Di^{2}Pose: Discrete Diffusion Model for Occluded 3D Human Pose Estimation

Weiquan Wang[1], **Jun Xiao**[1], **Chunping Wang**[2], **Wei Liu**[3], **Zhao Wang**[1], **Long Chen**[4*]
[1]Zhejiang University [2]Finvolution Group [3]Tencent
[4]Hong Kong University of Science and Technology
{wqwangcs, junx}@zju.edu.cn, wangchunping02@xinye.com,
wl2223@columbia.edu, zhao_wang@zju.edu.cn, longchen@ust.hk

Abstract

Diffusion models have demonstrated their effectiveness in addressing the inherent uncertainty and indeterminacy in monocular 3D human pose estimation (HPE). Despite their strengths, the need for large search spaces and the corresponding demand for substantial training data make these models prone to generating biomechanically unrealistic poses. This challenge is particularly noticeable in occlusion scenarios, where the complexity of inferring 3D structures from 2D images intensifies. In response to these limitations, we introduce the **Di**screte **Di**ffusion **Pose** (**Di^{2}Pose**), a novel framework designed for occluded 3D HPE that capitalizes on the benefits of a discrete diffusion model. Specifically, Di^{2}Pose employs a two-stage process: it first converts 3D poses into a discrete representation through a *pose quantization step*, which is subsequently modeled in latent space through a *discrete diffusion process*. This methodological innovation restrictively confines the search space towards physically viable configurations and enhances the model's capability to comprehend how occlusions affect human pose within the latent space. Extensive evaluations conducted on various benchmarks (*e.g.*, Human3.6M, 3DPW, and 3DPW-Occ) have demonstrated its effectiveness.

1 Introduction

3D Human Pose Estimation (HPE) from monocular images remains a challenging yet pivotal research in the realm of computer vision, boasting a wide range of applications including human-machine interaction, autonomous driving, and animations [57, 81, 5, 70]. Generally, the mainstream approaches, including Direct Estimation [68, 43, 47] and 2D-to-3D Lifting [87, 51, 86], aim to perform 3D HPE by either directly predicting 3D poses from 2D images or lifting detected 2D poses into 3D space. These approaches aim to address the inherent 2D-3D ambiguity in 3D HPE tasks by learning mapping from training data. Despite significant advancements, accurately estimating 3D poses from monocular images remains a formidable challenge, particularly when humans are partially occluded [39]. Such occlusions introduce considerable uncertainty and indeterminacy into the estimation process.

Existing 3D HPE methods try to handle the occlusion challenges with pose priors/constraints [58, 62] or data augmentation strategies (*e.g.*, annotations augmentation [61], pose transformation [35], and differentiable operations [82]). However, due to the inherent discreteness of 3D poses (primarily defined by discrete anatomical landmarks), these methods tend to represent poses using coordinate vectors or heatmap embeddings, treating joints as independent units and overlooking the interdependencies among body joints. Recent research [21] has introduced a compositional pose representation that captures the dependencies among joints by converting a pose into multiple tokens, enabling the

*Long Chen is the corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

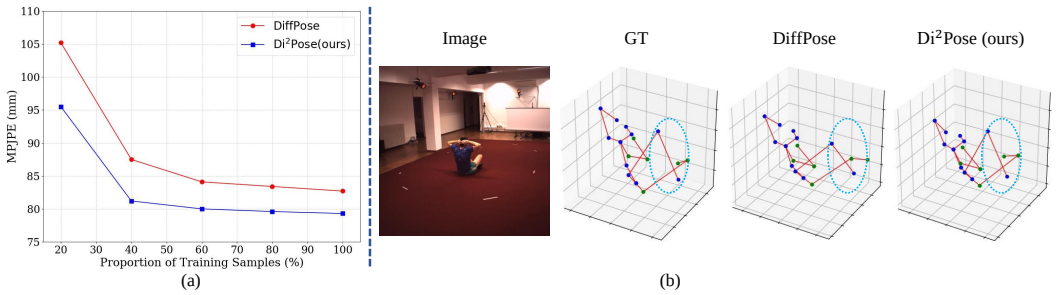


Figure 1: (a) Results of DiffPose [22] and Di^{2}Pose in Human3.6M [72] dataset (with MPJPE metric), across varying proportions of training samples. (b) Prediction results of two methods under occlusion.

use of mutual context between joints. This approach, which learns from real pose datasets, results in each learned token corresponding to a physically realistic prototype. Nevertheless, Geng *et al.* [21] casts HPE to a classification task, where the system simply classifies tokens based on prototype poses. Unfortunately, such scheme does not account for the effects of occlusions in the estimation process, potentially leading to inaccuracies due to unresolved uncertainty and indeterminacy.

Recent studies have shown marked progress in 3D HPE via generative models [2, 84, 19, 27]. Notably, diffusion models [25] have demonstrated effectiveness in handling complex and uncertain data distributions, making them suitable for handling uncertainty and indeterminacy in 3D HPE tasks [19, 91, 66, 27, 37]. They excel at generating samples that conform to a target data distribution by iteratively removing noise through a series of diffusion steps, ultimately predicting more accurate 3D poses. However, these diffusion-based 3D HPE methods initialize the 3D pose from random noise at the begining of the diffusion process, where each joint can be sampled from the continuous 3D space. Since the continuous 3D space has an infinite number of points, training such diffusion-based models requires a large amount of 3D pose data to achieve optimal outcomes [23, 75, 3]. This demand implies a substantial need for training data, presenting a stark contradiction to the limited availability of 3D human pose datasets. As illustrated in Figure 1(a), the predictive performance of DiffPose [22] declines more rapidly as the proportion of training data decreases. Given the scarcity of 3D pose training data, previous diffusion models may generate physically implausible configurations that do not adhere to human biomechanics, leading to inaccurate human pose estimations, particularly in occluded scenes (*c.f.*, Figure 1(b) with DiffPose).

In this paper, we propose a novel framework for 3D HPE with occlusions: **Di**screte **Di**ffusion **Pose** (**Di^{2}Pose**), drawing on compositional pose representation and diffusion model to achieve the best of two worlds. Specifically, Di^{2}Pose employs a two-stage approach: *a pose quantization step* followed by *a discrete diffusion process*. The pose quantization step leverages the discrete nature of 3D poses and represents them as quantized tokens by capturing the local interactions between joints. This step effectively confines the search space to physically plausible configurations by learning from real 3D human poses. Subsequently, the discrete diffusion process models the quantized pose tokens in the latent space through a conditional diffusion model. By integrating the forward and reverse processes, our framework adeptly simulates the transition of a 3D pose from occluded to recovered. By modeling occlusion implicitly within the latent space, Di^{2}Pose enhances its understanding of how occlusions affect human poses, providing valuable insights during the training phase.

For the pose quantization step, we devise a pose quantization step inspired by VQ-VAE [71], consisting of a pose encoder, a quantization process, and a pose decoder. To effectively capture local interactions between 3D joints, we introduce the Local-MLP block for both pose encoder and decoder. Within each Local-MLP block, a simple Joint Shift operation integrates information from different joints. The pose encoder utilizes several Local-MLP blocks to convert a 3D pose into multiple rich token features, each representing a sub-structure of the overall pose. These tokens are quantized using a shared codebook, yielding corresponding discrete indices. Additionally, we implement the finite scalar quantization (FSQ) [49] to address the codebook collapse issue observed in traditional VQ-VAE methods [59, 89, 56, 42]. This strategy ensures that the generated codewords are meaningful, a crucial aspect for the subsequent success of the discrete diffusion process.

For the discrete diffusion process, during the training phase, we introduce `occlude` and `replace` strategies to model the quantized pose tokens, enabling the discrete diffusion model to predict occluded tokens and update potential tokens. The occluded token represents the occlusion of the corresponding sub-structure of the 3D pose. The token replacement mechanism is designed to enhance the diversity of potential sub-structures, reflecting the indeterminacy in occluded parts. During the inference phase, pose tokens are either occluded or initialized randomly. The denoising diffusion process estimates the probability density of pose tokens step-by-step based on the input 2D image until the tokens are completely reconstructed. Each step leverages contextual information from all tokens of the entire pose as predicted in the previous step, facilitating the estimation of a new probability density distribution and the prediction of the current step's tokens. This sequential approach ensures a detailed and accurate reconstruction of 3D poses from occluded scenes.

We extensively evaluate our approach in 3D HPE on three challenging benchmarks (Human3.6M [34], 3DPW [72] and 3DPW-Occ). Di^{2}Pose consistently yields lower errors compared to state-of-the-art methods. In particular, it achieves significantly better results when evaluated on occluded scenarios, verifying its advantages of occlusion-handling capability. Our contributions are threefold:

- We propose the Di^{2}Pose framework, which integrates the inherent discreteness of 3D pose data into the diffusion model, offering a new paradigm for addressing 3D HPE under occlusions.
- The designed pose quantization step represents 3D poses in a compositional manner, effectively capturing local correlations between joints and confining search space to reasonable configurations.
- The constructed discrete diffusion process simulates the complete process of a 3D pose transitioning from occluded to recovered, which introduces the impact of occlusions into pose estimation process.

2 Related Work

Monocular 3D HPE. Existing approaches can generally be classified into frame-based and video-based methodologies. **Frame-based methods** predict the 3D pose from a single RGB image, employing different networks in various studies [18, 20, 52, 54, 55] to directly output the human pose from the 2D image. Alternatively, a significant number of studies [48, 77, 85, 88] initially determine the 2D pose, which subsequently forms the foundation for inferring the 3D pose. In contrast, **video-based methods** leverage temporal relationships across video frames. Such methods predominantly [9, 11, 15, 31, 65, 73, 76] commence with the extraction of 2D pose sequences using a 2D pose detector from the video clips, aiming to harness the essential spatio-temporal data for 3D pose estimation. To validate the efficacy of our approach, we evaluate our Di^{2}Pose on the more challenging frame-based setting, wherein the 3D human pose is directly inferred from a 2D image.

Occluded 3D HPE. Occlusions significantly challenge 3D HPE. As evidenced by research [76, 58, 62], pose priors and constraints have been proven crucial for mitigating such issue. Approaches typically involve statistical models to deduce occluded parts from visible cues [76, 41, 44] or pre-defined rules to constraint poses [61, 1]. Moreover, due to the scarcity of 3D pose data, data augmentation, including synthetic occlusions [7, 38, 60, 64, 13] and pose transformations [35, 82], remains vital for enhancing model robustness. Diverging from these aforementioned methods, our method innovatively introduces occlusion in the latent space without extra priors or explicit augmentations, providing a deeper feature-based understanding of occlusion's effects on pose estimation.

Diffusion Models for 3D HPE. Recent advancements have shown that diffusion models are capable of managing complex and uncertain data distributions [25, 26, 17, 4, 32, 8, 10, 80, 40, 36, 74, 78], which is particularly beneficial for 3D HPE. Typically, these models predict 3D poses by progressively refining the pose distribution from high to low uncertainty [22, 14, 19, 91, 63]. Other approaches use diffusion models to generate multiple pose hypotheses from a single 2D observation [66, 27]. These 3D pose estimators effectively reduce uncertainty and indeterminacy throughout the estimation process. Moreover, discrete diffusion models have also gained attention in various domains [30, 40, 24, 33]. Inspired by these advancements, our work introduces a discrete diffusion model for occluded 3D HPE, which aligns more closely with the inherent discreteness of 3D pose data and effectively incorporates occlusion into the estimation process, providing a novel perspective in the field.

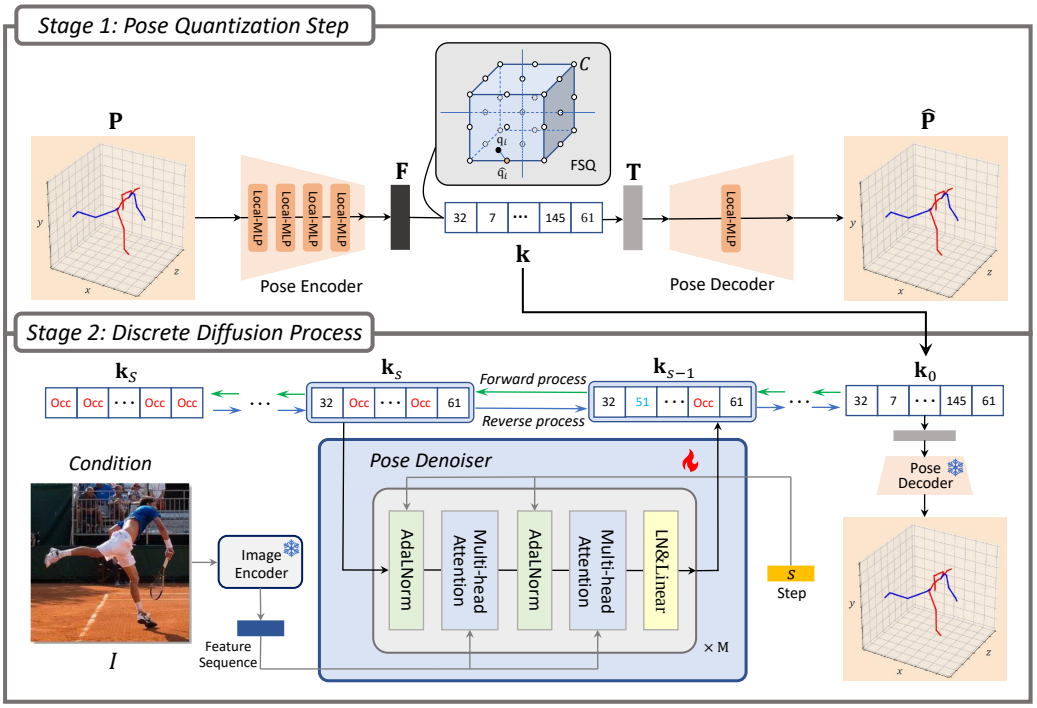


Figure 2: Overview of our two-stage Di^{2}Pose framework. In the stage 1, we train a pose quantization step that transforms a 3D pose $\mathbf{P}$ into multiple discrete tokens $\mathbf{k}$, each token representing the indices of implied codebook $\mathcal{C}$. In the stage 2, we model $\mathbf{k}$ in the discrete space by discrete diffusion process. In the forward process, each token is probabilistically occluded with Occ token or replaced with another available token. In the reverse process, the model leverages an independent image encoder and a pose denoiser to reconstruct all the tokens based on the condition 2D image. These reconstructed tokens are finally decoded by the pose decoder, resulting in the recovered 3D pose. Notably, we only update the parameters of pose denoiser, pose decoder and image encoder are frozen.

3 Di^{2}Pose

Given an 2D image $I \in \mathbb{R}^{H\times W\times 3}$, the goal of 3D HPE is to predict $\hat{\mathbf{P}} \in \mathbb{R}^{J\times 3}$, which represents the 3D coordinates of all the J joints of the human body. In this paper, we construct occluded 3D HPE task as a two-stage framework including the pose quantization step and the discrete diffusion process.

As shown in Figure 2, **in the training phase**, *Stage 1* learns a pose quantization step by a VQ-VAE like structure (Sec. 3.1), which is able to encode a 3D pose into multiple quantized tokens. *Stage 2* models quantized pose tokens in the latent space by the forward and reverse process of a conditional diffusion model (Sec. 3.2). **In the inference phase**, we only use *the reverse process* of Stage 2 and the pre-trained pose decoder of Stage 1 to recover 3D pose from the 2D image. Notably, pose tokens are either occluded or initialized randomly at the beginning of the inference phase. The model reconstructs all the tokens based on the condition 2D image step-by-step. These reconstructed tokens are finally decoded by the pre-trained pose decoder, resulting in the recovered 3D pose.

3.1 Pose Quantization Step

As depicted in Figure 2, a pose quantization step comprises a pose encoder, the quantization process, and a pose decoder. Initially, for a real 3D pose $\mathbf{P} \in \mathbb{R}^{J\times 3}$, the pose encoder $f_{PE}(\cdot)$ converts $\mathbf{P}$ to token features $\mathbf{F}$. During the quantization process, we utilize FSQ to quantize $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_N)$ $(\mathbf{f}_i \in \mathbb{R}^D)$ into tokens $\mathbf{T} = (\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_N)$ $(\mathbf{t}_i \in \mathbb{R}^D)$. Finally, the quantized tokens $\mathbf{T}$ are decoded by the pose decoder $f_{PD}(\cdot)$ to reconstruct 3D pose $\hat{\mathbf{P}}$.

Pose Encoder. Considering the interdependencies among human body joints, our goal is to represent 3D poses in a compositional manner, moving away from reliance on coordinates vectors or heatmap

embeddings. The VQ-VAE architecture, incorporating MLP-Mixer blocks [69] within its encoder and decoder, has been proven effective in decomposing a pose into multiple token features, each corresponding to a sub-structure of the pose [21]. However, the MLP-Mixer block is designed to extract global information across all joints, which can not adequately capture the local relationships between joints within individual sub-structure.

In response to aforementioned limitation, we design Local-MLP block to capture the local interactions between 3D joints. The pose encoder $f_{PE}(\cdot)$, comprising several Local-MLP blocks, converts $\mathbf{P}$ to N token features:

$$\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_N) = f_{PE}(f_{emb}(\mathbf{P})), \tag{1}$$

where $f_{emb}(\cdot)$ embeds $\mathbf{P}$ to $\mathbf{P_{emb}} \in \mathbb{R}^{J \times D}$ by a linear layer.

As shown in Figure 3(a), a Local-MLP block is composed of a Layer Normalization layer, a Joint Shift block (JS-Block), a Channel MLP, and a residual connection. The JS-Block is specifically designed to capture local interactions among X joints. It extracts features by linear projection, and the Joint Shift operation enables feature translation along joint connection directions. As shown in Figure 3(b), with the input $\mathbf{P}^{\top}_{\mathbf{emb}} \in \mathbb{R}^{D \times J}$, the feature is evenly divided into X segments ($X = 3$ in the example), each segment being shifted incrementally by units from $-\lfloor X/2 \rfloor$ to $\lfloor X/2 \rfloor$. The central segment remains stationary, while the segments to the left and right are symmetrically shifted away from the center by up to $\pm\lfloor X/2 \rfloor$ units. Zero padding is used to maintain dimensionality. Features highlighted within the dashed box are selected for further linear projection. Finally, the Channel MLP processes these features channel-wise to facilitate information integration.

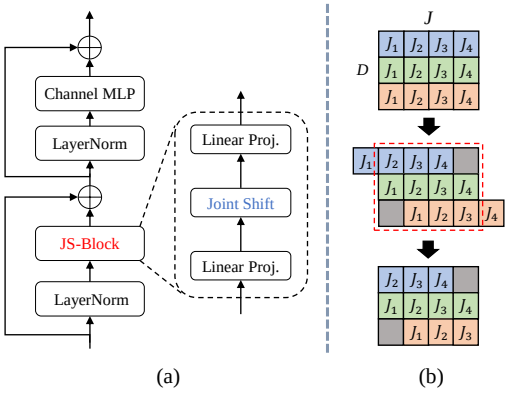


Figure 3: (a) depicts the structure of the Local-MLP block; (b) shows the Joint Shift operation, where the arrows indicate the steps, and different subscript numbers represent the features of different joints. The gray blocks indicate zero padding.

Quantization Process. During this process, we exploit FSQ [49] to enhance the utilization of codewords. FSQ quantizes token features $\mathbf{F}$ as corresponding token indices:

$$\mathbf{k} = (k_1, k_2, \cdots, k_N) = \text{FSQ}(f_{proj}(\mathbf{F})), \tag{2}$$

where $f_{proj}(\cdot)$ projects each $\mathbf{f}_i \in \mathbb{R}^D$ of $\mathbf{F}$ to $\mathbf{q}_i \in \mathbb{R}^d$, and each k_i of $\mathbf{k}$ denotes the entries of implied codebook $\mathcal{C}$. For each $\mathbf{q}_i$, FSQ employs a bounding function $f_{bnd} : \mathbf{q}_i \mapsto \lfloor L/2 \rfloor \cdot \tanh(\mathbf{q}_i)$ to constrain each channel of d. As a result, each channel in $\hat{\mathbf{q}}_i = \text{round}(f_{bnd}(\mathbf{q}_i))$ takes one of L unique values. This procedure yields $\hat{\mathbf{q}}_i \in \mathcal{C}$, where the total number of unique codebook entries is $|\mathcal{C}| = \prod_{i=1}^{d} L_i$ (mapping the i-th channel to L_i values). The vectors in $\mathcal{C}$ can be enumerated, establishing a bijective mapping from any $\hat{\mathbf{q}}_i$ to an integer within $\{1, \ldots, |\mathcal{C}|\}$. In addition, the corresponding codeword of k_i, which is denoted as $\mathbf{t}_i \in \mathbb{R}^D$, represents the quantized result of $\mathbf{f}_i$. Thereby, using FSQ, the token features $\mathbf{F}$ are quantized as $\mathbf{T} = (\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_N)$.

Pose Decoder. The pose decoder $f_{PD}(\cdot)$ is designed to recover 3D pose $\hat{\mathbf{P}}$ from $\mathbf{T}$. $f_{PD}(\cdot)$ adopts a structure similar to the pose encoder but in reverse, utilizing a reduced number of Local-MLP blocks.

Loss. The pose quantization step, including the pose encoder, quantization process, and pose decoder, is jointly optimized by minimizing L1 loss $\mathcal{L}_{PQ} = ||P - \hat{P}||_1$ across the training dataset.

3.2 Discrete Diffusion Process

After training the pose quantization step, we can acquire N quantized tokens $\mathbf{k}$ from the original 3D pose $\mathbf{P}$. The next step in Di^{2}Pose pipeline is to model $\mathbf{k}$ in the latent space by the discrete diffusion process. In the following, we first briefly introduce the diffusion models and clarify the basic principles of the discrete diffusion model. Then we explain the details of discrete diffusion

process, including the designed transition matrix and loss function. Eventually, we illustrate the architecture and training and inference process.

Discrete Diffusion Model. Our discrete diffusion model is characterized by two distinct processes: 1) **Forward process**: It progresses through discrete steps $s \in \{0, 1, 2, ..., S\}$, gradually transforming the initial tokens $\mathbf{k}_0$ (the quantized token $\mathbf{k}$) into a noise-infused latent representation $\mathbf{k}_S$. 2) **Reverse process**: It is tasked with reconstructing the original data $\mathbf{k}_0$ from the latent $\mathbf{k}_S$, following a reverse temporal sequence $s \in \{S, S-1, ..., 1, 0\}$.

Followed previous studies [67, 3, 28], we use a transition probability matrix $[\mathbf{M}_s]_{ij} = q(\mathbf{k}_s = i|\mathbf{k}_{s-1} = j) \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{C}|}$ elucidate the likelihood of transitioning from $\mathbf{k}_{s-1}$ to $\mathbf{k}_s$. Then the forward process for the entire sequence of tokens is expressed as:

$$q(\mathbf{k}_s|\mathbf{k}_{s-1}) = \boldsymbol{c}^\top(\mathbf{k}_s)\mathbf{M}_s\boldsymbol{c}(\mathbf{k}_{s-1}), \tag{3}$$

where $\boldsymbol{c}(\cdot)$ symbolizes a function capable of converting a scalar into a one-hot column vector. The distribution of $\mathbf{k}_s$ follows a categorical distribution, determined by the vector $\mathbf{M}_s\boldsymbol{c}(\mathbf{k}_{s-1})$. Leveraging the Markov chain property, it is feasible to bypass intermediate stages, directly computing the probability of $\mathbf{k}_s$ from $\mathbf{k}_0$ for any given step as:

$$q(\mathbf{k}_s|\mathbf{k}_0) = \boldsymbol{c}^\top(\mathbf{k}_s)\overline{\mathbf{M}}_s\boldsymbol{c}(\mathbf{k}_0), \text{with } \overline{\mathbf{M}}_s = \mathbf{M}_s \dots \mathbf{M}_1 \tag{4}$$

Moreover, the posterior of the reverse process, $q(\mathbf{k}_{s-1}|\mathbf{k}_s, \mathbf{k}_0)$, can be ascertained as:

$$q(\mathbf{k}_{s-1}|\mathbf{k}_s, \mathbf{k}_0) = \frac{q(\mathbf{k}_s|\mathbf{k}_{s-1}, \mathbf{k}_0)q(\mathbf{k}_{s-1}|\mathbf{k}_0)}{q(\mathbf{k}_s|\mathbf{k}_0)} = \frac{\left(\boldsymbol{c}^\top(\mathbf{k}_s)\mathbf{M}_s\boldsymbol{c}(\mathbf{k}_{s-1})\right)\left(\boldsymbol{c}^\top(\mathbf{k}_{s-1})\overline{\mathbf{M}}_{s-1}\boldsymbol{c}(\mathbf{k}_0)\right)}{\boldsymbol{c}^\top(\mathbf{k}_s)\overline{\mathbf{M}}_s\boldsymbol{c}(\mathbf{k}_0)}. \tag{5}$$

Occlude and Replace Transition Matrix. Notably, a suitable design for transition probability matrix $\mathbf{M}_s$ is significant to train the discrete diffusion process. As illustrated in Section 3.1, through pre-trained pose encoder and FSQ, $\mathbf{P}$ can be converted to $\mathbf{k} = (k_1, k_2, \cdots, k_N)$, each k_i corresponding to a sub-structure of the overall pose. With this foundation, we specifically devise the `occlude` and `replace` scheme, which is inspired by [24], for tackling the challenges of occluded 3D HPE. In occlusion scenes, the human body is always occluded in various situations (self-occlusions, object or people-to-person occlusions), and the typical manifestation is that some sub-structures of the pose are invisible. Consequently, we design the `occlude` scheme simulating the occlusion of corresponding joints, which introduces occlusion impact in the training process. Additionally, recognizing the inherent uncertainty in occlusion scenarios where a single occluded region may correspond to multiple potential 3D human poses, we develop the `replace` strategy to update certain token with another available token.

In practice, each quantized token k_i has a probability of γ_s to transition to the `Occ` token. Moreover, k_i is also subject to a probability of $|\mathcal{C}|\beta_s$ to be uniformly resampled across all $|\mathcal{C}|$ categories. Furthermore, k_i retains a probability of $\alpha_s = 1 - |\mathcal{C}|\beta_s - \gamma_s$ to remain unchanged. Then, the transition matrix $\mathbf{M}_s \in \mathbb{R}^{(|\mathcal{C}|+1)\times(|\mathcal{C}|+1)}$ is defined as:

$$\mathbf{M}_s = \begin{bmatrix} \alpha_s + \beta_s & \beta_s & \cdots & 0 \\ \beta_s & \alpha_s + \beta_s & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_s & \gamma_s & \cdots & 1 \end{bmatrix}, \tag{6}$$

where $\alpha_s, \beta_s \in [0, 1]$. The prior distribution of step S can be derived as: $p(\mathbf{k}_S) = [\overline{\beta}_S, \overline{\beta}_S, \cdots, \overline{\gamma}_S]$, where $\overline{\alpha}_S = \prod_{i=1}^{S} \alpha_i$, $\overline{\gamma}_S = 1 - \prod_{i=1}^{S}(1 - \gamma_i)$ and $\overline{\beta}_S = (1 - \overline{\alpha}_S - \overline{\gamma}_S)/|\mathcal{C}|$. In this study, we adapt the linear schedule [25] as noise schedule to pre-define the value of transition matrices ($\overline{\alpha}_S$, $\overline{\beta}_S$, and $\overline{\gamma}_S$). Subsequently, we can calculate $q(\mathbf{k}_s|\mathbf{k}_0)$ according to Eq. (4). However, when the number of categories $|\mathcal{C}|$ and time step S is too large, it can quickly become impractical to store all of the transition matrices $\mathbf{M}_s$ in memory, as the memory usage grows like $O(|\mathcal{C}|^2 S)$. Actually, it is unnecessary to store all of the transition matrices. Instead we only store all of $\overline{\alpha}_s$ and $\overline{\beta}_s$ in advance, since we can calculate $q(\mathbf{k}_s|\mathbf{k}_0)$ according to following formula (refer to Appendix for proofs):

$$\overline{\mathbf{M}}_s\boldsymbol{c}(\mathbf{k}_0) = \overline{\alpha}_s\boldsymbol{c}(\mathbf{k}_0) + (\overline{\gamma}_s - \overline{\beta}_s)\boldsymbol{c}(|\mathcal{C}| + 1) + \overline{\beta}_s. \tag{7}$$

Training Objectives. We train a network $f_\theta(\mathbf{k}_{s-1}|\mathbf{k}_s, \boldsymbol{y})$ to estimate $q(\mathbf{k}_{s-1}|\mathbf{k}_s, \mathbf{k}_0)$ in the reverse process. The network is trained to minimize the variational lower bound (VLB):

$$\mathcal{L}_{vlb} = D_{KL}(q(\mathbf{k}_S|\mathbf{k}_0)||p(\mathbf{k}_S)) + \textstyle\sum_{s=1}^{S-1}\{D_{KL}[q(\mathbf{k}_{s-1}|\mathbf{k}_s, \mathbf{k}_0)||f_\theta(\mathbf{k}_{s-1}|\mathbf{k}_s, \boldsymbol{y})]\}, \tag{8}$$

In addition, we follow [50, 24] to utilize the reparameterization trick, which lets Di^{2}Pose predict the noiseless token distribution $f_\theta(\hat{\mathbf{k}}_0|\mathbf{k}_s, \boldsymbol{y})$ at each reverse step, and then compute $f_\theta(\mathbf{k}_{s-1}|\mathbf{k}_s, \boldsymbol{y})$ as:

$$f_\theta(\mathbf{k}_{s-1}|\mathbf{k}_s, \boldsymbol{y}) = \sum_{\hat{\mathbf{k}}_0=1}^{H} q(\mathbf{k}_{s-1}|\mathbf{k}_s, \hat{\mathbf{k}}_0) f_\theta(\hat{\mathbf{k}}_0|\mathbf{k}_s, \boldsymbol{y}). \tag{9}$$

Based on the Eq. (9), an auxiliary denoising objective loss is introduced, which encourages the network to predict $f_\theta(\hat{\mathbf{k}}_0|\mathbf{k}_s, \boldsymbol{y})$:

$$\mathcal{L}_{k_0} = -\log f_\theta(\hat{\mathbf{k}}_0|\mathbf{k}_s, \boldsymbol{y}). \tag{10}$$

Our final loss function is defined as:

$$\mathcal{L} = \lambda \mathcal{L}_{\mathbf{k}_0} + \mathcal{L}_{vlb}, \tag{11}$$

where λ is a hyper-parameter to control the weight of the auxiliary loss $\mathcal{L}_{\mathbf{k}_0}$.

Diffusion Architecture. As depicted in Figure 2, our discrete diffusion model consists of three main components: an image encoder, a pose denoiser, and a pose decoder. The pre-trained image encoder processes the 2D image to produce a conditional feature sequence. The pose denoiser, receiving the quantized pose tokens $\mathbf{k}_s$ and and step S, predicts the distribution of noiseless tokens $f_\theta(\hat{\mathbf{k}}_0|\mathbf{k}_s, \boldsymbol{y})$. This component is equipped with several transformer blocks, each featuring an AdaLNorm operator [6], multi-head attention blocks that combine the image feature information with $\mathbf{k}_s$, and layer normalization and linear layers. At the end of the reverse process, all recovered tokens are obtained, and the final prediction of 3D pose is decoded by the well-trained pose decoder.

Training and Inference Process. *In the training process*, as for step s, we sample $\mathbf{k}_s$ from $q(\mathbf{k}_s|\mathbf{k}_0)$ based on Eq. (7) in the forward process. We then estimate $f_\theta(\mathbf{k}_{s-1}|\mathbf{k}_s, \boldsymbol{y})$ in the reverse process. The final loss will be calculated according to Eq. (11). *In the inference process*, all pose tokens are either masked or initialized randomly. Subsequently, we predict $f_\theta(\mathbf{k}_{s-1}|\mathbf{k}_s, \boldsymbol{y})$ step by step until the tokens are completely recovered. Finally, reconstructed tokens are decoded by the pose decoder, resulting in the recovered 3D pose. The complete algorithms are summarized in Appendix.

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets. Human3.6M [34] is the most extensive benchmark for 3D HPE, consisting of 3.6 million images. We follow [22] with same protocol, which involves training on subjects S1, S5, S6, S7, and S8, and testing on subjects S9 and S11. **3DPW** [72] is the first dataset in the wild that includes video footage taken from a moving phone camera. We also evaluate our method on this dataset to measure the robustness and generalization. Additionally, to further verify the occlusion-robustness, we evaluate Di^{2}Pose on the **3DPW-Occ** [83], which is a subset of the 3DPW.

Evaluation Metrics. For Human3.6M and 3DPW, we follow the standard protocols. Mean per joint position error (**MPJPE**) calculates the mean Euclidean distance between the root-aligned reconstructed poses and ground truth joint coordinates. **PA-MPJPE** employs a Procrustes alignment between the poses before calculating the MPJPE. In addition, to further evaluate the effectiveness of our method under occlusion scenes, we devise an adversarial protocol, termed **3DPW-AdvOcc**, following the previous research [84]. We apply occlusion patches to the input image to identify the most challenging predictions. This process involves assessing the relative performance degradation on the visible joints. Similar to [84], we utilize textured patches generated by randomly cropping texture maps from the DTD [16]. We employ two square patch sizes: 40 and 80 relative to a 256 × 192 image, denoted as Occ@40 and Occ@80 respectively, with a stride of 10.

4.2 Implementation Details

Pose Quantization Step. The pose encoder is constructed with four Local-MLP blocks, while the pose decoder incorporates a single block. Within these Local-MLP blocks, the embedding dimensions D for the pose encoder and decoder are configured to 2048 and 512, respectively. For the quantization process, the projected vector $\mathbf{q}_i$ features the channel $d = 5$. The levels per channel, denoted as $[L_1, \cdots, L_d]$, are specified as $[7, 5, 5, 5, 5]$. The number of quantized tokens N is set to 100.

Table 1: Results on Human3.6M in millimeters under MPJPE. The best results are in **bold**, and the second-best ones are underlined.

Methods	Dir	Disc	Eat	Gr.	Phon.	Phot.	Pose	Pur.	Sit	SitD.	Sm.	Wait	W.D.	Walk	W.T.	Avg
Pavlakos *et al.* [54] *CVPR'17*	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
Martinez *et al.* [48] *ICCV'17*	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Hossain *et al.* [29] *ECCV'18*	48.4	50.7	57.2	55.2	63.1	72.6	53.0	51.7	66.1	80.9	59.0	57.3	62.4	46.6	49.6	58.3
Zhao *et al.* [85] *CVPR'19*	48.2	60.8	51.8	64.0	64.6	**53.6**	51.1	67.4	88.7	**57.7**	73.2	65.6	48.9	64.8	51.9	60.8
Liu *et al.* [45] *ECCV'18*	46.3	52.2	47.3	50.7	55.5	67.1	49.2	46.0	60.4	71.1	51.5	50.1	54.5	40.3	43.7	52.4
Xu *et al.* [77] *CVPR'21*	45.2	49.9	47.5	50.9	54.9	66.1	48.5	46.3	59.7	71.5	51.4	48.6	53.9	39.9	44.1	51.9
Zhao *et al.* [88] *CVPR'22*	45.2	50.8	48.0	50.0	54.9	65.0	48.2	47.1	60.2	70.0	51.6	48.7	54.1	39.7	43.1	51.8
Geng *et al.* [21] *CVPR'23*	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	50.8
Choi *et al.* [14] *IROS'23*	44.3	51.6	46.3	51.1	**50.3**	54.3	49.4	45.9	57.7	71.6	**48.6**	49.1	52.1	44.0	44.4	50.7
Zhang *et al.* [82] *TPAMI'23*	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	50.2
Gong *et al.* [22] *CVPR'23*	42.8	49.1	45.2	**48.7**	52.1	63.5	46.3	**45.2**	58.6	66.3	50.4	47.6	52.0	37.6	**40.2**	49.7
Di^{2}Pose (Ours)	**41.9**	**47.8**	**45.0**	49.0	51.5	62.2	**45.7**	45.6	**57.6**	67.1	50.1	**45.3**	**51.4**	**37.3**	40.9	**49.2**

Table 2: Evaluation on 3DPW, 3DPW-Occ, and 3DPW-AdvOcc. The number 40 and 80 after 3DPW-AdvOcc denote the occluder size. * denotes the results from our implementation. The best results are in **bold**, and the second-best ones are underlined.

Methods	3DPW [72]		3DPW-Occ [83]		3DPW-AdvOcc@40		3DPW-AdvOcc@80	
	MPJPE ↓	PA-MPJPE ↓	MPJPE ↓	PA-MPJPE ↓	MPJPE ↓	PA-MPJPE ↓	MPJPE ↓	PA-MPJPE ↓
Cai *et al.* [9] *ICCV'19*	112.9	69.6	115.8	72.3	241.1	101.4	355.9	116.3
Pavllo *et al.* [55] *CVPR'19*	101.8	63.0	106.7	67.1	221.6	99.4	334.3	112.9
Cheng *et al.* [12] *AAAI'21*	—	64.2	—	85.7	279.4	113.2	371.4	119.8
Zheng *et al.* [90] *ICCV'21*	118.2	73.1	132.8	80.5	247.9	106.2	359.6	115.5
Zhang *et al.* [82] *TPAMI'23*	91.1	54.3	94.6	56.7	142.5	73.8	251.8	103.9
Geng *et al.* * [21] *CVPR'23*	83.1	53.9	82.8	53.7	127.2	71.9	192.5	92.1
Gong *et al.* * [22] *CVPR'23*	82.7	53.8	82.1	53.5	121.4	70.9	189.3	92.4
Di^{2}Pose (Ours)	**79.3**	**50.1**	**79.6**	**50.7**	**108.4**	**59.8**	**153.6**	**78.7**

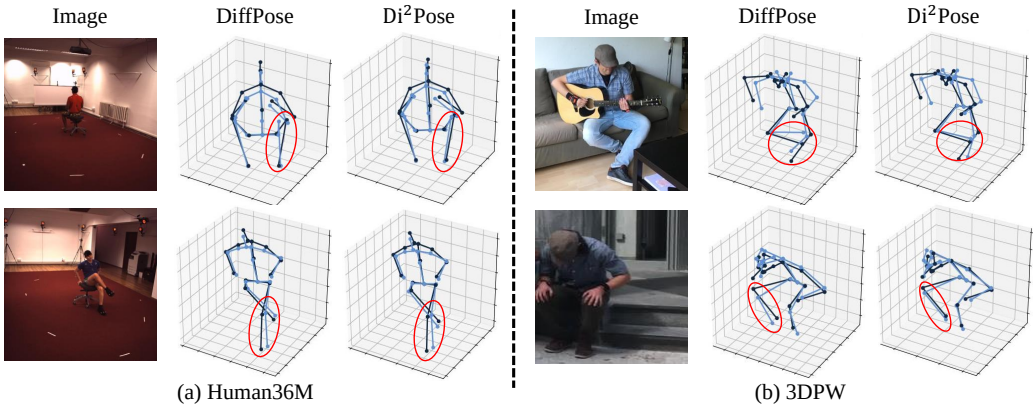
Discrete Diffusion Process. For the occlude and replace transition matrix, we linearly increase $\overline{\beta}_s$ and $\overline{\gamma}_s$ from 0 to 0.1 and 0.9, respectively, and decrease $\overline{\alpha}_s$ from 1 to 0. For the discrete diffusion model, we use off-the-shelf image encoder [79] to extract feature sequence of conditional 2D image. As for the pose denoiser, we build a 21-layer 16-head transformer with the dimension of 1024. We set steps S as 100 and loss weight λ is set to 5e-4. Please refer to Appendix for more details.

4.3 Comparsion with State-of-the-Arts

Human3.6M. To explore the effectiveness of Di^{2}Pose, we evaluate its performance in the challenging context of frame-based 3D pose estimation. Specifically, within the discrete diffusion process, context information is extracted from a single input frame using the image encoder. As shown in Table 1, we benchmark Di^{2}Pose against SOTA 3D HPE methods on the Human3.6M. Our Di^{2}Pose achieves 49.2mm in average MPJPE, surpassing the performance of the SOTA diffusion model [22] by 0.5mm, which indicates that Di^{2}Pose is able to enhance monocular 3D HPE in indoor scenes.

3DPW. Beyond indoor settings, we evaluate the performance of Di^{2}Pose on the in-the-wild 3DPW dataset. As Table 2 shows, Di^{2}Pose achieves the SOTA performance, and outperforms the SOTA method [22] by 3.4mm in MPJPE and 3.7mm in PA-MPJPE. On the occlusion-centric **3DPW-Occ**, Di^{2}Pose maintains its superiority. When assessed under the 3DPW-AdvOcc protocol, all methods exhibit performance drops—MPJPE surges by up to 129% and PA-MPJPE by up to 72%. Despite this, Di^{2}Pose remains markedly robust, leading the SOTA by significant margins in both MPJPE and PA-MPJPE, underscoring its effectiveness in handling occlusions.

Qualitative Results. Figure 4 presents the qualitative results of DiffPose [22] in comparison with our Di^{2}Pose across two datasets. It can be observed that our method yields more accurate predictions than compared diffusion model (DiffPose), especially under various occlusion scenarios (self-occlusion and object occlusion). This demonstrates the superior occlusion-robustness of our Di^{2}Pose.



(a) Human36M (b) 3DPW

Figure 4: Qualitative results on two datasets. The black lines represent the ground truth poses and the blue lines are prediction results.

Table 3: Ablations on Human3.6M. P-1 and P-2 represent MPJPE and PA-MPJPE, respectively.

Loc. Num.	P-1	P-2
1	14.5	13.0
3	**13.6**	**12.5**
5	14.1	12.8

(a) Different local joint number X of Joint Shift operations in JS-Block.

Levels	P-1	P-2
$[8,5,5,5]$	15.2	15.9
$[7,5,5,5,5]$	**13.6**	**12.5**
$[8,8,8,6,5]$	13.8	12.7

(b) Different levels per channel $[L_1, \cdots, L_d]$ of quantization process FSQ.

Occ. Rate	P-1	P-2
0	50.4	39.3
0.3	50.7	39.3
0.6	49.5	39.1
0.9	**49.2**	**39.0**
1.0	51.0	39.5

(c) Different final occlude rate $\overline{\gamma}_S$ for the occlude and replace transition matrix.

Inference Steps \ Training Steps	25	50	100
25	51.7	51.1	50.3
50	—	50.6	49.9
100	—	—	**49.2**

(d) Different number of training and inference steps S. P-1 are reported.

4.4 Ablation Study

Effectiveness of Pose Quantization Step. Our pose quantization step, which consists of Local-MLP blocks, is designed for representing 3D human pose by capturing the local interactions between 3D joints. Table 4 displays the MPJPE metrics comparing the original 3D poses with those reconstructed via various methods. The results show that our pose quantization step reconstructs 3D poses with lower errors compared to previous method [21], which uses an MLP-Mixer for global joint information extraction. It indicates that our model learns a more accurate representation of 3D poses. In addition, we conducted other ablation studies to investigate different local joint numbers X and levels per channel $[L_1, \cdots, L_d]$ within pose quantization step, as shown in Table 3a and Table 3b. As to different X, note that when $X = 1$, we only extract feature of individual joint, and when $X > 1$, JS-Block is able to capture local interactions of different joints. Experimental results indicate that $X = 3$ reaches lowest reconstruct error. As for $[L_1, \cdots, L_d]$, the best level of FSQ for pose quantization is $[7, 5, 5, 5, 5]$.

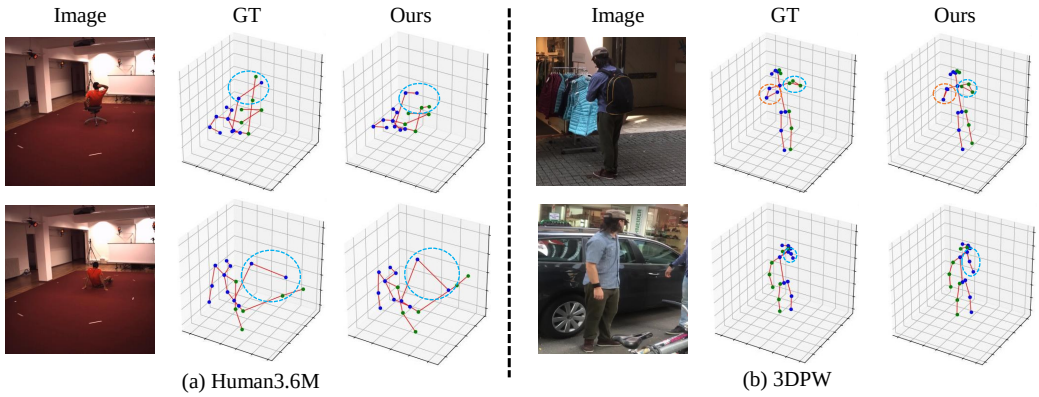
Table 4: Different representation methods for 3D HPE.

Pose Repr.	MPJPE	PA-MPJPE
PCT [21]	15.2	15.9
Ours	**13.6**	**12.5**

Impact of Different Transition Matrices. To demonstrate the effectiveness of the specifically designed occlude and replace transition matrix, we constructed three transition matrices for discrete diffusion process: occlude transition matrix, replace transition matrix, and occlude and replace transition matrix. Table 5 illustrates that the optimal results are achieved when utilizing the occlude and replace transition matrix. The suboptimal performance observed when exclusively employing the other two transition matrices can be attributed to the following reasons: Utilizing solely the replace transition matrix introduces the challenge of random, irrelevant sub-structures, complicating the learning of the reverse process; Conversely, relying exclusively on the occlude

Table 5: Different transition matrices for discrete diffusion model.

Matrices	MPJPE	PA-MPJPE
Occlude	51.0	39.5
Replace	50.4	39.3
Both	**49.2**	**39.0**



(a) Human3.6M (b) 3DPW

Figure 5: Failure cases of our Di^{2}Pose for 3D HPE. These instances primarily occur in scenarios with severe occlusions, as compared against ground truth (GT) poses. The content encircled by the dashed line indicates the parts where differences exist.

transition matrix causes the model to overly focus on the occluded portions, neglecting the contextual information from other visible parts. These clarifications can be verified in Table 3c, where we investigate the impact of different $\overline{\gamma}_S$. When $\overline{\gamma}_S = 0$, the occlude and replace transition matrix can be seen as the replace transition matrix, and when $\overline{\gamma}_S = 1$, the occlude and replace transition matrix can be seen as the occlude transition matrix. The best performance is obtained when $\overline{\gamma}_S = 0.9$.

In addition, we conducted an ablation study to investigate the impact of S on the training and inference processes, as shown in Table 3d. We observed that using larger numbers of steps during both training and inference stages improves performance but also increases time complexity. Moreover, the results indicate that performance remains satisfactory even when the number of inference steps is reduced by 75% (e.g., from 100 steps during training to 25 steps during inference). This finding suggests a viable strategy for enhancing generation speed without significantly compromising quality.

5 Limitations

Figure 5 illustrates several results of 3D human pose estimation. When substantial occlusions cover the human body—obscuring the exact pose to the extent that it confounds even human observers—the predictions made by Di^{2}Pose may deviate from GT 3D pose. This deviation primarily stems from the inherent limitation of inferring 3D poses directly from 2D images, which lack critical spatial depth information. Such limitations introduce uncertainty and indeterminacy in the predictions.

Despite these challenges, Di^{2}Pose manages occlusions effectively by producing physically plausible outcomes. This capability is attributed to the integration of a pose quantization step within Di^{2}Pose, which constrains the model's search space to physically reasonable configurations. Note that the pose quantization step is trained on real 3D human pose data, enhancing its reliability under severe occlusions.

Currently, Di^{2}Pose is primarily designed for frame-based 3D HPE and does not utilize interframe data from videos. Future enhancements will focus on incorporating interframe information to refine the accuracy of 3D pose predictions further within the Di^{2}Pose framework.

6 Conclusion

This paper presents Di^{2}Pose, a novel diffusion-based framework that tackles occluded 3D HPE in discrete space. Di^{2}Pose first captures the local interactions of joints and represents a 3D pose by multiple quantized tokens. Then, the discrete diffusion process models the discrete tokens in latent space through a conditional diffusion model, which implicitly introduces occlusion into the modeling process for more reliable 3D HPE with occlusions. Experimental results show that our method surpasses the state-of-the-art approaches on three widely used benchmarks.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62337001) and the Fundamental Research Funds for the Central Universities (226-2024-00058). Long Chen is supported by HKUST Special Support for Young Faculty (F0927) and HKUST Sports Science and Technology Research Grant (SSTRG24EG04).

References

[1] I. Akhter and M. J. Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1446–1455, 2015.

[2] T. Alldieck, H. Xu, and C. Sminchisescu. imghum: Implicit generative models of 3d human shape and articulated pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5461–5470, 2021.

[3] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

[4] O. Avrahami, D. Lischinski, and O. Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022.

[5] S. Azadi, A. Shah, T. Hayes, D. Parikh, and S. Gupta. Make-an-animation: Large-scale text-conditional 3d human motion generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15039–15048, 2023.

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

[7] B. Biggs, D. Novotny, S. Ehrhardt, H. Joo, B. Graham, and A. Vedaldi. 3d multi-bodies: Fitting sets of plausible 3d human models to ambiguous image data. *Advances in neural information processing systems*, 33:20496–20507, 2020.

[8] E. A. Brempong, S. Kornblith, T. Chen, N. Parmar, M. Minderer, and M. Norouzi. Denoising pretraining for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4175–4186, 2022.

[9] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2272–2281, 2019.

[10] S. Chen, P. Sun, Y. Song, and P. Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19830–19843, 2023.

[11] T. Chen, C. Fang, X. Shen, Y. Zhu, Z. Chen, and J. Luo. Anatomy-aware 3d human pose estimation with bone-based pose decomposition. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(1):198–209, 2021.

[12] Y. Cheng, B. Wang, B. Yang, and R. T. Tan. Graph and temporal convolutional networks for 3d multi-person pose estimation in monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1157–1165, 2021.

[13] H.-g. Chi, S. Chi, S. Chan, and K. Ramani. Pose relation transformer refine occlusions for human pose estimation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6138–6145. IEEE, 2023.

[14] J. Choi, D. Shim, and H. J. Kim. Diffupose: Monocular 3d human pose estimation via denoising diffusion probabilistic model. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3773–3780. IEEE, 2023.

[15] H. Ci, C. Wang, X. Ma, and Y. Wang. Optimizing network structure for 3d human pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2262–2271, 2019.

[16] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[17] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[18] Z. Fan, J. Liu, and Y. Wang. Motion adaptive pose estimation from compressed videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11719–11728, 2021.

[19] R. Feng, Y. Gao, T. H. E. Tse, X. Ma, and H. J. Chang. Diffpose: Spatiotemporal diffusion model for video-based human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14861–14872, 2023.

[20] L. G. Foo, J. Gong, Z. Fan, and J. Liu. System-status-aware adaptive network for online streaming video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10514–10523, 2023.

[21] Z. Geng, C. Wang, Y. Wei, Z. Liu, H. Li, and H. Hu. Human pose as compositional tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 660–671, 2023.

[22] J. Gong, L. G. Foo, Z. Fan, Q. Ke, H. Rahmani, and J. Liu. Diffpose: Toward more reliable 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13041–13051, 2023.

[23] S. Gong, M. Li, J. Feng, Z. Wu, and L. Kong. Diffuseq-v2: Bridging discrete and continuous text spaces for accelerated seq2seq diffusion models. *arXiv preprint arXiv:2310.05793*, 2023.

[24] 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.

[25] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[26] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

[27] K. Holmquist and B. Wandt. Diffpose: Multi-hypothesis human pose estimation using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15977–15987, 2023.

[28] E. Hoogeboom, D. Nielsen, P. Jaini, P. Forré, and M. Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.

[29] M. R. I. Hossain and J. J. Little. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 68–84, 2018.

[30] M. Hu, Y. Wang, T.-J. Cham, J. Yang, and P. N. Suganthan. Global context with discrete diffusion in vector quantised modelling for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11502–11511, 2022.

[31] W. Hu, C. Zhang, F. Zhan, L. Zhang, and T.-T. Wong. Conditional directed graph convolution for 3d human pose estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 602–611, 2021.

[32] R. Huang, Z. Zhao, H. Liu, J. Liu, C. Cui, and Y. Ren. Prodiff: Progressive fast diffusion model for high-quality text-to-speech. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2595–2605, 2022.

[33] N. Inoue, K. Kikuchi, E. Simo-Serra, M. Otani, and K. Yamaguchi. Layoutdm: Discrete diffusion model for controllable layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10167–10176, 2023.

[34] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

[35] W. Jiang, S. Jin, W. Liu, C. Qian, P. Luo, and S. Liu. Posetrans: A simple yet effective pose transformation augmentation for human pose estimation. In *European Conference on Computer Vision*, pages 643–659. Springer, 2022.

[36] Z. Jiang, Z. Wang, and L. Chen. Combing text-based and drag-based editing for precise and flexible image editing. *arXiv preprint arXiv:2410.03097*, 2024.

[37] Z. Jiang, Z. Zhou, L. Li, W. Chai, C.-Y. Yang, and J.-N. Hwang. Back to optimization: Diffusion-based zero-shot 3d human pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6142–6152, 2024.

[38] H. Joo, N. Neverova, and A. Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In *2021 International Conference on 3D Vision (3DV)*, pages 42–52. IEEE, 2021.

[39] M. Kocabas, C.-H. P. Huang, O. Hilliges, and M. J. Black. Pare: Part attention regressor for 3d human body estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11127–11137, 2021.

[40] H. Kong, K. Gong, D. Lian, M. B. Mi, and X. Wang. Priority-centric human motion generation in discrete latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14806–14816, 2023.

[41] J. N. Kundu, S. Seth, M. Rahul, M. Rakesh, V. B. Radhakrishnan, and A. Chakraborty. Kinematic-structure-preserved representation for unsupervised 3d human pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11312–11319, 2020.

[42] D. Lee, C. Kim, S. Kim, M. Cho, and W.-S. Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022.

[43] J. Li, S. Bian, A. Zeng, C. Wang, B. Pang, W. Liu, and C. Lu. Human pose regression with residual log-likelihood estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11025–11034, October 2021.

[44] J. Li, C. Xu, Z. Chen, S. Bian, L. Yang, and C. Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3383–3393, 2021.

[45] K. Liu, R. Ding, Z. Zou, L. Wang, and W. Tang. A comprehensive study of weight sharing in graph networks for 3d human pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 318–334. Springer, 2020.

[46] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[47] W. Mao, Y. Ge, C. Shen, Z. Tian, X. Wang, Z. Wang, and A. v. den Hengel. Poseur: Direct human pose regression with transformers. In *Proceedings of the European conference on computer vision (ECCV)*, pages 72–88. Springer, 2022.

[48] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2640–2649, 2017.

[49] F. Mentzer, D. Minnen, E. Agustsson, and M. Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023.

[50] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[51] Q. Nie, Z. Liu, and Y. Liu. Lifting 2d human pose to 3d with domain adapted 3d body concept. *International Journal of Computer Vision*, 131(5):1250–1268, 2023.

[52] S. Park, J. Hwang, and N. Kwak. 3d human pose estimation using convolutional neural networks with 2d pose information. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 156–169. Springer, 2016.

[53] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[54] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7025–7034, 2017.

[55] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7753–7762, 2019.

[56] J. Peng, D. Liu, S. Xu, and H. Li. Generating diverse structure for image inpainting with hierarchical vq-vae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10775–10784, 2021.

[57] I. A. Petrov, R. Marin, J. Chibane, and G. Pons-Moll. Object pop-up: Can we infer 3d objects and their poses from human interactions alone? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4726–4736, 2023.

[58] I. Radwan, A. Dhall, and R. Goecke. Monocular image 3d human pose estimation under self-occlusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1888–1895, 2013.

[59] A. Razavi, A. Van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[60] C. Rockwell and D. F. Fouhey. Full-body awareness from partial observations. In *European Conference on Computer Vision*, pages 522–539. Springer, 2020.

[61] G. Rogez and C. Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. *Advances in neural information processing systems*, 29, 2016.

[62] G. Rogez, P. Weinzaepfel, and C. Schmid. Lcr-net: Localization-classification-regression for human pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3433–3441, 2017.

[63] C. Rommel, E. Valle, M. Chen, S. Khalfaoui, R. Marlet, M. Cord, and P. Pérez. Diffhpe: Robust, coherent 3d human pose lifting with diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3220–3229, 2023.

[64] I. Sárándi, T. Linder, K. O. Arras, and B. Leibe. How robust is 3d human pose estimation to occlusion? *arXiv preprint arXiv:1808.09316*, 2018.

[65] W. Shan, Z. Liu, X. Zhang, S. Wang, S. Ma, and W. Gao. P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation. In *European Conference on Computer Vision*, pages 461–478. Springer, 2022.

[66] W. Shan, Z. Liu, X. Zhang, Z. Wang, K. Han, S. Wang, S. Ma, and W. Gao. Diffusion-based 3d human pose estimation with multi-hypothesis aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14761–14771, 2023.

[67] 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.

[68] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei. Integral human pose regression. In *Proceedings of the European conference on computer vision (ECCV)*, pages 529–545, 2018.

[69] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

[70] S. Tripathi, L. Müller, C.-H. P. Huang, O. Taheri, M. J. Black, and D. Tzionas. 3d human pose estimation via intuitive physics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 4713–4725, 2023.

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

[72] T. Von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European conference on computer vision (ECCV)*, pages 601–617, 2018.

[73] J. Wang, S. Yan, Y. Xiong, and D. Lin. Motion guided 3d pose estimation from videos. In *European conference on computer vision*, pages 764–780. Springer, 2020.

[74] Z. Wang, Y. Jiang, D. Zheng, J. Xiao, and L. Chen. Event-customized image generation. *arXiv preprint arXiv:2410.02483*, 2024.

[75] E. Xie, L. Yao, H. Shi, Z. Liu, D. Zhou, Z. Liu, J. Li, and Z. Li. Difffit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4230–4239, 2023.

[76] J. Xu, Z. Yu, B. Ni, J. Yang, X. Yang, and W. Zhang. Deep kinematics analysis for monocular 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on computer vision and Pattern recognition*, pages 899–908, 2020.

[77] T. Xu and W. Takano. Graph stacked hourglass networks for 3d human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16105–16114, 2021.

[78] Y. Xu, Z. Wang, J. Xiao, W. Liu, and L. Chen. Freetuner: Any subject in any style with training-free diffusion. *arXiv preprint arXiv:2405.14201*, 2024.

[79] Y. Xu, J. Zhang, Q. Zhang, and D. Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in Neural Information Processing Systems*, 35:38571–38584, 2022.

[80] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[81] A. Zanfir, M. Zanfir, A. Gorban, J. Ji, Y. Zhou, D. Anguelov, and C. Sminchisescu. Hum3dil: Semi-supervised multi-modal 3d humanpose estimation for autonomous driving. In *Conference on Robot Learning*, pages 1114–1124. PMLR, 2023.

[82] J. Zhang, K. Gong, X. Wang, and J. Feng. Learning to augment poses for 3d human pose estimation in images and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[83] T. Zhang, B. Huang, and Y. Wang. Object-occluded human shape and pose estimation from a single color image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7376–7385, 2020.

[84] Y. Zhang, P. Ji, A. Wang, J. Mei, A. Kortylewski, and A. Yuille. 3d-aware neural body fitting for occlusion robust 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9399–9410, 2023.

[85] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3425–3435, 2019.

[86] Q. Zhao, C. Zheng, M. Liu, and C. Chen. A single 2d pose with context is worth hundreds for 3d human pose estimation. *Advances in Neural Information Processing Systems*, 36, 2024.

[87] Q. Zhao, C. Zheng, M. Liu, P. Wang, and C. Chen. Poseformerv2: Exploring frequency domain for efficient and robust 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8877–8886, 2023.

[88] W. Zhao, W. Wang, and Y. Tian. Graformer: Graph-oriented transformer for 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20438–20447, 2022.

[89] C. Zheng and A. Vedaldi. Online clustered codebook. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22798–22807, 2023.

[90] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11656–11665, 2021.

[91] J. Zhou, T. Zhang, Z. Hayder, L. Petersson, and M. Harandi. Diff3dhpe: A diffusion model for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2092–2102, 2023.

Appendix

In this Appendix, we provide relevant preliminary knowledge, mathematical proofs, complete training and inference algorithms, additional experimental results, more implementation details about our Di^{2}Pose and broader impacts.

A Preliminary: Continuous Diffusion Model

The continuous diffusion model consists of two primary processes: the *forward process* and the *reverse process*. The forward process methodically corrupts the original data $\boldsymbol{x}_0$ into a noisy latent variable $\boldsymbol{x}_S$, which converges to a stationary distribution (e.g., a Gaussian distribution). Conversely, the reverse process aims to reconstruct the original data $\boldsymbol{x}_0$ from $\boldsymbol{x}_S$, utilizing learned parameters.

Forward Process Starting with $\boldsymbol{x}_0$ drawn from the distribution $q(\boldsymbol{x}_0)$, the forward process incrementally corrupts $\boldsymbol{x}_0$ through a sequence of latent variables $\boldsymbol{x}_{1:S} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_S)$, where each $\boldsymbol{x}_s$ retains the same dimensionality as $\boldsymbol{x}_0$. This transformation is modeled as a fixed Markov chain:

$$q(\boldsymbol{x}_{1:S}|\boldsymbol{x}_0) = \prod_{s=1}^{S} q(\boldsymbol{x}_s|\boldsymbol{x}_{s-1}). \tag{12}$$

where each transition $q(\boldsymbol{x}_s|\boldsymbol{x}_{s-1})$ is defined by a Gaussian distribution:

$$q(\boldsymbol{x}_s|\boldsymbol{x}_{s-1}) = \mathcal{N}(\boldsymbol{x}_s; \sqrt{1-\eta_s}\boldsymbol{x}_{s-1}, \eta_s \boldsymbol{I}) \tag{13}$$

Here, η_s is a small positive constant that follows a predefined schedule $(\eta_1, \eta_2, \ldots, \eta_S)$, allowing the data to progressively approach an isotropic Gaussian distribution, $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, as s increases. The overall transition from $\boldsymbol{x}_0$ to $\boldsymbol{x}_s$ can thus be expressed as:

$$q(\boldsymbol{x}_s|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_s; \sqrt{\overline{\zeta}_s}\boldsymbol{x}_0, (1-\overline{\zeta}_s)\boldsymbol{I}) \tag{14}$$

where $\zeta_s = 1 - \eta_s$ and $\overline{\zeta}_s = \prod_{i=1}^{s} \zeta_i$.

Reverse Process In the reverse process, the model aims to convert the latent variable $\boldsymbol{x}_S$, which is assumed to follow the distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, back into the original data $\boldsymbol{x}_0$. The joint probability distribution is given by:

$$p_\theta(\boldsymbol{x}_{0:S}) = p(\boldsymbol{x}_S) \prod_{s=1}^{S} p_\theta(\boldsymbol{x}_{s-1}|\boldsymbol{x}_s) \tag{15}$$

The conditional distributions involved are inferred using Bayes rule as follows:

$$q(\boldsymbol{x}_{s-1}|\boldsymbol{x}_s, \boldsymbol{x}_0) = \frac{q(\boldsymbol{x}_s|\boldsymbol{x}_{s-1}, \boldsymbol{x}_0) q(\boldsymbol{x}_{s-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_s|\boldsymbol{x}_0)} \tag{16}$$

To optimize the generative model $p_\theta(\boldsymbol{x}_0)$ for fitting the data distribution $q(\boldsymbol{x}_0)$, we minimize a variational upper bound on the negative log-likelihood:

$$\mathcal{L}_{vb} = \mathbb{E}_{q(\boldsymbol{x}_0)}\Big[D_{KL}[q(\boldsymbol{x}_S|\boldsymbol{x}_0)||p(\boldsymbol{x}_S)] + \sum_{s=1}^{S} \mathbb{E}_{q(\boldsymbol{x}_s|\boldsymbol{x}_0)}\big[D_{KL}[q(\boldsymbol{x}_{s-1}|\boldsymbol{x}_s, \boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{s-1}|\boldsymbol{x}_s)]\big]\Big]. \tag{17}$$

However, continuous diffusion models are not applicable in discrete spaces, such as quantized token indices $\mathbf{k} = (k_1, k_2, \ldots, k_N)$ where each k_i assumes one of $|\mathcal{C}|$ discrete values. This limitation arises because Gaussian noise cannot corrupt discrete elements in a meaningful way. Thus, modeling in discrete spaces necessitates the development of discrete diffusion processes.

B Mathematical Proofs

In this section, we provide a detailed mathematical proofs for Eq. (6), which can quickly calculate $q(\mathbf{k}_s|\mathbf{k}_0)$ according to Eq. (2).

Concretely, we use mathematical induction to prove Eq. (6). At first, we have following conditional information:

$$\begin{aligned} &\alpha_s, \beta_s \in [0,1], \alpha_s = 1 - |\mathcal{C}|\beta_s - \gamma_s, \\ &\overline{\alpha}_s = \prod_{i=1}^{s} \alpha_s, \overline{\gamma}_s = 1 - \prod_{i=1}^{s}(1-\gamma_s), \overline{\beta}_s = (1 - \overline{\alpha}_s - \overline{\gamma}_s)/|\mathcal{C}|. \end{aligned} \tag{18}$$

Now we want to prove that $\overline{\mathbf{M}}_s \boldsymbol{c}(\mathbf{k}_0) = \overline{\alpha}_s \boldsymbol{c}(\mathbf{k}_0) + (\overline{\gamma}_s - \overline{\beta}_s)\boldsymbol{c}(|\mathcal{C}|+1) + \overline{\beta}_s$. Firstly, when $s = 1$, we have:

$$\overline{\mathbf{M}}_1 \boldsymbol{c}(\mathbf{k}_0) = \begin{cases} \overline{\alpha}_1 + \overline{\beta}_1, & \mathbf{k} = \mathbf{k}_0 \\ \overline{\beta}_1, & \mathbf{k} \neq \mathbf{k}_0 \text{ and } \mathbf{k} \neq |\mathcal{C}|+1 \\ \overline{\gamma}_1, & \mathbf{k} = |\mathcal{C}|+1 \end{cases} \tag{19}$$

which is clearly hold. Suppose the Eq. (6) holds at step s, then for $s = s + 1$, we have:

$$\overline{\mathbf{M}}_{s+1} \boldsymbol{c}(\mathbf{k}_0) = \mathbf{M}_{\mathbf{k}+1} \overline{\mathbf{M}}_t \boldsymbol{c}(\mathbf{k}_0). \tag{20}$$

Now we consider three conditions:
(1) when $\mathbf{k} = \mathbf{k}_0$ in step $s+1$, we have:

$$\begin{aligned} \mathbf{M}_{s+1}\boldsymbol{c}(\mathbf{k}_0)_{(\mathbf{k})} &= \overline{\beta}_s\beta_{s+1}(|\mathcal{C}|-1) + (\alpha_{s+1}+\beta_{s+1})(\overline{\alpha}_s + \overline{\beta}_s) \\ &= \overline{\beta}_s(|\mathcal{C}|\beta_{s+1} + \alpha_{s+1}) + \overline{\alpha}_s(\alpha_{s+1}+\beta_{s+1}) \\ &= \frac{1}{|\mathcal{C}|}(\overline{\beta}_s(1-\gamma_{s+1}) + \overline{\alpha}_s\beta_{s+1} - \overline{\beta}_{s+1}) * |\mathcal{C}| + \overline{\alpha}_{s+1} + \overline{\beta}_{s+1} \\ &= \frac{1}{|\mathcal{C}|}[(1-\overline{\alpha}_s - \overline{\gamma}_s)(1-\gamma_{s+1}) + |\mathcal{C}|\overline{\alpha}_s\beta_{s+1} - (1-\overline{\alpha}_{s+1} - \overline{\gamma}_{s+1})] + \overline{\alpha}_{s+1} + \overline{\beta}_{s+1} \\ &= \frac{1}{|\mathcal{C}|}[(1-\overline{\gamma}_{s+1}) - \overline{\alpha}_s(1-\gamma_{s+1}-K\beta_{s+1}) - (1-\overline{\gamma}_{s+1}) + \overline{\alpha}_{s+1}] + \overline{\alpha}_{s+1} + \overline{\beta}_{s+1} \\ &= \overline{\alpha}_{s+1} + \overline{\beta}_{s+1}. \end{aligned} \tag{21}$$

(2) when $\mathbf{k} = |\mathcal{C}|+1$ in step $s+1$, we have:

$$\mathbf{M}_{s+1}\boldsymbol{c}(\mathbf{k}_0)_{(\mathbf{k})} = \overline{\gamma}_s + (1-\overline{\gamma}_s)\gamma_{s+1} = 1 - (1-\overline{\gamma}_{s+1}) = \overline{\gamma}_{s+1}. \tag{22}$$

(3) when $\mathbf{k} \neq \mathbf{k}_0$ and $\mathbf{k} \neq |\mathcal{C}|+1$ in step $s+1$, we have:

$$\begin{aligned} \mathbf{M}_{s+1}\boldsymbol{c}(\mathbf{k}_0)_{(\mathbf{k})} &= \overline{\beta}_s(\alpha_{s+1}+\beta_{s+1}) + \overline{\beta}_s\beta_{s+1}(|\mathcal{C}|-1) + \overline{\alpha}_s\beta_{s+1} \\ &= \overline{\beta}_s(\alpha_{s+1} + |\mathcal{C}|\beta_{s+1}) + \overline{\alpha}_s\beta_{s+1} \\ &= \frac{1-\overline{\alpha}_s - \overline{\gamma}_s}{|\mathcal{C}|} * (1-\gamma_{s+1}) + \overline{\alpha}_s\beta_{s+1} \\ &= \frac{1}{|\mathcal{C}|}(1-\overline{\gamma}_{s+1}) + \overline{\alpha}_s(\beta_{s+1} - \frac{1-\gamma_{s+1}}{|\mathcal{C}|}) \\ &= \overline{\beta}_{s+1}. \end{aligned} \tag{23}$$

The proof of Eq. (6) is completed. Notably, according to Eq. (6), the computation cost of $q(\mathbf{k}_s|\mathbf{k}_0)$ can be reduced from $O(|\mathcal{C}|^2 S)$ to $O(|\mathcal{C}|)$.

C Algorithms for Discrete Diffusion Process

In this section, we provide complete training and inference algorithms for discrete diffusion process.

C.1 Training Procedure

The discrete diffusion process aims to model quantized 3D pose tokens in a discrete space. This involves utilizing a 2D image I and its corresponding 3D human pose $\mathbf{P}$ as inputs. The image I serves as a contextual condition, while $\mathbf{P}$ is converted into discrete tokens for modeling.

Algorithm 1 Training Algorithm for the discrete diffusion process.

Require:
A transition matrix $\mathbf{M}_s$, the number of steps S, parameters of pose denoiser θ, training epoch T, pose dataset $\boldsymbol{D}$ (including 2D image I and 3D human pose $\mathbf{P}$), and the well-learned pose encoder $f_{PE}(\cdot)$.

for $i = 1$ to T **do**
  for $(I, \mathbf{P})$ in $\boldsymbol{D}$ **do**
    $\mathbf{k}_0 = \text{FSQ}(f_{PE}(\mathbf{P}))$, $\boldsymbol{y}$ =ImageEncoder(I);
    sample s from Uniform$\{1, 2, ..., S-1, S\}$;
    calculate $q(\mathbf{k}_s|\mathbf{k}_0)$ based on Eq. (6);
    estimate $f_\theta(\mathbf{k}_{s-1}|\mathbf{k}_s, \boldsymbol{y})$;
    calculate loss according to Eq. (10);
    update θ;
  end for
end for
return θ.

Algorithm 2 Inference Algorithm for the discrete diffusion process.

Require:
The number of steps S, input 2D image I, the pose decoder $f_{PD}(\cdot)$, parameters of pose denoiser θ, stationary distribution $p(\mathbf{k}_S)$;

$s = S$, $\boldsymbol{y}$ = ImageEncoder(I);
sample $\mathbf{k}_s$ from $p(\mathbf{k}_S)$;
while $s > 0$ **do**
  $\mathbf{k}_s \leftarrow$ sample from $p_\theta(\mathbf{k}_{s-1}|\mathbf{k}_s, \boldsymbol{y})$
  $s \leftarrow (s-1)$
end while
return $f_{PD}(\mathbf{k}_s)$.

Firstly, the 3D human pose $\mathbf{P}$ is encoded by $f_{PE}(\cdot)$ and subsequently quantized using the FSQ technique, resulting in multiple discrete tokens. Concurrently, a pre-trained Image Encoder extracts contextual features from I, producing a conditional feature sequence $\boldsymbol{y}$. During the forward process, we sample s from a uniform distribution $\{1, 2, ..., S-1, S\}$ and compute $q(\mathbf{k}_s|\mathbf{k}_0)$ based on Eq. (6). In the reverse process, the pose denoiser $f_\theta(\mathbf{k}_{s-1}|\mathbf{k}_s, \boldsymbol{y})$ is trained to estimate $q(\mathbf{k}_{s-1}|\mathbf{k}_s, \mathbf{k}_0)$. Finally, the overall loss is calculated according to Eq. (10), and the parameters of the pose denoiser θ are updated accordingly.

The complete training algorithm for the discrete diffusion process is presented in Algorithm 1.

C.2 Inference Procedure

In the inference process, our objective is to recover the 3D human pose $\hat{\mathbf{P}}$ from an input 2D image and discrete tokens.

Initially, all pose tokens are either masked or initialized randomly, which is achieved by sampling from the stationary distribution $p(\mathbf{k}_S)$. The 2D image I is encoded using the pre-trained Image Encoder. Subsequently, we predict $f_\theta(\mathbf{k}_{s-1}|\mathbf{k}_s, \boldsymbol{y})$ step by step until the pose tokens are fully recovered. Finally, the reconstructed tokens are decoded using the pose decoder $f_{PE}(\cdot)$, yielding the recovered 3D pose $\hat{\mathbf{P}}$.

The complete inference algorithm for the discrete diffusion process is presented in Algorithm 2.

D Additional Implementation Details

All experiments are carried out on one NVIDIA A100 PCIe GPU. The proposed Di^{2}Pose is completely implemented in PyTorch [53]. In this section, we provide the detailed training settings for the pose quantization step and the discrete diffusion process.

Table 6: Results on Human3.6M in millimeters under PA-MPJPE. The best results are in bold, and the second-best ones are underlined.

Methods	Dir	Disc	Eat	Greet	Phone	Photo	Pose	Pur	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg
Martinez *et al.* [48] *ICCV'17*	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
Pavlakos *et al.* [54] *CVPR'17*	34.7	39.8	41.8	**38.6**	42.5	47.5	38.0	36.6	50.7	56.8	42.6	39.6	43.9	32.1	36.5	41.8
Liu *et al.* [45] *ECCV'18*	35.9	40.0	38.0	41.5	42.5	51.4	37.8	36.0	48.6	56.6	41.8	38.3	<u>42.7</u>	31.7	36.2	41.2
Zhang *et al.* [82] *TPAMI'23*	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	<u>39.1</u>
Choi *et al.* [14] *IROS'23*	36.7	41.1	37.6	42.2	40.5	44.1	37.8	36.3	47.0	60.5	39.8	38.9	42.7	33.7	35.1	40.9
Gong *et al.* [22] *CVPR'23*	**33.9**	**38.2**	<u>36.0</u>	<u>39.2</u>	<u>40.2</u>	**46.5**	<u>35.8</u>	<u>34.8</u>	<u>48.0</u>	<u>52.5</u>	<u>41.2</u>	<u>36.5</u>	**40.9**	<u>30.3</u>	<u>33.8</u>	39.2
Di^{2}Pose (Ours)	<u>34.5</u>	<u>38.4</u>	**35.1**	40.8	**39.8**	<u>47.0</u>	**34.9**	**34.7**	**47.1**	**52.3**	**40.4**	**36.1**	42.9	**30.0**	**33.4**	**39.0**

For the pose quantization step, we employ the AdamW [46] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, adhering to a base learning rate of 1e-3 and a weight decay parameter of 0.15. The training process is configured with a batch size of 256 across a total of 20 epochs.

For the discrete diffusion process, we still utilize the the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.96$, adhering to a base learning rate of 5.5e-4 and a weight decay parameter of 4.5e-2. The training process is configured with a batch size of 64 across a total of 50 epochs.

E Additional Experimental Results

We exhibit more experimental results to verify the effectiveness of our Di^{2}Pose.

E.1 Quantitative Results

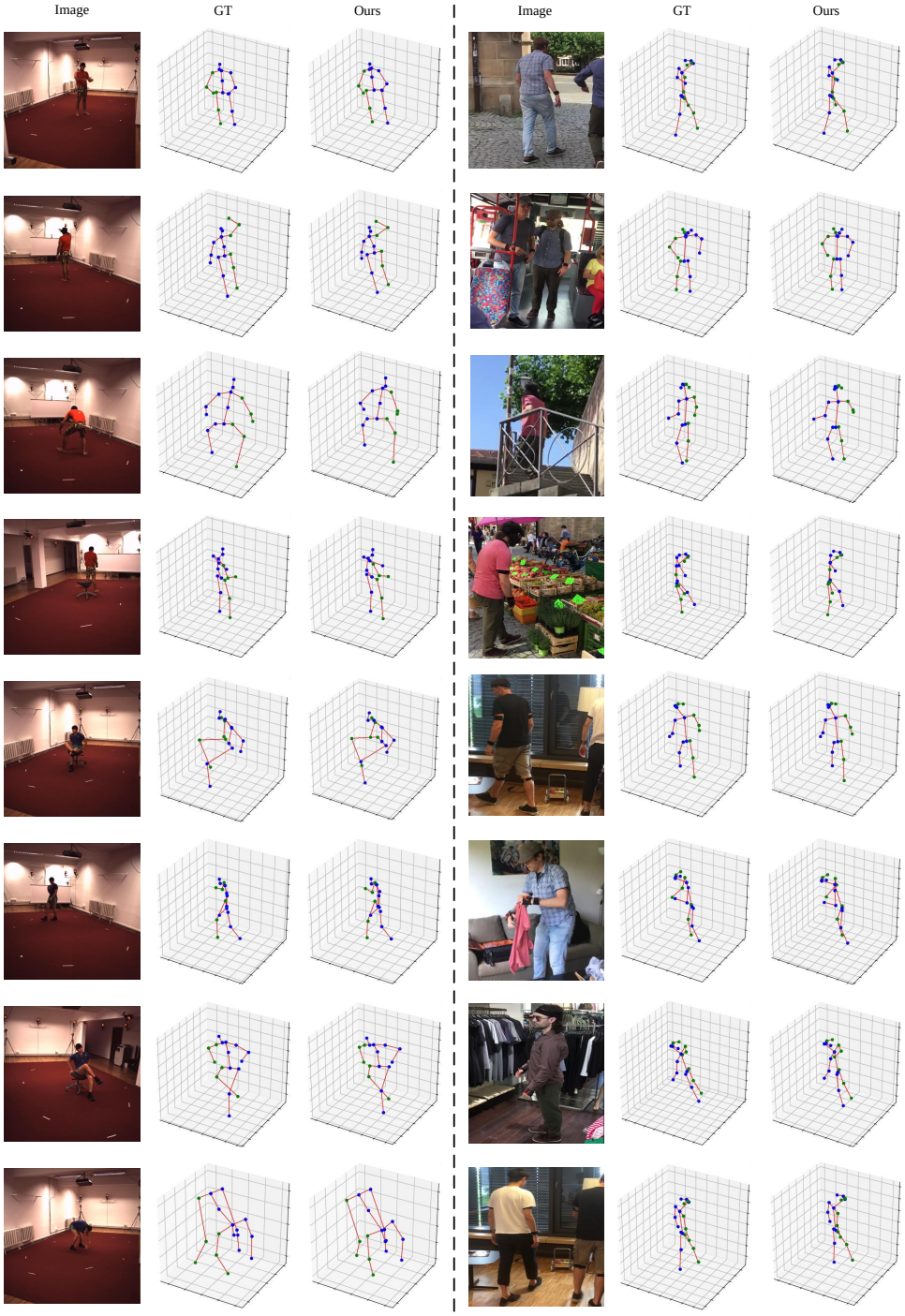
As shown in Table 6, we benchmark Di^{2}Pose against SOTA 3D HPE methods on the Human3.6M under PA-MPJPE protocol. Our Di^{2}Pose achieves 39.0mm in average PA-MPJPE, surpassing the performance of the compared SOTA 3D HPE methods, which indicates that Di^{2}Pose is able to enhance monocular 3D HPE in indoor scenes.

E.2 Qualitative Results

In this part, we present additional qualitative results on the Human3.6M and 3DPW datasets. As illustrated in Figure 6, our Di^{2}Pose model demonstrates the ability to accurately recover 3D human poses in both indoor and in-the-wild scenarios. Particularly noteworthy is its performance under various occlusion conditions, including self-occlusion and object occlusion. Even in these challenging situations, Di^{2}Pose consistently produces reasonable 3D pose estimations, highlighting its robustness to occlusions.

F Broader Impacts

This research focuses on estimating physically valid 3D human poses from monocular frames, especially under occlusion scenes. Such a method can be positively used for sports analysis, surveillance, healthcare, autonomous driving, etc. where clear, unobstructed views of the subject may not always be available. It can also lead to malicious use cases, such as illegal surveillance and video synthesis. Thus, it is essential to deploy these algorithms with care and make sure that the extracted human poses are with consent and not misused. Moreover, the diffusion-based model has a longer runtime compared to other CNN or GCN-based methods, causing more computational resources and energy consumption.



(a) Human3.6M (b) 3DPW

Figure 6: Qualitative results on two datasets. Joints on the right side are marked in green, while other joints are highlighted in blue.

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: Refer to the Abstract and Sec.1.

   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: Refer to Sec.5.

   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: [Yes]

Justification: Refer to Sec.3 and Sec.B.

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: Refer to Sec.4.1, Sec.4.2 and Sec.D.

   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: [Yes]

Justification: We will release code upon paper acceptance.

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: Refer to Sec.4.2 and Sec.D.

   Guidelines:

   - 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: Refer to Sec.4.

   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: Refer to Sec.D in 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: We make sure to conduct this paper conform 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: Refer to Sec.F in the Appendix.

    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: This paper poses no such risks.

    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: All used datasets and models in this paper are explicitly mentioned and properly cited.

    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 involve this issue.

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: We use publicly available 3D human pose datasets in this paper.

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 this issue.

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.