Estimating Ego-Body Pose from Doubly Sparse Egocentric Video Data

Seunggeun Chi[†]
Purdue University
chi65@purdue.edu

Pin-Hao Huang[∗], **Enna Sachdeva**[∗]
Honda Research Institute USA
{pin-hao_huang, enna_sachdeva}@honda-ri.com

Hengbo Ma[†]
Honda Research Institute USA
hengbo.academia@gmail.com

Karthik Ramani
Purdue University
ramani@purdue.edu

Kwonjoon Lee
Honda Research Institute USA
kwonjoon_lee@honda-ri.com

Abstract

We study the problem of estimating the body movements of a camera wearer from egocentric videos. Current methods for ego-body pose estimation rely on *temporally dense* sensor data, such as IMU measurements from *spatially sparse* body parts like the head and hands. However, we propose that even *temporally sparse* observations, such as hand poses captured intermittently from egocentric videos during natural or periodic hand movements, can effectively constrain overall body motion. Naively applying diffusion models to generate full-body pose from head pose and sparse hand pose leads to suboptimal results. To overcome this, we develop a two-stage approach that decomposes the problem into temporal completion and spatial completion. First, our method employs masked autoencoders to impute hand trajectories by leveraging the spatiotemporal correlations between the head pose sequence and intermittent hand poses, providing uncertainty estimates. Subsequently, we employ conditional diffusion models to generate plausible full-body motions based on these temporally dense trajectories of the head and hands, guided by the uncertainty estimates from the imputation. The effectiveness of our method was rigorously tested and validated through comprehensive experiments conducted on various HMD setup with AMASS and Ego-Exo4D datasets. Project page: https://sgchi.github.io/dsposer/

1 Introduction

The evolution of augmented reality (AR) devices such as the Apple Vision Pro, Meta Quest 3, Microsoft HoloLens 2, and etc. has dramatically reshaped interactive technologies. These head-mounted displays (HMDs) feature inertial measurement units (IMUs) and video capture capabilities, offering a unique egocentric perspective. However, their limited visibility of the user's body parts poses a significant challenge for accurate egocentric body pose estimation—a key element for immersive AR experiences.

Previous approaches have tackled this problem by spatially reconstructing the entire body from spatially sparse data. For instance, EgoEgo [16] first estimates head poses using SLAM on the egocentric video, then generates body poses from these estimated head positions. Other methods, such as AvatarPoser [13] and BoDiffusion [3], primarily depend on temporally dense tracking signal from spatially sparse body parts, notably the head and hands. This dependency on specific hardware

---

[∗]co-second authors

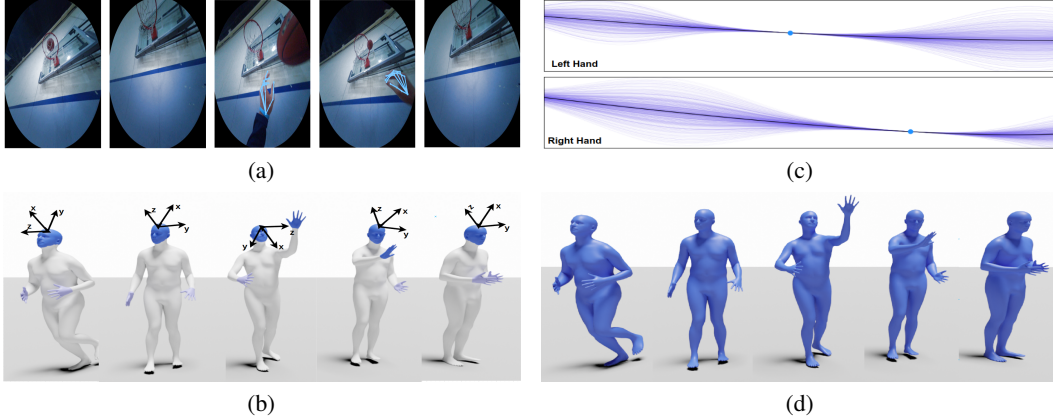[†]work done at Honda Research Institute USA.

Figure 1: Overview of DSPoser. Our goal is to estimate ego-body pose without dependency on hand controllers in an HMD environment. (a) Given the egocentric video and head tracking signals as input, (b) our approach first predicts the hand pose in the frames where hands are visible (dark blue). It then estimates the hand poses in frames with invisible hands (light blue) using imputation, and (c) estimates uncertainty associated with the hand poses where the hands are invisible, (d) The predicted and imputed hand pose is then used with head pose to predict the 3D full body pose.

such as head-mounted displays and hand controllers constrains their versatility and diminishes their applicability in broader AR/VR scenarios where hand controllers might not be used, like sports training or analysis applications where the user needs to move freely without holding any devices, or augmented reality experiences in outdoor environments where carrying controllers is impractical.

We observe that even *temporally sparse* observations, such as hand poses captured intermittently from egocentric videos during natural or periodic hand movements, can effectively constrain overall body motion. While it is possible to utilize other visible body parts such as feet or elbows, we opted to rely on hand poses. This decision is based on the availability of hand pose detectors [24, 14] and the fact that hands are visible in approximately 20% of video frames, as demonstrated in Table 8. Unlike previous work that concentrated only on spatial completion, our method incorporates temporal completion by leveraging the intermittent appearance of hands in egocentric videos. This dual completion approach not only enhances the robustness of body pose estimation under varying conditions but also reduces reliance on specific sensor hardware, making it more adaptable to various AR environments. In our setup, we use temporally sparse 3D hand poses from detections in egocentric videos combined with dense head tracking signals to reconstruct the full body. Initially, we temporally complete sparse hand information using a Masked Autoencoder (MAE) [11], which estimates hand pose trajectories by capturing the spatiotemporal correlations between intermittent hand poses and head tracking signals. We develop a probabilistic extension of the MAE to provide uncertainty estimates of the predicted hand pose sequence. Subsequently, using a conditional diffusion model, we spatially reconstruct the full body based on the head tracking signal data and imputed hand trajectories along with their predictive uncertainties. We call our approach **DSPoser** (Doubly Sparse Poser) because it can effectively utilize data that is *doubly sparse* (sparse both temporally and spatially), as shown in Figure 1.

This flexible framework is designed to seamlessly adapt to diverse AR/VR setups and devices, ranging from spatially sparse scenarios (e.g., using only head tracking signal or combining it with hand controllers) to doubly sparse scenarios (utilizing head signal data alongside hand detection from egocentric video). The key advantage lies in the assumption that the HMD's tracking signal is consistently available, enabling our approach to function across a wide range of environments and hardware configurations. Extensive experiments have proved our model's versatility and accurate pose estimation capabilities in various settings. Furthermore, our ablation studies highlight the significance of incorporating uncertainty estimates, as this crucial information enhances the overall quality of pose estimation, resulting in more reliable outputs. By addressing both temporal and spatial completion through our double completion approach, we have developed a robust and adaptable solution that reduces dependency on specific sensor hardware, making it well-suited for immersive AR experiences in diverse scenarios, such as sports training, outdoor environments, and beyond.

In summary, our research presents three key contributions:

- A robust and versatile framework for egocentric body pose estimation tailored for HMDs. The framework adapts to various AR/VR settings and can leverage tracking signals available in most modern HMD devices without controllers.

- We decomposed the problem into temporal completion and spatial completion. Our approach captures the uncertainty from hand trajectory imputation to guide the diffusion model for accurate full-body motion generation.

- Extensive evaluations demonstrating the effectiveness of our framework on diverse datasets, outperforming existing methods and underscoring its potential for enhancing user interaction and immersion in AR experiences.

2 Problem Formulation

In our work, we aim to estimate the 3D human pose of a HMD user from sequences of RGB video and head tracking signal. We note that head tracking signal data is commonly accessible from IMU in most HMDs, such as Meta Quest and Apple Vision Pro. Suppose we are given an egocentric video $\mathcal{V}_{\text{ego}} = \{\mathcal{V}_1, \ldots, \mathcal{V}_{T_w}\}$ where $\mathcal{V}_\tau$ is an RGB image and T_w denotes the sequence length, and a corresponding head tracking signal sequence $\mathcal{T}_{\text{head}} = \{\mathcal{T}_1, \ldots, \mathcal{T}_{T_w}\}$ where $\mathcal{T}_\tau \in \mathbb{R}^{D_{head}}$ and D_{head} is the dimension of head tracking signal including 3D pose. Our goal is to estimate the full body pose $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_{T_w}\}$, where pose state $\mathcal{P}_\tau \in \mathbb{R}^{J \times D}$ at time τ, J is the number of body joints and D is the dimensionality of pose state. We solve the problem of estimating $p(\mathcal{P}|\mathcal{V}_{ego}, \mathcal{T}_{\text{head}})$ by decomposing it into 2 the stages of imputation and generation, assuming that we have *temporally sparse* hand data $\dddot{\mathcal{H}}$ from hand detection module $f(\cdot)$: $\dddot{\mathcal{H}} = f(\mathcal{V}_{ego})$. We first *temporally complete* hand trajectory $\widetilde{\mathcal{H}}$ based on $\dddot{\mathcal{H}}$ and $\mathcal{T}_{\text{head}}$, which can be written as $p(\widetilde{\mathcal{H}}|\dddot{\mathcal{H}}, \mathcal{T}_{\text{head}})$. Then, we *spatially complete* full body pose $\mathcal{P}$ from the imputed hands $\widetilde{\mathcal{H}}$ and $\mathcal{T}_{\text{head}}$, which can be written as $p(\mathcal{P}|\widetilde{\mathcal{H}}, \mathcal{T}_{\text{head}})$. Since $\widetilde{\mathcal{H}}$ is a probabilistic variable, we need to marginalize over $\widetilde{\mathcal{H}}$ as follows:

$$p(\mathcal{P}|\mathcal{V}_{ego}, \mathcal{T}_{\text{head}}) = \int_{\widetilde{\mathcal{H}}} p(\mathcal{P}|\widetilde{\mathcal{H}}, \mathcal{T}_{\text{head}}) p(\widetilde{\mathcal{H}}|f(\mathcal{V}_{ego}), \mathcal{T}_{\text{head}}). \tag{1}$$
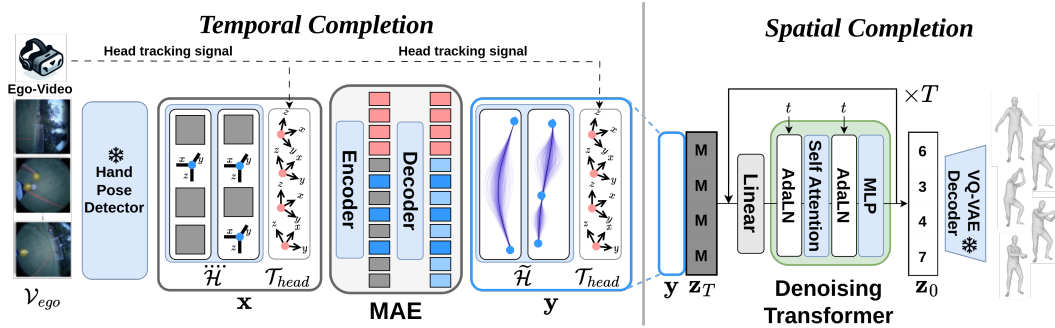
3 Methods

Figure 2: Overall pipeline of our proposed work DSPoser, composed of *Temporal Completion* stage and *Spatial Completion* stage to tackle pose estimation problem from *doubly sparse* data.

3.1 Detection: Hand Pose Estimation from Egocentric Video

In this work, we estimate the 3D position of the hand from an egocentric camera using a two-step process. First, we use FrankMocap [24] to predict hand poses as SMPL-X parameters [22], from which we extract local 3D hand joint positions relative to the root of the hand model's kinematic tree, denoted as $\mathcal{H}_h^{3D} \in \mathbb{R}^{21 \times 3}$. Simultaneously, we use RTM-Pose [14] to estimate 2D hand joint positions in the image, $\mathcal{H}_I^{2D} \in \mathbb{R}^{21 \times 2}$. Finally, we determine the 3D hand joint positions in the camera coordinate system, $\mathcal{H}_I^{3D} = \mathcal{H}_h^{3D} + \mathbf{d}$ by solving for $\mathbf{d} \in \mathbb{R}^3$ that minimizes the reprojection

error $\|\mathcal{H}_I^{2D} - \mathbf{K}(\mathcal{H}_h^{3D} + \mathbf{d})\|_2$. Here, $\mathbf{K}$ is the intrinsic matrix, obtained by transforming the original camera parameters into a pinhole model through undistortion.

To better constrain the hand trajectories, we attempted to obtain rotation information from the 3D hand detection. However, due to the inconsistent quality of hand detection, the rotational information derived from the hand pose was noisy. Therefore, we decided not to incorporate this rotational information into our hand tracking approach on the Ego-Exo4D dataset. We utilized only the 3D wrist location from the Ego-Exo4D dataset, represented by $D_{hand} = 3$. In contrast, for the AMASS dataset, we leveraged both rotational information and 3D location, as this data is readily available, resulting in $D_{hand} = 9$.

3.2 Temporal Completion: Hand Trajectory Imputation from Sparse Hand Pose

Masked Auto-Encoder (MAE)   In our work, we employed a Masked Autoencoder (MAE) [11] to impute missing hand trajectories using head tracking signal $\mathcal{T}_{head}$ and detected hand pose $\overset{\cdots}{\mathcal{H}}$. Inspired by Vision Transformer (ViT), we treated each $\mathcal{T}_\tau$ and $\overset{\cdots}{\mathcal{H}}_\tau$ at time τ as a token similar to an image patch in ViT. To accommodate this, we implemented two embedding layers, one for head tracking signal $\mathcal{T}_\tau \in \mathbb{R}^{D_{head}}$ and the other for hand $\overset{\cdots}{\mathcal{H}}_\tau \in \mathbb{R}^{D_{hand}}$, both projecting into the common token dimension D_M. For the AMASS dataset, we follow the head tracking signal representation $D_{head} = 18$ as in [13]. For the Ego-Exo4D dataset, $D_{head} = 15$, which includes head position and left/right IMU signals. Consequently, the total number of token amounts to $3 \times T_w$, where 3 accounts for the head and both hands, and T_w is the sequence length. Sinusoidal positional encoding (PE) is used for both the encoder and decoder patches after tests showed it suffices for learning different modalities, compared to learnable PE. In an HMD environment, we assume that the head tracking signal $\mathcal{T}_{head}$ is always available, but hand visibility depends on the egocentric video. Thus, masking is applied only to the hand tokens based on their visibilities within egocentric view.

In contrast to the MAE [11] training approach, which maintains a consistent number of masked patches due to a fixed masking ratio, the count of frames with invisible hand varies across instances in our setup. To address this variability, our encoder selectively applies attention masking to these inputs, ensuring that queries do not attend to tokens where hand is invisible. This attention masking technique adapts dynamically to the fluctuating numbers of missing frames across the instances, enhancing the model's ability to handle data sparsity effectively. For decoder, we adopted MAE decoder design except the last projection layer to guide the uncertainty. To capture the uncertainty, we split the final projection layer into two heads for mean and variance of a Gaussian distribution.

Uncertainty-aware MAE   Following the [26, 30], to make the MAE aware of the predictive uncertainty of imputed hand pose sequence, we employ the β-NLL loss [26] function to manage uncertainty by using a set of mean heads $\mu_i(\mathbf{x})$ and variance heads $\sigma_i^2(\mathbf{x})$, which are derived from M models initialized differently, where $\mathbf{x} = [\overset{\cdots}{\mathcal{H}}; \mathcal{T}]$ is an input to the MAE and $i \in [1, M]$. The mean heads $\mu_i(\mathbf{x})$ and variance heads $\sigma_i^2(\mathbf{x})$ are trained using the Gaussian negative log-likelihood loss, which applies to each sample indexed by n with input $\mathbf{x}_n$ and ground truth hand pose sequence $\mathbf{y}_n$.

$$L_{\beta-\text{NLL}}(\mathbf{y}_n, \mathbf{x}_n) = \text{sg}(\sigma_i^{2\beta}) L_{\text{NLL}}(\mathbf{y}_n, \mathbf{x}_n) \text{ where,} \tag{2}$$

$$L_{\text{NLL}}(\mathbf{y}_n, \mathbf{x}_n) = \frac{\log \sigma_i^2(\mathbf{x}_n)}{2} + \frac{(\mu_i(\mathbf{x}_n) - \mathbf{y}_n)^2}{2\sigma_i^2(\mathbf{x}_n)}. \tag{3}$$

The L_{NLL} loss function causes the predicted variance to act as a weighting factor for each data point, emphasizing those with higher variances. The parameter β adjusts the intensity of this weighting. The $\text{sg}(\cdot)$ function is used to apply the stop-gradient operation, thus preventing gradients from propagating through this part of the computation.

After training, we measure the aleatoric (data) uncertainty $\mathcal{U}_{ale}(\cdot)$ by averaging the variances across models, and epistemic (model) uncertainty $\mathcal{U}_{epi}(\cdot)$ by calculating variance of model means, and total uncertainty by adding both uncertainties:

$$\mathcal{U}_{ale}(\mathbf{x}) = \mathbb{E}_i[\sigma_i^2(\mathbf{x})] \approx M^{-1} \sum_i \sigma_i^2(\mathbf{x}) \tag{4}$$

$$\mathcal{U}_{epi}(\mathbf{x}) = \text{Var}_i[\mu_i(\mathbf{x})] \tag{5}$$

$$\mathcal{U}_{tot}(\mathbf{x}) = \mathcal{U}_{ale}(\mathbf{x}) + \mathcal{U}_{epi}(\mathbf{x}) \tag{6}$$

Note that $\mathcal{U}_{ale}(\cdot)$ and $\mathcal{U}_{epi}(\cdot)$ provide uncertainties for each frame and each pose state dimension. The captured uncertainty is visualized in Figure 3, demonstrating that MAE effectively captures uncertainty.

3.3 Spatial Completion: Uncertainty-guided Body Pose Generation from Imputed Hand Trajectories and Head Tracking Signal

We employed the VQ-Diffusion [23] to generate full body poses from imputed hand trajectories and head tracking signal. The exposition of VQ-Diffusion can be found in Section D of the Appendix. As illustrated in Figure 2, our motion generation module is designed to generate human motion sequences from the temporally dense hand and head trajectories with uncertainty obtained from the MAE model.

VQ-VAE   We first train the VQ-VAE to represent human motion with a discrete codebook representation as described in Appendix D.1. We mostly followed the architectural design and training methods of [35]. After the codebook representation is learned by the VQ-VAE, we utilize this latent codebook representation to train a denoising diffusion model.

Denoising Transformer   Motivated by the work of VQ-Diffusion, we design a denoising transformer that estimates the distribution $p(\mathbf{z}_0|\mathbf{z}_t, \mathbf{y})$. An overview of our proposed model is depicted in Figure 2. We closely follow the implementation of [4]. To incorporate the diffusion step t into the network, we employ the adaptive layer normalization (AdaLN) [2, 15]. We concatenated the estimated hand and head trajectory with codebook after a embedding layer, to match the dimension with codebook representation. Finally, we use the decoder to decode $\mathbf{z}_0$ to obtain a full body pose sequence.

Uncertainty Guidance   We introduce several strategies to guide the denoising process using uncertainty estimates of imputed hand trajectories: sampling, dropout, and distribution embedding.

For *sampling*, we sample a hand sequence from the distribution $\widetilde{\mathcal{H}} \sim \mathcal{N}(\mu^*(\mathbf{x}), \sqrt{\mathcal{U}^*(\mathbf{x})})$ and regard it as the conditioning vector $\mathbf{y}$, where $\mu^*(\mathbf{x}) = \mathbb{E}_i[\mu_i(\mathbf{x})] \approx M^{-1}\sum_i \mu_i(\mathbf{x})$ and $\mathcal{U}^*(\mathbf{x})$ is measured by one of Eq. (9), (10), and (11). While it would be ideal to sample multiple times to better approximate the marginalization in Equation 1, we find just using one sample provides a competitive performance.

For *dropout*, we set each dimension of $\mu(\mathbf{x})$ to zero with a certain probability, which is determined by the corresponding dimension of $\mathcal{U}^*(\mathbf{x})$, and denote the result as y. The probability of the d-th dimension of $\mu(\mathbf{x})$ being zero is $p_d = 1 - (\mathcal{U}_d^*(\mathbf{x}) - \mathcal{U}_{d^{min}}^*(\mathbf{x}))/(\mathcal{U}_{d^{max}}^*(\mathbf{x}) - \mathcal{U}_{d^{min}}^*(\mathbf{x}))$ where $\mathcal{U}_d^*(\mathbf{x})$ is the d-th dimension of $\mathcal{U}^*(\mathbf{x})$, and $\mathcal{U}_{d^{min}}^*(\mathbf{x}), \mathcal{U}_{d^{max}}^*(\mathbf{x})$ are the minimum and maximum values over the sequence length, respectively.

For *distribution embedding* [28], we embed the Gaussian distribution $\mathcal{N}(\mu^*(\mathbf{x}), \sqrt{\mathcal{U}^*(\mathbf{x})})$ to a vector by concatenating the $\mu^*(\mathbf{x})$ and $\mathcal{U}^*(\mathbf{x})$ in the feature dimension. The resulting embedding will be further concatenated with the head pose sequence to form a conditioning vector $\mathbf{y}$.

4   Experiments

4.1   Datasets & Evaluation Metrics

Ego-Exo4D dataset   Ego-Exo4D [9] contains simultaneous captures of egocentric (first-person) and exocentric (third-person) video perspectives of participants performing complex activities like sports, dance, and mechanical tasks. The dataset comprises 1,422 hours of video ranging from 1 to 42 minutes per video. In addition to video, it provides camera poses, IMU data, and human pose annotations. Specifically for the egopose task, it includes separate training and validation video sets containing 334 and 83 videos respectively. Our problem formulation of ego body pose estimation differs from the ego body pose prediction task from [9], which aims to predict a single future frame given a specific time window.

AMASS dataset   The AMASS dataset [19] is a large human motion database that unifies different existing optical marker-based MoCap datasets by converting them into realistic 3D human meshes

Table 1: Performance comparisons across baseline models for doubly sparse video data on the **AMASS** test set. We report MPJRE [°], MPJPE [cm], and MPJVE [cm/s], with the best results highlighted in **boldface**. Models trained by us are marked with *. The notation $\overset{....}{\text{data}}$ denotes temporally sparse data, $\widetilde{\text{data}}$ indicates imputed data, and all other cases involve dense data. T_s indicates the sliding window, **x** indicates the input of our whole pipeline, and **y** indicates the input of denoising Transformer.

Methods	T_s	x	Imputation	y	MPJPE	MPJVE	MPJRE
VQ-VAE (Recons)	20	Full body	-	Full body	1.26	11.37	1.81
EgoEgo* [16]	20	👤	-	👤	19.17	46.17	7.30
Bodifusion* [3]	20	👤	-	👤	19.27	60.29	8.51
DSPoser (Ours)	20	👤	-	👤	$12.08^{\pm0.04}$	$75.07^{\pm0.26}$	$7.04^{\pm0.02}$
DSPoser (Ours)	20	👤	MAE	👤 & $\widetilde{✋}$	$\mathbf{7.06^{\pm0.02}}$	$\mathbf{28.26^{\pm0.05}}$	$\mathbf{5.00^{\pm0.01}}$
Bodiffusion [3]	20	👤 & $\overset{..}{✋}$	Interpolation	👤 & ✋	46.45	75.33	17.99
Bodiffusion [3]	20	👤 & $\overset{..}{✋}$	MAE	👤 & $\widetilde{✋}$	7.35	31.33	5.47
DSPoser (Ours)	20	👤 & $\overset{..}{✋}$	MAE	👤 & $\widetilde{✋}$	$\mathbf{5.51^{\pm0.02}}$	$\mathbf{24.19^{\pm0.10}}$	$\mathbf{4.09^{\pm0.02}}$
AvatarPoser [13]	1	👤 & $\overset{..}{✋}$	Interpolation	👤 & ✋	40.42	64.07	16.37
AvatarJLM [36]	1	👤 & $\overset{..}{✋}$	Interpolation	👤 & ✋	25.02	68.42	14.14
AvatarPoser [13]	1	👤 & $\overset{..}{✋}$	MAE	👤 & $\widetilde{✋}$	9.88	62.31	5.98
AvatarJLM [36]	1	👤 & $\overset{..}{✋}$	MAE	👤 & $\widetilde{✋}$	7.12	**37.60**	5.24
DSPoser (Ours)	1	👤 & $\overset{..}{✋}$	MAE	👤 & $\widetilde{✋}$	$\mathbf{5.87^{\pm0.13}}$	$49.12^{\pm0.24}$	$\mathbf{4.31^{\pm0.10}}$

represented by SMPL [17] model parameters. Following the AvatarPoser [13] evaluation, we used the CMU [5], BMLrub [29], and HDM05 [21] subsets from the AMASS dataset and their preprocessing of tracking signal information. Since AMASS does not include RGB images, we set $D_{hand} = 9$ assuming that 3D hand position and 6D rotation is available when the hand is "visible". To determine visibility, we compute the angle between the z-axis vector of the head rotation and the vector from the head position to the hand. We define the hand as "visible" if this angle is within a 45° range, corresponding to a 90° field of view (FoV) of HMD devices.

Evaluation metric   We evaluate our results using the following metrics: Mean Per Joint Position Error (MPJPE), Mean Per Joint Velocity Error (MPJVE), and Mean Per Joint Rotation Error (MPJRE), following the evaluation of [13, 3]. Since Ego-Exo4D dataset doesn't have the annotations for 6D rotation, MPJRE is reported only for AMASS. We report all values with the confidence interval of 95%. We also provide details on MPJPE across hands, upper body above the pelvis, and lower body below the pelvis, denoted as Hand PE, Upper PE, and Lower PE, respectively.

4.2   Full Body Pose Estimation from Doubly Sparse data

To demonstrate the effectiveness of our framework on doubly sparse egocentric video data, we investigated the results of our framework, DSPoser, on the AMASS dataset and Ego-Exo4D, as shown in Table 1 and Table 2, respectively. Since the task of body pose estimation from doubly sparse data is newly introduced in our paper, we compare our results to other baselines, EgoEgo [16], Bodiffusion [3], AvatarPoser [13], and AvatarJLM [3]. Those baselines are designed to estimate human body poses from spatially sparse data. EgoEgo estimates body poses from head poses, and the others estimate body poses from head and hand tracking signals. We report the experimental results using the sampling strategy with aleatoric uncertainty unless otherwise stated. To train the baslines on temporally sparse data, we extend the algorithm as follows: (1) *Interpolation*: we imputed hand poses with linear interpolation; (2) *MAE*: we use our trained MAE to impute the hand trajectory. In $T_s = 1$ setup, we report our result after averaging 16 samples while the result in $T_s = 20$ setup is from a single sample.

As shown in Table 1, DSPoser consistently outperforms baseline methods on AMASS across all metrics, underscoring the effectiveness of our two-stage approach for ego-body pose estimation. DSPoser achieves notable improvements in MPJPE for both sliding window sizes, $T_s = 20$ and $T_s = 1$. For $T_s = 20$, DSPoser reduces MPJPE from 7.35 cm to 5.51 cm, significantly outperforming the Bodiffusion extension, which uses MAE to impute invisible hands. For $T_s = 1$, DSPoser

6

Table 2: Performance comparisons across baseline models for doubly sparse video data on the **Ego-Exo4D** validation set. We report MPJPE [cm] and MPJVE [cm/s], with the best results highlighted in **boldface**. Models trained by us are marked with *. The notation $\ddot{\text{Data}}$ denotes temporally sparse data, $\tilde{\text{data}}$ indicates imputed data, and all other cases involve dense data.

Methods	T_s	**x**	Imputation	**y**	MPJPE	MPJVE
VQ-VAE (Recons)	20	Full body	-	Full body	6.77	33.29
EgoEgo* [16]	20	◯	-	◯	29.49	47.50
Bodiffusion* [3]	20	◯	-	◯	28.56	109.71
DSPoser (Ours)	20	◯	-	◯	$19.12^{\pm0.06}$	$48.54^{\pm0.11}$
DSPoser (Ours)	20	◯	MAE	◯ & $\tilde{✋}$	$\mathbf{18.46^{\pm0.06}}$	$\mathbf{40.67^{\pm0.11}}$
Bodiffusion* [3]	20	◯ & $\ddot{✋}$	Interpolation	◯ & $\tilde{✋}$	59.81	120.12
Bodiffusion* [36]	20	◯ & $\ddot{✋}$	MAE	◯ & $\tilde{✋}$	22.12	53.30
DSPoser (Ours)	20	◯ & $\ddot{✋}$	MAE	◯ & $\tilde{✋}$	$\mathbf{16.84^{\pm0.04}}$	$\mathbf{39.86^{\pm0.05}}$
AvatarPoser* [13]	1	◯ & $\ddot{✋}$	Interpolation	◯ & $\tilde{✋}$	47.28	89.34
AvatarJLM* [36]	1	◯ & $\ddot{✋}$	Interpolation	◯ & $\tilde{✋}$	43.01	61.98
AvatarPoser* [13]	1	◯ & $\ddot{✋}$	MAE	◯ & $\tilde{✋}$	24.54	62.34
AvatarJLM* [3]	1	◯ & $\ddot{✋}$	MAE	◯ & $\tilde{✋}$	21.08	**45.77**
DSPoser (Ours)	1	◯ & $\ddot{✋}$	MAE	◯ & $\tilde{✋}$	$\mathbf{19.09^{\pm0.21}}$	$55.82^{\pm0.27}$

Table 3: Performance comparisons across baseline models on the **AMASS** test set. We report MPJRE [°], MPJPE [cm], and MPJVE [cm/s], with the best results highlighted in **boldface**. Note that ‡ is trained only with dense data without uncertainty.

Methods	**y**	MPJPE	MPJVE	MPJRE	Hand PE	Upper PE	Lower PE
FinalIK [25]	◯ & ✋	18.09	59.24	16.77	-	-	-
LoBSTR [34]	◯ & ✋	9.02	44.97	10.69	-	-	-
VAE-HMD [7]	◯ & ✋	6.83	37.99	4.11	-	-	-
CollMoves [1]	◯ & ✋	5.55	65.28	4.58	-	-	-
AvatarPoser [13]	◯ & ✋	4.20	28.23	3.08	2.34	1.88	8.06
AvatarJLM [36]	◯ & ✋	**3.35**	**20.79**	**2.90**	**1.24**	**1.72**	**6.20**
DSPoser (Ours)‡	◯ & ✋	$3.73^{\pm0.08}$	$43.43^{\pm0.14}$	$2.94^{\pm0.09}$	3.26	1.92	6.53

achieves superior MPJPE compared to AvatarJLM, though it showswlimitations in MPJVE due to the stochasticity of the diffusion model. In the experimental results presented in Table 2, our DSPoser model demonstrates superior performance on the Ego-Exo4D validation set. The model outperforms existing baselines, achieving a lower MPJPE of 16.84 cm, which represents an improvement over the next best model by 5.49 cm. Additionally, DSPoser achieves an MPJVE of 39.86 cm/s, improving upon the basline of naive extension of Bodiffusion by 7.64 cm/s.

It is evident that by incorporating temporally sparse hand pose data, our DSPoser framework significantly enhances pose estimation accuracy. For instance, on the AMASS dataset, MPJPE improved dramatically from 12.08 cm to 5.51 cm, while on the Ego-Exo4D dataset, it improves from 19.12 cm to 16.84 cm in $T_s = 20$ setup. This indicates that even sparse hand trajectory data, when effectively utilized, can provide crucial information for refining the accuracy of ego body pose estimation. Our method's ability to harness sparsely available data underscores its potential in applications where capturing dense sequence is challenging.

4.3 Full Body Pose Estimation from Spatially Sparse data

To demonstrate the versatility of our framework, we conduct experiments on spatially sparse video data. In the temporally dense data setup, where there is no uncertainty regarding hand poses, the dense data directly works as a condition **y** for spatial completion on the right side of Figure 2. Table 3 presents the results, demonstrating that DSPoser performs comparably to baseline models designed specifically for dense data setups on MPJPE and MPJRE metrics, underscoring the versatility of our dual approach in handling dense data scenarios. As discussed in Section 4.2, the higher MPJVE error results from the inherent stochasticity of the diffusion model.

Table 4: Ablation study for uncertainty guidance strategy

Methods	MPJPE	MPJVE	MPJRE
w/o Uncertainty	$6.05^{\pm0.01}$	$30.12^{\pm0.04}$	$4.36^{\pm0.00}$
Sample	$\mathbf{5.51}^{\pm0.02}$	$\mathbf{24.19}^{\pm0.10}$	$\mathbf{4.09}^{\pm0.02}$
Distribution emb.	$5.67^{\pm0.02}$	$25.63^{\pm0.02}$	$4.16^{\pm0.02}$
Dropout	$5.55^{\pm0.02}$	$25.10^{\pm0.02}$	$4.11^{\pm0.02}$

Table 5: Ablation study for different types uncertainty

Methods	MPJPE	MPJVE	MPJRE
w/o Uncertainty	$6.05^{\pm0.01}$	$30.12^{\pm0.04}$	$4.36^{\pm0.00}$
Epistemic	$5.78^{\pm0.03}$	$27.55^{\pm0.09}$	$4.10^{\pm0.02}$
Aleatoric	$\mathbf{5.51}^{\pm0.02}$	$\mathbf{24.19}^{\pm0.10}$	$\mathbf{4.09}^{\pm0.02}$
Total	$5.59^{\pm0.02}$	$25.65^{\pm0.09}$	$4.16^{\pm0.02}$

Table 6: Ablation study for β for uncertainty capturing with MAE.

β	MPJPE (cm)
1.00	11.57
0.50	10.85
0.25	10.92

Table 7: Hand detection accuracy on Ego-Exo4D dataset.

	MPJPE (cm)
Left Hand	9.51
Right Hand	9.63

Table 8: Hand visibility ratio for AMASS and Ego-Exo4D dataset

Num. Hands	0	1	2
AMASS	82.77%	12.04%	5.19%
Ego-Exo4D	72.95%	19.44%	7.61%

4.4 Ablation Studies

Based on the ablation study results shown in Tables 4 and 5, we can analyze the impact of different uncertainty guidance strategies and types of uncertainty on the performance of the model for body pose estimation. The ablation study is conducted with AMASS dataset with the sliding window $T_s = 20$ to better analyze the effect of the uncertainty guidance. Table 4 investigates the effects of various uncertainty guidance strategies, including no uncertainty guidance, sample, distribution embedding, and dropout. The results suggest that incorporating uncertainty guidance through these strategies can improve the model's performance across different metrics. The sampling strategy achieves the best performance, with the lowest MPJPE of 5.51, MPJVE of 24.19, and MPJRE of 4.09, indicating its effectiveness in capturing uncertainty and improving pose estimation accuracy.

Table 5 examines the contributions of different types of uncertainty, including epistemic uncertainty, aleatoric uncertainty, and total uncertainty. The results show that accounting for aleatoric uncertainty leads to the best overall performance. This suggests that considering data uncertainty can provide complementary information and improve the robustness of the pose estimation model. Overall, the ablation study highlights the importance of incorporating uncertainty guidance and considering different types of uncertainty in the model design for accurate and reliable body pose estimation.

In Table 6, we analyzed the effect of different β values on the AMASS dataset during the uncertainty capturing process of the Masked Auto-Encoder (MAE). The results, shown in the table, indicate that $\beta = 0.5$ provides the best temporal completion for head and hand 3D positions from the doubly sparse input. Therefore, we set β to 0.5 for training the MAE.

4.5 Hand Detection Accuracy and Hand Visibility Statistics

We investigate the error of the hand detector applied to the Ego-Exo4D dataset in terms of MPJPE, as shown in Table 7. The detection results indicate an average error of less than 10 cm. We also analyze the visibility statistics for the AMASS and Ego-Exo4D datasets in Table 8. In the AMASS dataset, at least one hand is visible in 18% of all frames with a 90° field of view (FoV), whereas in the Ego-Exo4D dataset, at least one hand is visible in 27% of all frames.

4.6 Qualitative Results

We visualized the aleatoric uncertainty in Figure 3, captured by a model trained using MAE on the AMASS dataset. In cases of partial visibility, as shown in Figure 3 (a-1) and (a-2), the uncertainty range is notably small. Conversely, in frames where the subject is completely obscured, the uncertainty range increases significantly. Even in fully invisible scenarios, the model captures a range of uncertainty, likely influenced by head movements. Most of the estimated frames fall within the $\pm 2\sigma$ range.

We also visualized the qualitative results on the Ego-Exo4D dataset and AMASS dataset in Figure 4. The qualitative results for AMASS show that our method improves the estimation results when sparse
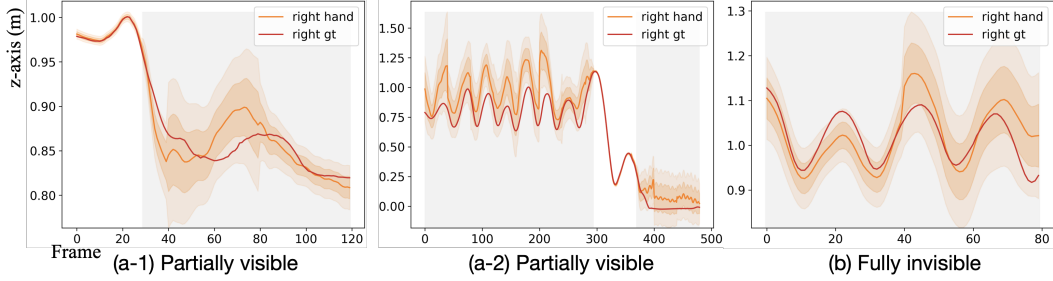
Figure 3: Uncertainty visualization of the right hand pose captured by the MAE. Gray areas represent frames where the hand is invisible, and white areas denote visible frames. We depict aleatoric uncertainty within ranges of $\pm 1\sigma$ and $\pm 2\sigma$ from the estimated μ.
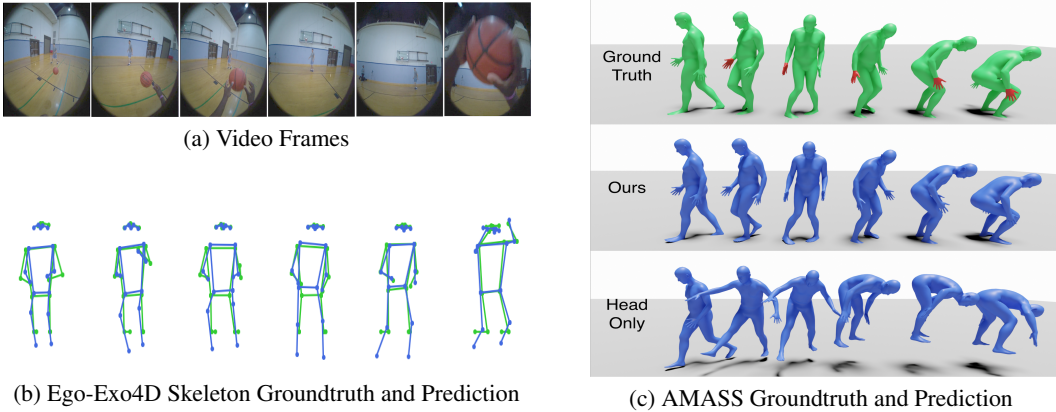

(a) Video Frames

(b) Ego-Exo4D Skeleton Groundtruth and Prediction

(c) AMASS Groundtruth and Prediction

Figure 4: (a) Ego-Exo4D video frames, (b) the corresponding skeleton ground truth and our prediction results, and (c) qualitative results on AMASS data under different input conditions. green indicates the ground truth, blue indicates the predicted result, and red indicates the visible hands. Head only estimates body pose from head trajectories, whereas Ours estimates body pose from imputed hand and head trajectories.

hand information is available, compared to the Head Only results. Additionally, in the Ego-Exo4D results, the hands are more aligned compared to the lower body when hands are available.

5 Related Works

Human Pose Estimation from Sparse Input A common capture setting in mixed reality involves using a head-mounted device and hand controllers. Estimating full-body motion from the sparse input of head and hand movements is challenging. Recently, several methods have been proposed to tackle this: AvatarPoser [13] is the first learning-based method to predict full-body poses in world coordinates using only head and hand motion inputs. It uses a Transformer encoder to extract deep features and decouples global motion from local joint orientations, refining arm positions with inverse kinematics for accurate full-body motion. BoDiffusion [3] employs a generative diffusion model for motion synthesis, addressing the under-constrained reconstruction problem. It uses a time and space conditioning scheme to leverage sparse tracking inputs, generating smooth and realistic full-body motion sequences. AvatarJLM [36] uses a two-stage framework where sparse signals are embedded into high-dimensional features and processed by an MLP to generate joint-level features. These features are then converted into tokens and fed into a transformer-based network to capture spatial and temporal dependencies, with an SMPL regressor transforming them into 3D full-body pose sequences. HMD-poser [6] combines a lightweight temporal-spatial feature learning network with regression layers and uses forward kinematics to achieve real-time human motion tracking. AGRoL [8] utilized conditional diffusion model to generate full body pose from sparse upper-body tracking signals. It is worth noting a concurrent work, EgoPoser [12], which also addresses ego body pose estimation

from doubly sparse observations. Their focus lies in preparing training data through field-of-view (FoV) modeling rather than introducing new algorithms. Our work is orthogonal to theirs, providing algorithmic contributions through a multi-stage pipeline including an uncertainty-aware masked auto-encoder (MAE).

Human Body Pose Estimation from Egocentric Videos   Estimating full 3D human body pose from egocentric videos is an ill-posed problem due to the partial visibility of wearer's body parts from the camera mounted on wearer's head. Recently, several approaches have been proposed to address this challenge. EgoEgo [16] integrates SLAM and a learned transformer to estimate head motion, then leverages estimated head pose to generate plausible full-body motions using diffusion models. [18] designs a kinematic policy to generate per-frame target motion from egocentric inputs, and leverages a pre-learned dynamics model to distill human dynamics information into the kinematic model. GIMO [37] integrates motion, 3D eye gaze, and 3D scene features to generate gaze informed long term intention-aware human motion prediction. [32] leverages external camera to generate pseudo labels to estimate full 3D body pose from single head mounted fish eye camera using weak supervision. [33] estimates geometry of surrounding objects and extracts 2D body pose features using EgoPW [32] to regress 3D body pose with a voxel-to-voxel network [20].

6   Conclusion

In this paper, we have addressed the problem of egocentric body pose estimation using temporally sparse observations from head-mounted displays (HMDs). By leveraging both temporal and spatial completion, our approach effectively utilizes intermittent hand pose detections from egocentric videos, alongside consistently available head pose data, to reconstruct full-body motions. Through comprehensive experiments on datasets such as AMASS and Ego-Exo4D, we have demonstrated the effectiveness of our framework. Our results indicate significant improvements over existing methods, particularly in scenarios where dense sensor data may not be available or practical. This advancement opens up new possibilities for beneficial augmented reality experiences in various applications, including sports training by providing feedback on body mechanics, and other scenarios where users need to move freely without additional sensors such as hand controllers. However, our method has not been explicitly tested for fairness across different demographic groups. Potential biases in the datasets used could result in uneven performance across various user populations. Careful curation of training datasets is necessary to prevent unfair failures for underrepresented groups.

7   Limitations

While our proposed method for estimating the body movements of a camera wearer from sparse tracking signals shows promising results, several limitations should be acknowledged. Firstly, our method has been tested with only one type of sparse body part tracking signal, specifically the hand. Incorporating the detection of other body parts, such as feet and elbows, may improve overall body pose estimation. Additionally, variations in lighting, occlusions, and the quality of the egocentric video can impact the accuracy of hand pose detection, subsequently affecting the overall body pose estimation.

The effectiveness of our method was validated using the AMASS and Ego-Exo4D datasets. Although these datasets are comprehensive, they may not encompass the full spectrum of possible real-world variations. Our study focused on pose estimation within a window size of less than a few seconds, following standard settings from the literature. It remains unclear how our method will perform with larger window sizes. Furthermore, the scalability of our method with larger datasets has not been thoroughly evaluated. The use of diffusion models for pose estimation may limit its utility for real-time applications due to their inference speed. Additionally, using multiple models to compute epistemic uncertainty can be computationally intensive.

References

[1] K. Ahuja, E. Ofek, M. Gonzalez-Franco, C. Holz, and A. D. Wilson. Coolmoves: User motion accentuation in virtual reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–23, 2021.

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] A. Castillo, M. Escobar, G. Jeanneret, A. Pumarola, P. Arbeláez, A. Thabet, and A. Sanakoyeu. Bodiffusion: Diffusing sparse observations for full-body human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4221–4231, 2023.

[4] S. Chi, H. Chi, H. Ma, N. Agarwal, F. Siddiqui, K. Ramani, and K. Lee. M2d2m: Multi-motion generation from text with discrete diffusion models. In *European conference on computer vision*. Springer, 2024.

[5] CMU Graphics Lab. Cmu graphics lab motion capture database. http://mocap.cs.cmu.edu/, 2000.

[6] P. Dai, Y. Zhang, T. Liu, Z. Fan, T. Du, Z. Su, X. Zheng, and Z. Li. Hmd-poser: On-device real-time human motion tracking from scalable sparse observations. *arXiv preprint arXiv:2403.03561*, 2024.

[7] A. Dittadi, S. Dziadzio, D. Cosker, B. Lundell, T. J. Cashman, and J. Shotton. Full-body motion from a single head-mounted device: Generating smpl poses from partial observations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11687–11697, 2021.

[8] Y. Du, R. Kips, A. Pumarola, S. Starke, A. Thabet, and A. Sanakoyeu. Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2023.

[9] K. Grauman, A. Westbury, L. Torresani, K. Kitani, J. Malik, T. Afouras, K. Ashutosh, V. Baiyya, S. Bansal, B. Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. *arXiv preprint arXiv:2311.18259*, 2023.

[10] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022.

[11] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[12] J. Jiang, P. Streli, M. Meier, and C. Holz. Egoposer: Robust real-time ego-body pose estimation in large scenes. In *European conference on computer vision*. Springer, 2024.

[13] J. Jiang, P. Streli, H. Qiu, A. Fender, L. Laich, P. Snape, and C. Holz. Avatarposer: Articulated full-body pose tracking from sparse motion sensing. In *European conference on computer vision*, pages 443–460. Springer, 2022.

[14] T. Jiang, P. Lu, L. Zhang, N. Ma, R. Han, C. Lyu, Y. Li, and K. Chen. Rtmpose: Real-time multi-person pose estimation based on mmpose. *arXiv preprint arXiv:2303.07399*, 2023.

[15] K. Lee, H. Chang, L. Jiang, H. Zhang, Z. Tu, and C. Liu. ViTGAN: Training GANs with vision transformers. In *International Conference on Learning Representations*, 2022.

[16] J. Li, K. Liu, and J. Wu. Ego-body pose estimation via ego-head pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17142–17151, 2023.

[17] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023.

[18] Z. Luo, R. Hachiuma, Y. Yuan, and K. Kitani. Dynamics-regulated kinematic policy for egocentric pose estimation. *Advances in Neural Information Processing Systems*, 34:25019–25032, 2021.

[19] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.

[20] G. Moon, J. Y. Chang, and K. M. Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018.

[21] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database hdm05. *Computer Graphics Technical Report CG-2007-2, Universität Bonn*, 7:11, 2007.

[22] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. 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.

[23] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. 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.

[24] Y. Rong, T. Shiratori, and H. Joo. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1749–1759, 2021.

[25] RootMotion. Final ik. https://assetstore.unity.com/packages/tools/animation/final-ik-14290, 2018.

[26] M. Seitzer, A. Tavakoli, D. Antic, and G. Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. *arXiv preprint arXiv:2203.09168*, 2022.

[27] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[28] B. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11:1517–1561, Apr. 2010.

[29] N. F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of vision*, 2(5):2–2, 2002.

[30] M. Valdenegro-Toro and D. S. Mori. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1508–1516. IEEE, 2022.

[31] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[32] J. Wang, L. Liu, W. Xu, K. Sarkar, D. Luvizon, and C. Theobalt. Estimating egocentric 3d human pose in the wild with external weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13157–13166, 2022.

[33] J. Wang, D. Luvizon, W. Xu, L. Liu, K. Sarkar, and C. Theobalt. Scene-aware egocentric 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13031–13040, 2023.

[34] D. Yang, D. Kim, and S.-H. Lee. Lobstr: Real-time lower-body pose prediction from sparse upper-body tracking signals. In *Computer Graphics Forum*, volume 40, pages 265–275. Wiley Online Library, 2021.

[35] J. Zhang, Y. Zhang, X. Cun, Y. Zhang, H. Zhao, H. Lu, X. Shen, and Y. Shan. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14730–14740, 2023.

[36] X. Zheng, Z. Su, C. Wen, Z. Xue, and X. Jin. Realistic full-body tracking from sparse observations via joint-level modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14678–14688, 2023.

[37] Y. Zheng, Y. Yang, K. Mo, J. Li, T. Yu, Y. Liu, C. K. Liu, and L. J. Guibas. Gimo: Gaze-informed human motion prediction in context. In *European Conference on Computer Vision*, pages 676–694. Springer, 2022.

A Additional Details

A.1 Architecture & Experimental Details

Module	# of Params	MACs	Time
VQ-VAE	17.9 M	3.6 G	3 ms
MAE	51.3 M	23.3 G	4 ms
VQ-Diffusion	74.2 M	1190.2 G	958 ms

Table 9: Experimental results for different modules

		Training step				Times
		25	33	50	100	(ms)
Inference	25	5.83	5.92	5.69	8.72	278
	33	–	5.67	5.63	5.58	348
	50	–	–	5.61	5.53	522
	100	–	–	–	5.51	1013

Table 10: Performance results for different inference and training step combinations

VQ-VAE   We adhered to the architectural details and training protocol of Zhang et al. [35], with modifications including setting both the encoder and decoder stride to 1, and adjusting the window size to 40. For the Ego-Exo4D dataset, we employed wing loss with a width of 5 and a curvature of 4. For AMASS, we opted for L2 loss. Additionally, to generate smooth motion, we applied both velocity and acceleration losses, assigning weights of 10 for each in the AMASS dataset, and weights of 1 for each in the Ego-Exo4D dataset. The input shape is represented as $T_w \times J \times D_{\text{data}}$, where J is the number of joints, T_w is time window, and D_{data} is the data representation. For AMASS, which uses a 6D rotation representation, $J = 22$ and $D_{data} = 6$. For Ego-Exo4D, $J = 17$ and $D_{data} = 3$. We use $T_w = 40$ for both AMASS and Ego-Exo4D.

VQ-Diffusion   We employed the same hyperparameters and training specifications as Gu et al. [10] for training VQ-Diffusion. Additionally, we replaced the absolute positional encoding with relative positional encoding, following the implementation of VQ-Diffusion for human motion generation proposed by Chi et al. [4]. We replaced the text condition module with the uncertainty-aware MAE module to feed the imputed trajectory and uncertainty as a conditional input.

Masked Auto-Encoder   We adapted the encoder to accommodate a variable number of visible hands and modified the last projection layer to guide the uncertainty, as detailed in the main paper. Otherwise, we followed the training details provided by He et al. [11]. We trained 4 models to measure the uncertainty, $M = 4$.

Dataloader   We adopted the dataloader of AMASS and evaluation configurations from Bodiffusion [3] for our experiments. For the Ego-Exo4D dataset, we employed the dataset implementation from the Ego-Exo4D [9].

Analaysis on Computaional Cost   We evaluated the computational cost of our approach by analyzing the number of parameters, multiply-accumulate operations (MACs), and inference time for each module in our pipeline. The reported times in Table 10 represent the total measured time for the entire pipeline, whereas Table 9 measures the time only for the corresponding module, excluding overhead between modules. As shown in Table 9, the VQ-Diffusion model is responsible for the majority of MACs and inference time. To mitigate this, we conducted further experiments to explore ways of reducing the VQ-Diffusion model's inference time. Table 10 presents the trade-off between performance and inference time based on the number of diffusion steps, offering multiple options. Notably, training with 50 steps and inferring with 25 steps yields approximately a 4x faster inference time with only about 3% reduction in performance.

B Compute Resource

We ran our experiments on one workstation, containing AMD Ryzen Threadripper PRO 7975WX, DDR5 RAM 256GB and 4 NVIDIA GeForce RTX 4090. AMASS dataset takes 512GB of storage and Ego-Exo4D dataset takes 11TB. One training run took around 18 hours on 1 GPU for AMASS and around 12 hours for Ego-Exo4D. Inference over AMASS validation set takes 40 minutes on 1 GPU and inference over Ego-Exo4D validation set takes 10 minutes on 1 GPU. In total, all experiments including preliminary or failed experiments took approximately 300 GPU-hours.

C   Licenses for Assets Used in the Paper

Code   We use the code of BoDiffusion [3] which is available at https://github.com/BCV-Uniandes/BoDiffusion.   Unfortunately, we could not locate the licensing terms for the source code.   For the Masked Auto Encoder, we use the implementation available at https://github.com/pengzhiliang/MAE-pytorch, but we could not find the licensing terms for this source code.

We also employ VQ-Diffusion [10], available at https://github.com/cientgu/VQ-Diffusion/tree/main?tab=readme-ov-file, which is licensed under Microsoft's Open Source Program.

For VQ-VAE, we use the implementation from T2M-GPT [35], which can be found at https://github.com/Mael-zys/T2M-GPT, and is licensed under the Apache License 2.0.

For 3D hand detection, we use the code of FrankMoCap [24]: https://github.com/facebookresearch/frankmocap, which is licensed under the CC BY-NC 4.0 license. We also used RTM-pose [14], which is available at https://github.com/open-mmlab/mmpose under the Apache License 2.0.

Data   We use the Ego-Exo4D dataset [9] https://ego-exo4d-data.org, which is licensed under a custom (commercial or non-commercial) license. We also use AMASS [19] https://amass.is.tue.mpg.de, which is licensed under a custom (non-commercial scientific research) license.

D   Preliminary

D.1   Discrete Diffusion Model

Discrete diffusion models [10] represent a category of diffusion models that progressively introduce noise into data while training to reverse this process. In contrast to continuous models, such as a latent diffusion model [23], which manipulate data in a continuous state space, discrete diffusion models operate within discrete state spaces.

VQ-VAE   Vector Quantized-Variational Autoencoder (VQ-VAE) [31] is a generative model that extends the concept of Variational Autoencoders (VAEs) by incorporating discrete latent representations via vector quantization. The encoder $E(\mathbf{x})$ compresses input data $\mathbf{x}$ into discrete latent vectors by mapping each encoded representation to the closest vector $\mathbf{z}_q$ to the nearest codebook entry from a learned codebook of prototypes using the nearest-neighbor search: $\mathbf{z}_q = Q(\mathbf{z}) = \mathrm{argmin}_{\mathbf{c}_i \in \mathcal{C}} ||\mathbf{z} - \mathbf{c}_i||_2$. Here, $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$, where K is the total number of codebooks. The decoder $D(\mathbf{z}_q)$ reconstructs the input data $\mathbf{x}$ from these quantized vectors, yielding a reconstructed output $\hat{\mathbf{x}} = D(\mathbf{z}_q)$. The optimization process involves minimizing a combination of reconstruction loss and commitment loss. The reconstruction loss is expressed as $||\mathbf{x} - \hat{\mathbf{x}}||_2$, while the commitment loss ensures the encoder commits to the nearest prototype in the codebook: $||\mathrm{sg}[\mathbf{z_q}] - \mathbf{z}||_2$, where sg is the stop-gradient operator. The overall loss function, which the VQ-VAE model minimizes, is: $\mathcal{L}_{\mathrm{VQ}} = ||\mathbf{x} - \hat{\mathbf{x}}||_2 + ||\mathbf{z_q} - \mathrm{sg}[\mathbf{z}]||_2 + \lambda_{\mathrm{VQ}}||\mathrm{sg}[\mathbf{z_q}] - \mathbf{z}||_2$. Here, λ_{VQ} is a coefficient for the commitment loss.

Forward Diffusion Process.   Building on the foundation laid by the discrete diffusion models introduced by [27], VQ-Diffusion [10] refined the diffusion process with a mask-and-replace strategy. In VQ-Diffusion, during the forward diffusion process, tokens can either transition to other tokens or to a special <MASK> token. The transition probability from token $\mathbf{z}^i$ to $\mathbf{z}^j$ at diffusion step t is defined by the matrix $\mathbf{Q}_t[i,j]$. The transition matrix $\mathbf{Q}_t$, structured in $\mathbb{R}^{(K+1)\times(K+1)}$, follows:

$$\mathbf{Q}_t = \left[\begin{array}{c|c} \hat{\mathbf{Q}}_t & 0 \\ \hline \gamma_t \cdot \mathbf{1}^\top & 1 \end{array} \right], \text{where } \hat{\mathbf{Q}}_t = \alpha_t \mathbf{I} + \beta_t \mathbf{1} \mathbf{1}^\top. \tag{7}$$

Here, α_t adjusts to ensure conservation of probability, such that $\alpha_t = 1 - K\beta_t - \gamma_t$ is the probability of transitioning between tokens, and γ_t governs transitions to the <MASK> token. The transition from step $t-1$ to t is expressed as: $q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \boldsymbol{v}^\top(\mathbf{z}_t)\boldsymbol{Q}_t\boldsymbol{v}(\mathbf{z}_{t-1})$, where $\boldsymbol{v}(\mathbf{z}_t) \in \mathbb{R}^{(K+1)\times 1}$ is an one-hot encoded vector representing the token index of $\mathbf{z}_t$. Using the Markov property, the

probability of transitioning from any initial step 0 to step t is $q(\mathbf{z}_t|\mathbf{z}_0) = \boldsymbol{v}^\top(\mathbf{z}_t)\overline{\boldsymbol{Q}}_t\boldsymbol{v}(\mathbf{z}_0)$, where $\overline{\mathbf{Q}}_t = \mathbf{Q}_t\mathbf{Q}_{t-1}\cdots\mathbf{Q}_1$ is the cumulative transition matrix. This defines the cumulative probabilities as $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i, \bar{\gamma}_t = 1 - \prod_{i=1}^t(1-\gamma_i)$, and $\bar{\beta}_t = (1-\alpha_t-\gamma_t)/K$.

Conditional Denoising Process. In the conditional denoising process, a neural network denoted as p_θ aims to predict the original, noiseless token $\mathbf{z}_0$ given a corrupted token and the associated condition, such as a embedded hand trajectories. The posterior distribution for the discrete diffusion process can be defined as:

$$q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0) = \frac{q(\mathbf{z}_{t-1}|\mathbf{z}_0)q(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{z}_0)}{q(\mathbf{z}_t|\mathbf{z}_0)}$$
$$= \frac{\left(\boldsymbol{v}^\top(\mathbf{z}_t)\boldsymbol{Q}_t\boldsymbol{v}(\mathbf{z}_{t-1})\right)\left(\boldsymbol{v}^\top(\mathbf{z}_{t-1})\overline{\boldsymbol{Q}}_{t-1}\boldsymbol{v}(\mathbf{z}_0)\right)}{\boldsymbol{v}^\top(\mathbf{z}_t)\overline{\boldsymbol{Q}}_t\boldsymbol{v}(\mathbf{z}_0)}. \tag{8}$$

With this, the reverse transition distribution is determined as:

$$p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{y}) = \sum_{\tilde{\mathbf{z}}_0=1}^K q(\mathbf{z}_{t-1}|\mathbf{z}_t, \tilde{\mathbf{z}}_0)p_\theta(\tilde{\mathbf{z}}_0|\mathbf{z}_t, \mathbf{y}), \tag{9}$$

where the network iteratively denoises tokens from step T down to 1, eventually generating the token $\mathbf{z}_0$ conditioned on $\mathbf{y}$. To train the network p_θ, the training approach includes not only a denoising objective but also the standard variational lower bound objective [27], denoted as $\mathcal{L}_{\text{vlb}}$. The comprehensive training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{vlb}} + \lambda\mathbb{E}_{\mathbf{z}_t \sim q(\mathbf{z}_t|\mathbf{z}_0)}[-\log p_\theta(\mathbf{z}_0|\mathbf{z}_t, \mathbf{y})], \tag{10}$$

where λ is the coefficient for the denoising loss.

E   Additional Experimental results

We illustrate additional uncertainty visualization on Fig. 5. In addition, we demonstrate the additional qualitative results with our method on Ego-Exo4D dataset and AMASS dataset in Fig. 6 and Fig. 7 & 8, respectively.
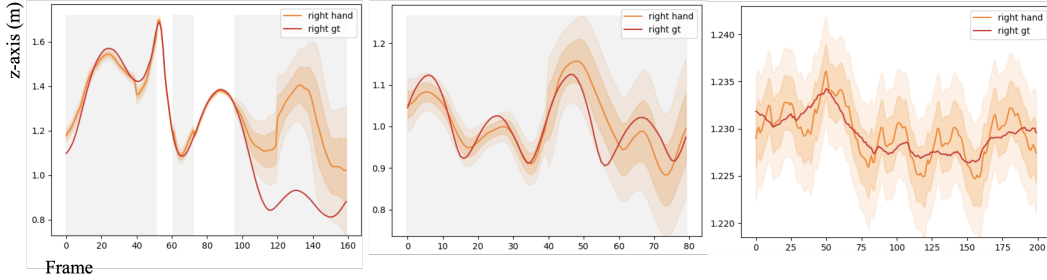

Figure 5: Additional uncertainty visualization of the right hand pose captured by the MAE. Gray areas represent frames where the hand is invisible, and white areas denote visible frames. We depict aleatoric uncertainty within ranges of $\pm 1\sigma$ and $\pm 2\sigma$ from the estimated μ.
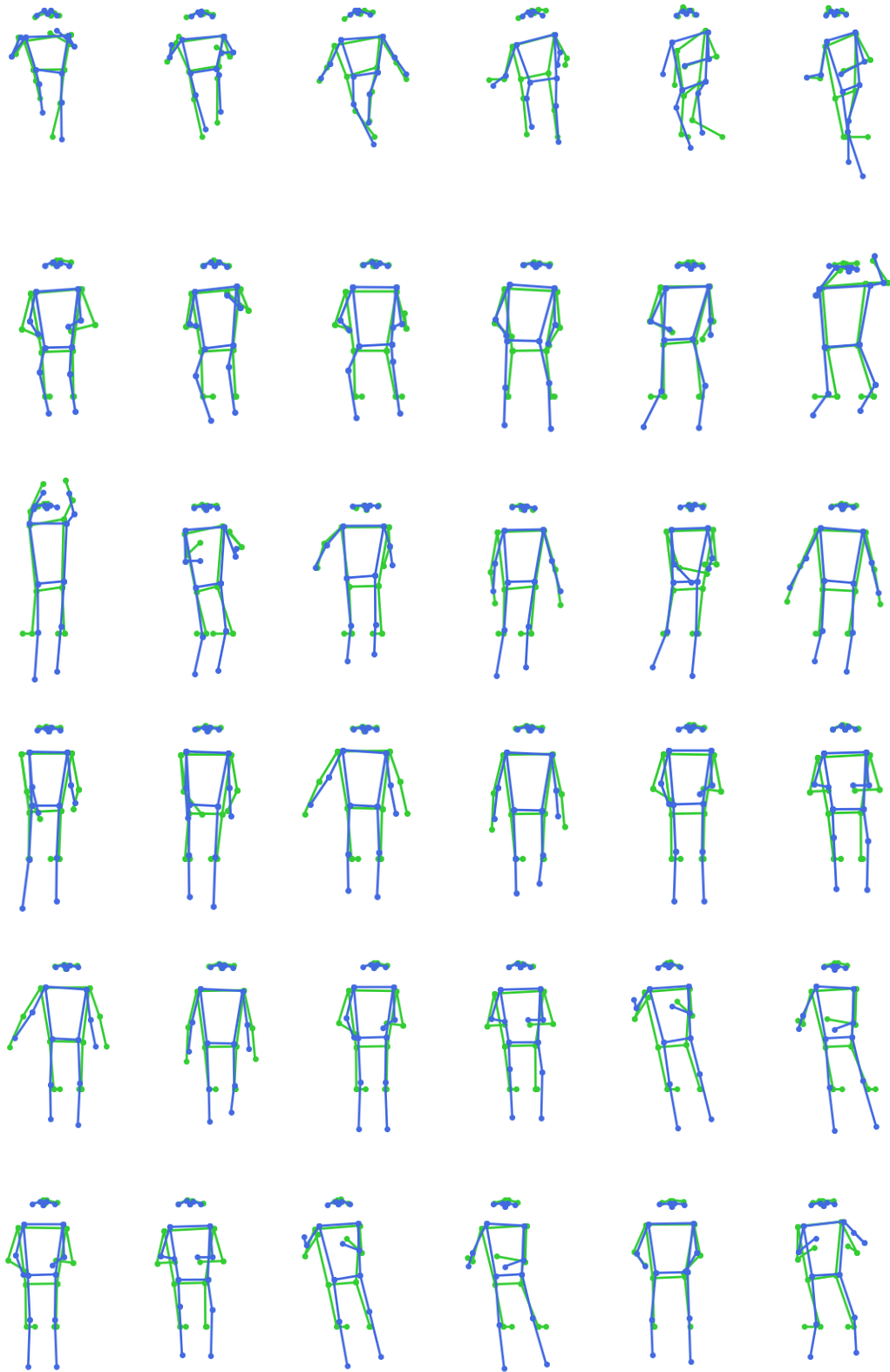
Figure 6: Qualitative results showing the groundtruth in green and predicted human pose in blue using our method on the Ego-Exo4D dataset.
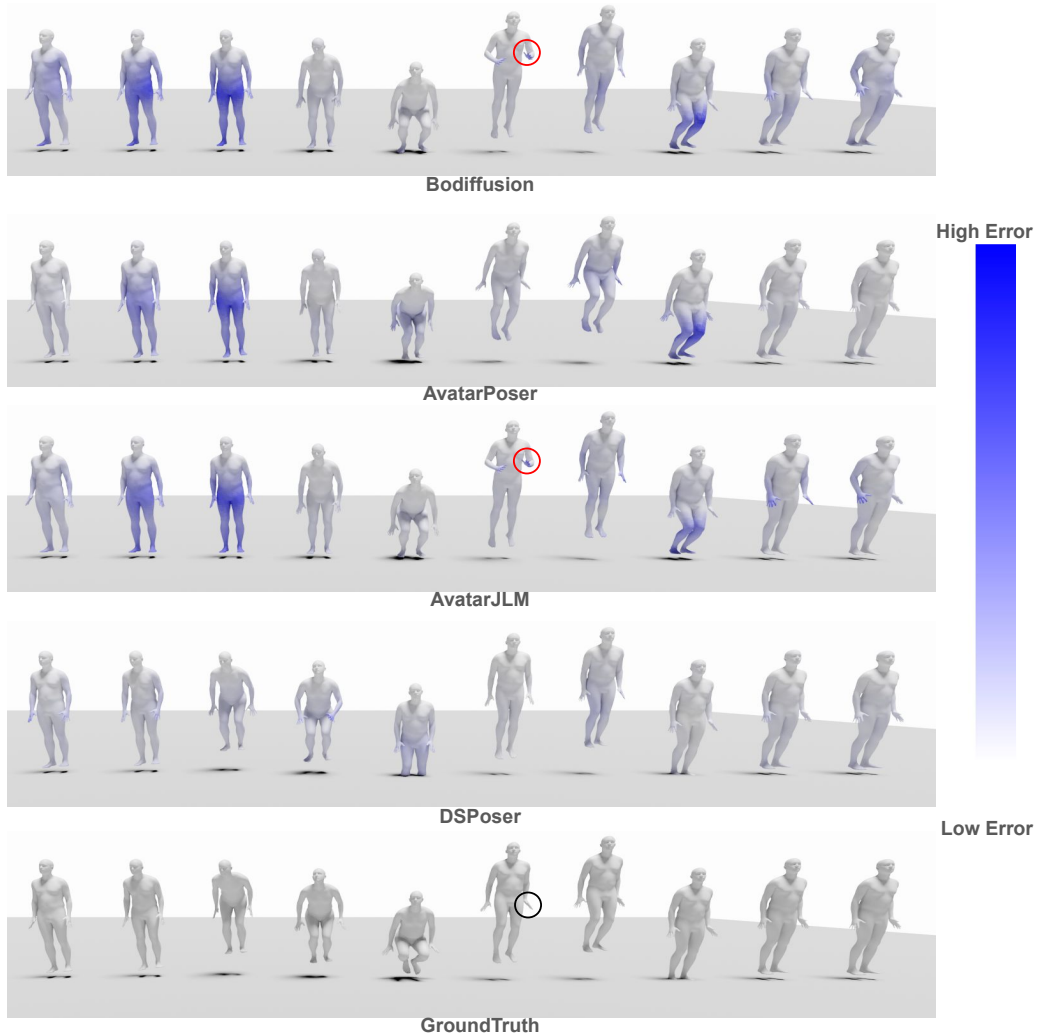
Figure 7: Qualitative results on AMASS dataset comparing DSPoser (Ours) against the baselines. Color gradient indicates an absolute positional error, with a higher error corresponding to higher blue intensity. Results demonstrate that motions generated by DSPoser exhibit greater similarity to the ground truth. Furthermore, it highlights higher errors (indicated with red circles) for baselines when the hand is occluded in the ground truth pose (indicated with a black circle).
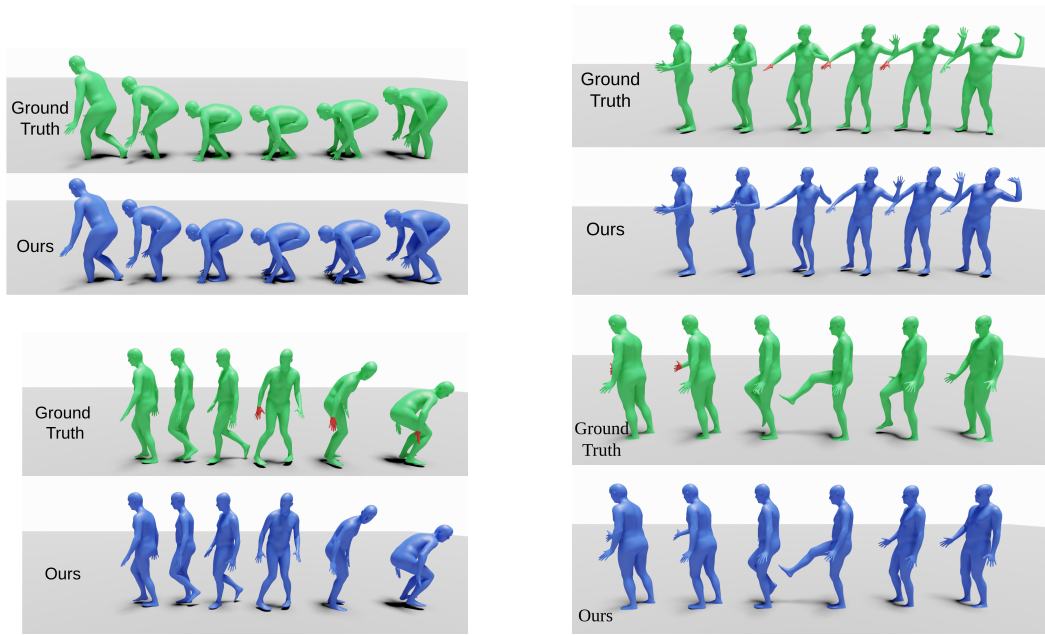
Figure 8: Qualitative results showing the groundtruth in Green and predicted human pose in blue using our method on AMASS dataset, with red indicating the visible hands.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims accurately reflect this paper's empirical contributions. The introduction section discusses contributions made in the paper and important assumptions and limitations. Limitations of our work are described in Section 7.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of our work are described in Section 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Implementation details are explained in Section 3 and Section A.1

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code release is challenging due to our organization's policy.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We specified the experimental setting and details in Section A.1 and Section 4.1.

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report 95% confidence interval in the experimental tables in Section 4. The confidence interval was computed using closed form formula. The variability is due to random drawing of noise vectors in denoising diffusion and uncertainty guidance.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Compute resources used in this paper can be found in Section B of the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: Our research conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes] ,

Justification: Both potential positive societal impacts and negative societal impacts of the work are discussed in Section 6.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We believe this paper does not pose a high risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Citations and licenses of assets used in this paper can be found in Section C of the Appendix.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.