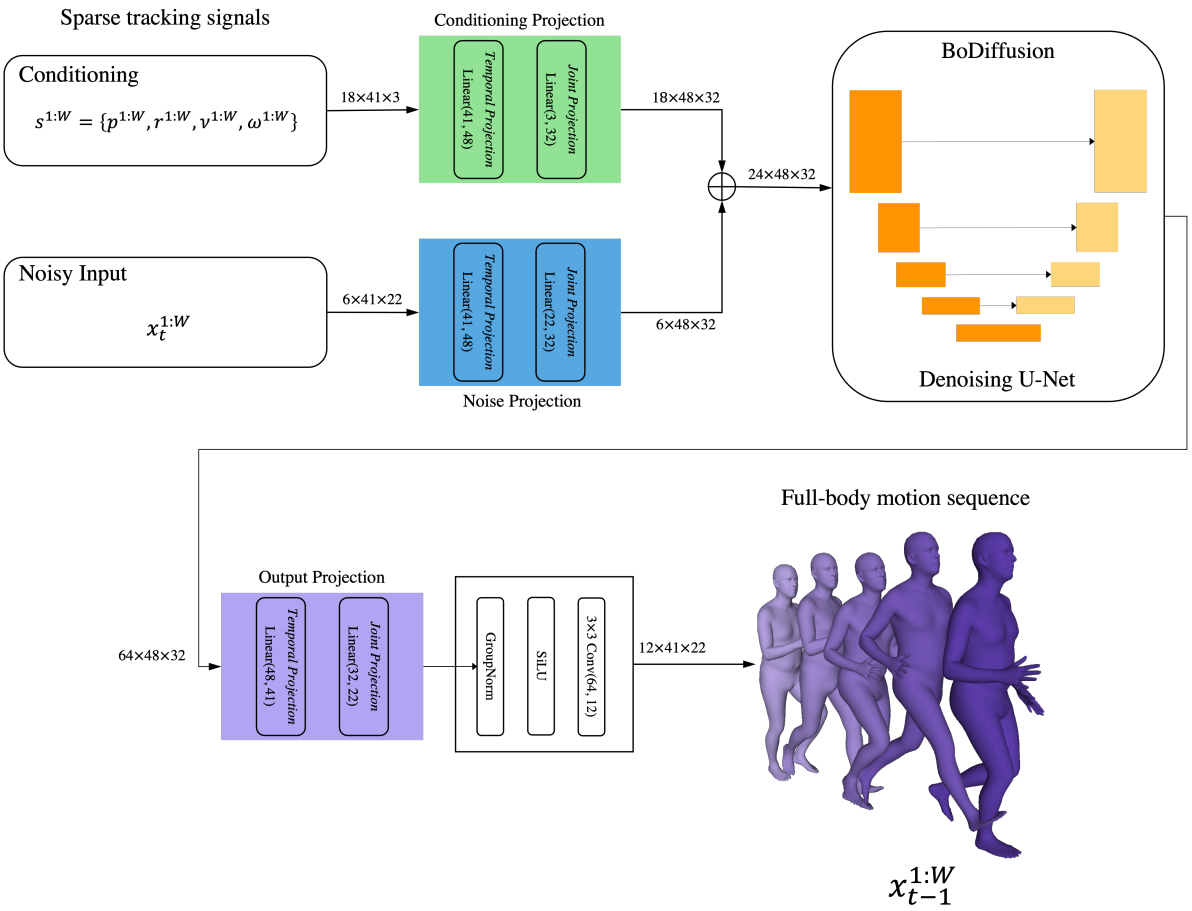

Figure 7. **Overview of BoDiffusion-UNet.** Here we show a version of BoDiffusion based on the U-Net architecture. We condition the input signals after a conditioning projection at the top (green block). Similarly, we project the noisy input local rotations (blue block) to match the sizes from the conditioning 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 ϵ_θ and Σ_θ 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.

and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 1

[10] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 1

[11] 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. 1

[12] 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. 4, 5, 6

[13] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1

[14] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020. 1

[15] 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. 3, 4