This ICCV workshop paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

Diff3DHPE: A Diffusion Model for 3D Human Pose Estimation

Jieming Zhou[1], Tong Zhang[2], Zeeshan Hayder[3], Lars Petersson[3], Mehrtash Harandi[4]
[1]Australian National University, [2]EPFL, [3]CSIRO, [4]Monash University
jieming.zhou@anu.edu.au, tong.zhang@epfl.ch,
{zeeshan.hayder, Lars.Petersson}@data61.csiro.au, mehrtash.harandi@monash.edu

Abstract

Accurately estimating 3D human pose (3D HPE) and joint locations using only 2D keypoints is challenging. The noise in the predictions produced by conventional 2D human pose estimators often impeded the accuracy. In this paper, we present a diffusion-based model for 3D pose estimation, named Diff3DHPE, inspired by diffusion models' noise distillation abilities. The proposed model takes a temporal sequence of 2D keypoints as the input of a GNN backbone model to extract the 3D pose from Gaussian noise using a diffusion process during training. The model then refines it using a reverse diffusion process. To overcome oversmoothing issues in GNNs, Diff3DHPE is integrated with a discretized partial differential equation, which makes it a particular form of Graph Neural Diffusion (GRAND). Extensive experiments show that our model outperforms current state-of-the-art methods on two benchmark datasets, Human3.6M and MPI-INF-3DHP, achieving up to 39.1% improvement in MPJPE on MPI-INF-3DHP. The code is available at https://github.com/socoolzjm/Diff3DHPE.

1. Introduction

Human pose estimation (HPE) estimates the configuration of human body parts from data collected by various sensors such as RGB and depth cameras. This task has been widely studied due to its relevance to real-world applications such as augmented reality, virtual reality, and motion analysis [50]. Notably, 3D human pose estimation (3D HPE) from 2D imagery captured by a monocular RGB camera is attracting attention. A single image or a video sequence can then provide a cost-efficient way of estimating the coordinates of human joints in 3D space. Recent works have demonstrated that approaches using 2D keypoints generated by off-the-shelf 2D human pose estimators are superior to end-to-end methods that instead take images as the inputs [3, 25, 28, 19, 52]. However, due to the ill-posed nature of predicting 3D from 2D, we can only predict accurate 3D poses by imposing priors, and occlusions of body parts deteriorate this situation. Therefore, leveraging temporal information, the current state-of-the-art works use a sequence of images from a video as the input to enforce the temporal smoothness and estimate more accurate 3D poses [33, 8, 1, 43, 24].

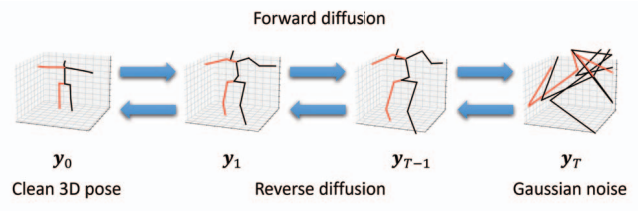

Figure 1. General framework of the diffusion model. In the forward diffusion process, the diffusion model learns the Gaussian noise that distorts a clean 3D pose $\boldsymbol{y}_0$. Then, the diffusion model reconstructs the clean 3D pose from a Gaussian noise after T iterations during the reverse diffusion process.

Given the 2D keypoints of a person in a video, the input data can be represented as a spatial-temporal graph where nodes are joints. Edges are bones between different joints and connections across frames of the same joint. Therefore, Graph Neural Networks (GNNs) naturally fit the 3D HPE task. Approaches like [6], [7], [23], and [49] illustrate the effectiveness of GNNs on 3D HPE. Recently, the transformer [42] is introduced to the 3D HPE task [51, 48, 37, 44, 21, 9], which achieves extraordinary improvement due to the ability to aggregate long-range information. However, the performance of all these approaches still heavily relies on the quality of the input 2D keypoints. Current approaches could be more robust to noisy inputs, such as unstable 2D pose estimations and missing body parts, which commonly exist in the real world.

To relieve the influence of the imperfect input 2D keypoints, we propose the Diff3DHPE that uses 2D keypoints as conditions of the diffusion model [39] and reconstructs 3D poses from Gaussian noise through iterations of sampling. Fig. 1 illustrates a general framework for applying the diffusion model to the 3D HPE task. Instead of predicting the parameters of Gaussian noise, we modify the target

of the backbone model inside the diffusion model, which directly predicts the ground truth 3D poses given noisy 3D poses and 2D keypoint conditions during the training stage. By doing so, we effectively reduce the computational cost and increase the accuracy in the sampling stage in practice. In this paper, we evaluate MixSTE [48] and PoseFormer [51] as backbone models to demonstrate the flexibility of our proposed diffusion model scheme in 3D HPE. Furthermore, by slightly modifying the attention function inside, we use a partial differential equation (PDE) to control the message-passing speed between joints and suppress the over-smoothing issue caused by aggregation of highly similar features. We illustrate that this modification transforms the sampling function of the diffusion model into a particular form of Graph Neural Diffusion (GRAND) [2] and further boosts performance in practice.

Our contributions are summarized as follows:

- We propose a novel diffusion model scheme for 3D HPE, Diff3DHPE, which uses various transformer-based backbone models to aggregate information in the spatial-temporal space, and generates 3D pose predictions under 2D keypoint conditions from Gaussian noise.
- We modify the target of the backbone model inside the diffusion model. By directly predicting the original ground truth 3D pose and iteratively refining it, Diff3DHPE dramatically accelerates the sampling time of the diffusion model.
- We bridge the diffusion model and GRAND by modifying the transformer used in the backbone model, making the sampling function a discretized explicit solver of a PDE.
- We evaluate Diff3DHPE on two popular 3D HPE datasets: Human3.6M [16] and MPI-INF-3DHP [27], which empirically proves its state-of-the-art performance and outstanding robustness to noisy 2D keypoint inputs.

2. Related Work

3D Human Pose Estimation. Reconstructing the 3D coordinates of a person's joints captured from a single view is one of the most widely studied 3D HPE tasks [20, 32, 3, 28, 19, 52, 22, 43]. Generally, these approaches are divided into two categories. The first approach employs an *end-to-end* network to predict 3D poses from the input images directly. Li and Chan [20] use multi-task learning to train a convolutional neural network (CNN) where the model detects body parts and simultaneously estimates 3D pose from an input image. To simplify the problem, the second approach, *2D-to-3D lifting*, takes 2D keypoints generated by an off-the-shelf 2D pose estimator, such as CPN [5] and HRNet [41], as the input. As a result, 2D-to-3D lifting methods benefit from high-quality 2D pose estimations and perform better than end-to-end designs. To make up for the missing depth of a given 2D keypoint, Chen and Ramanan [3] match it to the closest 3D pose exemplar from a known 3D pose library and camera projection matrices. Thus, their method can obtain a highly accurate 3D pose estimation in a short amount of time. Even though using more priors, it is very likely that a single 2D pose can be mapped to multiple 3D poses. Moreover, occlusions of body parts within an image make this issue worse.

Because of the constraints of the physical world, the human poses should follow the time consistency given a continuous time. Thus, state-of-the-art methods tend to use a fix-frame-length video as their input to overcome the multiple-mapping issue. The temporal information contained in the 2D poses reduces the number of possible 3D poses and dramatically increases the accuracy of the 3D estimators. Normally, a *seq2frame* method takes a sequence as the input and only predicts the 3D pose of the central frame. VideoPose3D [33] employs a series of dilated convolution layers to aggregate long-term dependencies in the temporal dimension. Dabral *et al*. [8] first use a structure-aware network to estimate a 3D pose for each frame in the input sequence. To complement the spatial information with the temporal correlations, they pass the 3D poses to a temporal network to output a refined 3D pose for the central frame. Some approaches aim to make the 3D estimator further coherent and efficient. Using the *seq2seq* style, these estimators simultaneously predict 3D poses for all frames in the input sequence. [22] and [14] use LSTM [13] to recurrently predict the 3D poses of the input sequence. However, their methods suffer from the low computational efficiency caused by the LSTM. To avoid the computational efficiency issue, UGCN [43] adopts the spatial-temporal graph convolution proposed by [46] to model motion in multiple time scales in the input sequence. In this paper, we separately select *seq2frame* and *seq2seq* based backbone models to evaluate our proposed 3D HPE scheme comprehensively.

GNN and Transformer Recently, message-passing-based GNNs have shown their efficiency and capacity to learn graph representations by gradually aggregating node and edge features between neighbors through stacking layers [45]. Since the input 2D keypoint sequence can be represented as a spatial-temporal graph, it is natural to introduce GNNs to the 3D HPE task. LCN [7] combines fully connected and graph convolutional network designs (GCNs) characteristics. Each node in LCN has an individual filter to learn a representation flexibly. As a trade-off between using a shared weight filter and individual filters for each node, Zou and Tang [53] apply different modulation vectors to a shared weight filter, which reduces the number of network parameters. In addition, they use a learnable affinity

matrix to explore additional joint correlations further. Notably, the transformer is a particular form of GNN where the input graph is a complete graph. [42] first proposes the transformer to solve natural language processing (NLP) tasks in which self-attention can aggregate long-range dependency. Benefiting from the self-attention, PoseFormer [51] uses the Transformer to aggregate spatial and temporal features across all joints. Because the residual connections and normalization components in the Transformer can reduce the impact of over-smoothing, MixSTE [48] designs a deeper and more powerful 3D pose estimator by alternately stacking spatial and temporal Transformers. Moreover, as a *seq2seq* method, MixSTE achieves faster inference time compared to other *seq2frame* methods, such as PoseFormer, when having similar number of parameters.

Diffusion Models. First proposed by [39], diffusion models have become state-of-the-art methods for various generative tasks. From Gaussian noise, diffusion models gradually remove the noise through iterations, generating an output that obeys the target distribution. Combining with conditions such as corrupted images or text, diffusion models can output high-quality results for super-resolution [36], inpainting [17], and text-to-image synthesis tasks [35, 34]. These applications show that diffusion models are adaptable to tasks in which people commonly use regression-based methods. Thus, we propose a diffusion-model-based scheme for the 3D HPE task in which 2D keypoints are used as the condition. Although diffusion models have advantages in stable training compared to other generative methods, such as Generative adversarial networks (GAN) [11], it suffers from high computational costs caused by many iterations for high-quality results. DDIM [40] samples a subset from the original iteration steps, which makes a trade-off between quality and speed. However, we find that directly applying DDIM to the 3D HPE task will cause an insufficient accuracy issue when using a small step subset. In contrast, increasing the number of steps is infeasible for large-scale datasets like Human3.6M [16]. Therefore, we propose our alternative design of diffusion models, which can significantly reduce the iteration steps while keeping high-quality results.

Graph Diffusion. Graph diffusion focuses on how the information of each node diffuses on a graph according to a diffusion equation [10]. Random walks are often used as a diffusion equation, which depicts the transition probabilities among the nodes [26]. [30] reveals that GNNs are low-pass filters that perform the diffusion on graphs. This characteristic raises the problem of over-smoothing, in which node features tend to be similar as the depth of the GNNs grow. GRAND [2] proposes a broad new class of GNNs that modifies the self-attention to a PDE as the diffusion equation. Using Runge-Kutta to solve the discretized PDE, GRAND turns GNNs into band-pass filters, significantly relieving the over-smoothing issue. We apply the design of the diffusion equation in GRAND to transformer-based backbone models and show that our proposed diffusion model scheme is a particular form of GRAND.

3. Method

Given a 2D keypoint sequence of one person $\mathbf{x} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(F)}] \in \mathbb{R}^{F\times J\times 2}$, a typical *frame2seq* 3D HPE approach predicts the middle frame's 3D coordinates $\boldsymbol{y}^{(F/2)} \in \mathbb{R}^{J\times 3}$, where F is the number of frames, and J is the number of joints. We begin by introducing the diffusion model to this basic framework. We then present an alternative design for the diffusion model that reduces the sampling time while maintaining accuracy. To ensure our method matches the performance of state-of-the-art techniques, we adopt Transformer-based models as our backbone. Additionally, we address the over-smoothing issue caused by excessive iterations by introducing a PDE-based modification to the backbone models. Finally, our approach, Diff3DHPE, integrates all these innovations into a cohesive scheme. Specifically, as Fig 2 shows, Diff3DHPE can predict 3D coordinates of all frames $\mathbf{y} = [\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, ..., \boldsymbol{y}^{(F)}] \in \mathbb{R}^{F\times J\times 3}$ at the same time when the backbone model is designed in *seq2seq* style.

3.1. 3D HPE via Diffusion Model

Dealing with 3D HPE using 2D keypoints can be difficult since depth information is lost in the input. However, the diffusion model is a viable solution to this problem. It can gradually convert a Gaussian distribution to the target distribution, ultimately generating the missing information.

We first define the diffusion process q by adding Gaussian noise to a ground truth 3D coordinate $\boldsymbol{y}_0$ over T-step iterations as [12]:

$$q(\boldsymbol{y}_{1:T}|\boldsymbol{y}_0) = \prod_{t=1}^{T} q(\boldsymbol{y}_t|\boldsymbol{y}_{t-1}), \tag{1}$$

$$q(\boldsymbol{y}_t|\boldsymbol{y}_{t-1}) = \mathcal{N}\big(\boldsymbol{y}_t|\sqrt{\alpha_t}\boldsymbol{y}_{t-1}, (1-\alpha_t)\boldsymbol{I}\big), \tag{2}$$

where the scalars $\alpha_{1:T}$ are either predefined or learned variances, *s.t.* $1 > \alpha_1 > \alpha_2 > ... > \alpha_T > 0$. To simplify the training process, we sample $\boldsymbol{y}_t$ arbitrarily:

$$\boldsymbol{y}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{y}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{3}$$

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

Therefore, the target of the backbone model f_θ underlying the diffusion model is to predict the Gaussian noise when given the 2D keypoint sequence $\mathbf{x}$ and the noisy 3D coordinate $\boldsymbol{y}_t$ at step t. The objective function is formulated as follows:

$$L_{diff} = \mathbb{E}_{t\sim[1,T],\mathbf{x},\boldsymbol{y}_0,\boldsymbol{\epsilon}}\big[\,\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_t\|^2\,\big], \tag{4}$$

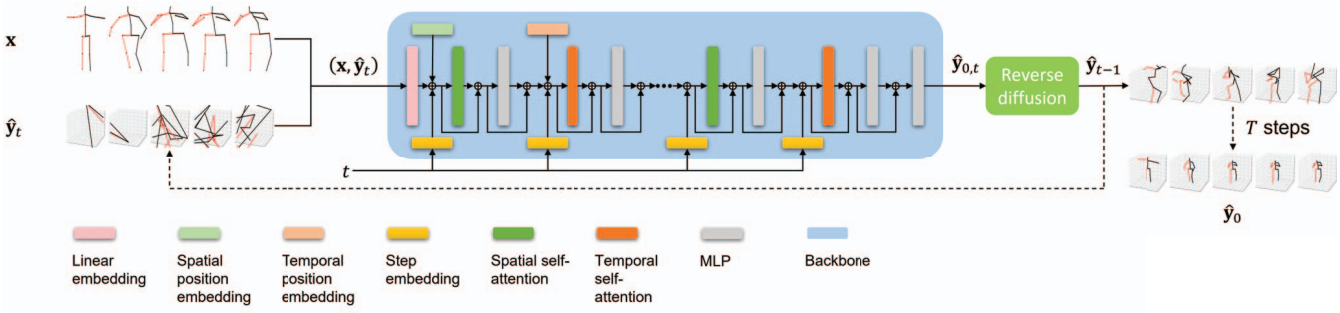

Figure 2. Overall framework of Diff3DHPE during the reverse diffusion process in *seq2seq* style. In the iteration step t, a 2D keypoint sequence $\mathbf{x}$ is concatenated with its corresponding noisy 3D predicted sequence $\hat{\mathbf{y}}_t$ along the channel dimension as the input $(\mathbf{x}, \hat{\mathbf{y}}_t)$. The backbone model takes $(\mathbf{x}, \hat{\mathbf{y}}_t)$ and t to predict a final 3D sequence $\hat{\mathbf{y}}_{0,t}$ at the step t. Then, $\hat{\mathbf{y}}_{t-1}$ is obtained from a predefined reverse diffusion function and sent to the next iteration for refining. To note, the backbone model is MixSTE in this example.

$$\hat{\boldsymbol{\epsilon}}_t = f_\theta(\mathbf{x}, \boldsymbol{y}_t, t). \tag{5}$$

Commonly, T is larger than 1000 to make the model smoothly learn the diffusion process in practice. However, this approach, in turn, dramatically increases the computation for the reverse diffusion process. Thus, we choose DDIM [40] to estimate the reverse diffusion process to reduce the iteration as:

$$\hat{\boldsymbol{y}}_{\tau_{i-1}} = \sqrt{\bar{\alpha}_{\tau_{i-1}}}\left(\frac{\hat{\boldsymbol{y}}_{\tau_i} - \sqrt{1-\bar{\alpha}_{\tau_i}}\hat{\boldsymbol{\epsilon}}_{\tau_i}}{\sqrt{\bar{\alpha}_{\tau_i}}}\right) + \sqrt{1-\bar{\alpha}_{\tau_{i-1}}}\hat{\boldsymbol{\epsilon}}_{\tau_i}, \tag{6}$$

$$\hat{\boldsymbol{y}}_0 = \frac{\hat{\boldsymbol{y}}_{\tau_1} - \sqrt{1-\bar{\alpha}_{\tau_1}}\hat{\boldsymbol{\epsilon}}_{\tau_1}}{\sqrt{\bar{\alpha}_{\tau_1}}}, \tag{7}$$

where τ_i is sampled every $\lceil T/S \rceil$ steps from $\{t_1, t_2, ..., t_T\}$, $\tau_1 < \tau_2 < ... < \tau_S \in [1, T]$, $S < T$, $\hat{\boldsymbol{y}}_t$ is the estimated 3D coordinates at step t, and $\hat{\boldsymbol{y}}_{\tau_S} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

3.2. Alternative Design of Diffusion Model

3D HPE typically focuses on the pose of the human body instead of its position in the global space. Thus, we remove the global offsets of 3D coordinates of joints in the sequence by centering a joint to $(0, 0, 0)$. We further normalize the value of 3D coordinates to $[-1, 1]$ because the $\hat{\boldsymbol{y}}_{\tau_S}$ is initialized with a Gaussian noise during the reverse diffusion process. Given a centralized 3D coordinate of a human joint, we hypothesise that the value is between -1000 mm and 1000 mm. When α_t is generated by the *cosine* schedule [29], we find that S must be larger than 1.55×10^5 to make the noise value introduced by $\sqrt{1-\bar{\alpha}_{\tau_1}}\hat{\boldsymbol{\epsilon}}_{\tau_1}$ in the Eq. 7 has 95% probability smaller than 1 mm.

To address this issue and further reduce the number of iterations of DDIM, we change the target of the backbone model in Eq. 5 to directly predict the $\boldsymbol{y}_0$ at step t during training:

$$\hat{\boldsymbol{y}}_{0,t} = f_\theta(\mathbf{x}, \boldsymbol{y}_t, t). \tag{8}$$

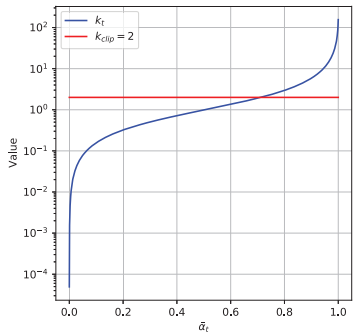

Figure 3. Curve of variable weight k_t throughout the training stage of Diff3DHPE. We clip the maximum value at $k_{clip} = 2$ to avoid extreme gradients.

To note, we switch $\boldsymbol{y}_t$ in Eq. 8 to $\hat{\boldsymbol{y}}_t$ during the reverse diffusion process. In the previous works [39, 40], the objective function uses the error between the target Gaussian noise $\boldsymbol{\epsilon}_t$ and the predicted noise $\hat{\boldsymbol{\epsilon}}_t$. Following Eq. 3, we have:

$$\begin{aligned}\boldsymbol{\epsilon}_t - \hat{\boldsymbol{\epsilon}}_t &= \frac{\boldsymbol{y}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{y}_0}{\sqrt{1-\bar{\alpha}_t}} - \frac{\boldsymbol{y}_t - \sqrt{\bar{\alpha}_t}\hat{\boldsymbol{y}}_{0,t}}{\sqrt{1-\bar{\alpha}_t}} \\ &= k_t(\hat{\boldsymbol{y}}_{0,t} - \boldsymbol{y}_0),\end{aligned} \tag{9}$$

$$k_t = \sqrt{\frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}}, \tag{10}$$

during the training stage. Therefore, we propose a new objective function formulated with the variable weight k_t as follows:

$$L_{diff} = \mathbb{E}_{t\sim[1,T],\mathbf{x},\boldsymbol{y}_0,\boldsymbol{\epsilon}}\left[\left(1+\text{Min}(k_t, k_{clip})\right)\|\boldsymbol{y}_0 - \hat{\boldsymbol{y}}_{0,t}\|_2^2\right]. \tag{11}$$

The curve of k_t is shown as Fig. 3. The intuition is that a more significant penalty should be applied to the prediction at an earlier step where the noise variance is small. We clip the maximum k_t at $k_{clip} = 2$ to avoid an overly large gradient. By adding a constant 1, we punish the network

even though the input 3D poses are very noisy because the input 2D poses are still relatively clean.

According to Eq. 3, the prediction of the Gaussian noise at step τ_i during the reverse diffusion process can be formulated as:

$$\hat{\boldsymbol{\epsilon}}_{\tau_i} = \frac{\hat{\boldsymbol{y}}_{\tau_i} - \sqrt{\bar{\alpha}_{\tau_i}}\hat{\boldsymbol{y}}_{0,\tau_i}}{\sqrt{1-\bar{\alpha}_{\tau_i}}}. \tag{12}$$

Combining Eq. 12 with 6 and 7, our alternative design of diffusion model has the following reverse diffusion process:

$$\hat{\boldsymbol{y}}_{\tau_{i-1}} = \sqrt{\bar{\alpha}_{\tau_{i-1}}}\hat{\boldsymbol{y}}_{0,\tau_i} + \sqrt{1-\bar{\alpha}_{\tau_{i-1}}}\frac{\hat{\boldsymbol{y}}_{\tau_i} - \sqrt{\bar{\alpha}_{\tau_i}}\hat{\boldsymbol{y}}_{0,\tau_i}}{\sqrt{1-\bar{\alpha}_{\tau_i}}}, \tag{13}$$

$$\hat{\boldsymbol{y}}_0 = \hat{\boldsymbol{y}}_{0,\tau_1}, \tag{14}$$

where no Gaussian noise term is introduced to the final prediction.

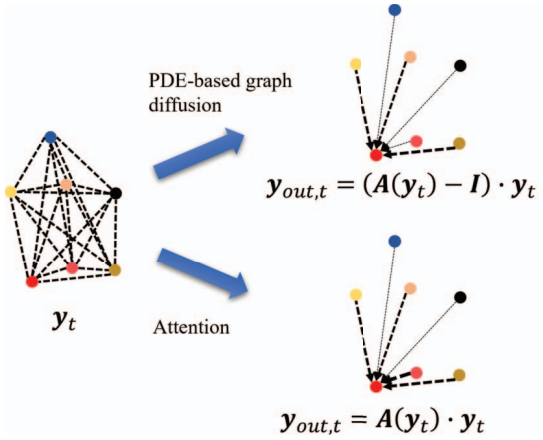

Figure 4. Feature aggregation in a simplified GNN layer. The red dot is the central node. Dots in other colors are neighbors. Solid lines are edges. Longer dash lines indicate lower feature similarities between central node and neighbors. Bolder dash lines indicate larger weights. Compared to attention, PDE-based graph diffusion suppresses highly similar information passed to the central node.

3.3. Bridge between Diffusion Model and GRAND

Given a graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$, $\mathbb{V}$ is a node set, $\mathbb{E}$ is an edge set, $|\mathbb{V}| = N$, $\boldsymbol{y}_t \in \mathbb{R}^{N\times C}$ is the node feature matrix at step t output by a GNN in the reverse diffusion process, and C is the number of channels. Let a simplified attention block be the GNN. The output $\boldsymbol{y}_{out,t}$ is generated by:

$$\boldsymbol{y}_{out,t} = \boldsymbol{A}(\boldsymbol{y}_t) \cdot \boldsymbol{y}_t, \tag{15}$$

where $\boldsymbol{A}(\boldsymbol{y}_t)$ is the $N \times N$ attention matrix. As Fig. 4 illustrates, neighbors with high similarity to the central node have a large weight in the aggregation, which causes the over-smoothing problem when the depth of the GNN increases. To overcome this issue, we adapt the graph diffusion equation proposed by GRAND [2]. Here, we define the PDE of the graph diffusion equation as:

$$\frac{\partial \hat{\boldsymbol{y}}_t}{\partial t} = \big(\boldsymbol{A}(\hat{\boldsymbol{y}}_t) - \boldsymbol{I}\big) \cdot \hat{\boldsymbol{y}}_t. \tag{16}$$

GRAND learns to produce node embeddings $\hat{\boldsymbol{y}} = \phi(\hat{\boldsymbol{y}}_0)$,

$$\hat{\boldsymbol{y}}_0 = \boldsymbol{y}_T + \int_T^0 \frac{\partial \hat{\boldsymbol{y}}_t}{\partial t} dt, \tag{17}$$

where ϕ is a learnable network.

In the 3DHPE task, features of nearby nodes from the same joint in the temporal dimension are highly similar because position differences are insignificant among the nearby frames. By introducing the GRAND, highly similar information is suppressed during the aggregation. Meanwhile, the function of $\hat{\boldsymbol{y}}_{\tau_1}$ can be derived from Eq. 13:

$$\hat{\boldsymbol{y}}_{\tau_1} = a\boldsymbol{y}_{\tau_S} + \sum_{i=2}^{S} b_i c_i \hat{\boldsymbol{y}}_{0,\tau_i}, \tag{18}$$

$$\boldsymbol{y}_{\tau_S} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{19}$$

$$a = \sqrt{\frac{1-\bar{\alpha}_{\tau_1}}{1-\bar{\alpha}_{\tau_S}}}, \tag{20}$$

$$b_i = \sqrt{\bar{\alpha}_{\tau_{i-1}}} - \sqrt{\frac{\bar{\alpha}_{\tau_i}(1-\bar{\alpha}_{\tau_{i-1}})}{1-\bar{\alpha}_{\tau_i}}}, \tag{21}$$

$$c_i = \sqrt{\frac{1-\bar{\alpha}_{\tau_{S-i+1}}}{1-\bar{\alpha}_{\tau_{S-i}}}}. \tag{22}$$

And, the clean 3D pose prediction is:

$$\hat{\boldsymbol{y}}_0 = f_\theta(\mathbf{x}, \hat{\boldsymbol{y}}_{\tau_1}, t). \tag{23}$$

Thus, the reverse diffusion process of our alternative design of the diffusion model becomes a particular discretized form of GRAND when the backbone model f_θ consists of PDE-based graph diffusion defined in Eq. 16.

In the final design of our Diff3DHPE, we select Transformer-based models as backbones and apply the PDE-based graph diffusion equation to their transformer blocks.

4. Experiments

4.1. Datasets

We evaluate our Diff3DHPE and other state-of-the-art 3D HPE methods on Human3.6 [16] and MPI-INF-3DHP [27].

Human3.6M is a large-scale dataset widely used for the 3D HPE task. This dataset has 3.6 million human poses captured by four cameras from different views in an indoor environment providing highly accurate measurements. Following previous works [51, 48, 24, 33], we select the subjects $S1$, $S5$, $S6$, $S7$, and $S8$ as the training set that contains 15 actions in each subject and 17 joints in each frame. We use the mean per joint position error (MPJPE) and Procrustes MPJPE (P-MPJPE) [50] to measure the performance of methods on this dataset. MPJPE calculates the average Euclidean distance between estimated joint 3D coordinates and their ground truth in millimetres. P-MPJPE computes the post-processed MPJPE after rigid alignment between the estimation and ground truth.

MPI-INF-3DHP contains more than 1.4 million frames captured from 14 cameras in indoor and outdoor environments. Eight actors in the dataset perform eight activities, such as walking, sitting, complex exercise poses, and dynamic actions. We report the percentage of correct keypoints (PCK) within 150 mm and the area under curve (AUC) along with MPJPE for evaluating methods on this dataset.

4.2. Experimental Setup

We implement our models using Pytorch [31] and run on eight NVIDIA GeForce RTX 2080 Ti. Following [51, 48], we horizontally flip poses as data augmentation for training and testing. We train each mode for 200 epochs with the Adam [18] optimizer and 0.1 weight decay. After each epoch, the learning rate will multiply with the decay factor 0.99. The dropout rate is 0.1. The number of forward diffusion steps $T = 1000$. We follow the procedure used in [51] to select other hyper-parameters, *i.e.* learning rate and the number of the reverse diffusion steps S. The hyper-parameter search space and final selections are listed in the supplementary material.

4.3. Results and Analysis

To showcase the effectiveness of Diff3DHPE, we employ MixSTE as the backbone model, which is currently state-of-the-art. To ensure a fair comparison, we use the same number of frames, F, as used in MixSTE for Human3.6 (F = 81, 243) and MPI-INF-3DHP (F = 27). For evaluation, we input 2D keypoints detected by CPN [5] and ground truth 2D keypoints separately when evaluating on Human3.6M. On MPI-INF-3DHP, we adopt the protocol used in [51, 48, 37, 4] and use ground truth 2D keypoints of 17 joints as input, evaluating our networks on valid frames in the test set.

Results on Human3.6M. Table 1 compares our models and the most recent state-of-the-art using CPN input. For the $F = 81$ configuration, Diff3DHPE-M outperforms other methods on most actions in terms of MPJPE. Notably, Diff3DHPE-M achieves the best results across all actions and surpasses MixSTE by 1.2 mm on average when evaluating P-MPJPE. Furthermore, in the $F = 243$ setting, Diff3DHPE-M widens the overall MPJPE gap between the second-best model from 0.4 mm to 0.9 mm. Finally, Table 2 showcases the results of models using ground truth 2D keypoints. In this case, Diff3DHPE-M outperforms MixSTE in terms of MPJPE by a more considerable margin of 1.7 mm and 1.4 mm in $F = 81$ and $F = 243$ settings, respectively.

Results on MPI-INF-3DHP. Typically, models using more frames as the input can achieve better results. However, our Diff3DHPE-M exceeds the second-best model by 39.1% in MPJPE when only using 1/3 length of its input on MPI-INF-3DHP, which is shown in Table 3. This result further demonstrates Diff3DHPE-M's superiority over other methods.

4.4. Ablation Study

4.4.1 Effect of PDE-based Diffusion model

To demonstrate the effectiveness of our PDE-based diffusion model, we conducted additional experiments on Human3.6M with CPN input by training Diff3DHP models without the PDE-based design. During training, all Diff3DHPE models use the same number of forward diffusion steps, $T = 1000$. We then ran the reverse diffusion process with varying reverse steps, S. As shown in Fig. 5, the PDE-based graph diffusion models outperformed the corresponding models without this design, with a smaller performance drop as the number of iterations increased. Since increasing the number of iterations can be considered as increasing the depth of the backbone GNN model, the PDE-based graph diffusion design effectively mitigated the impact of over-smoothing.

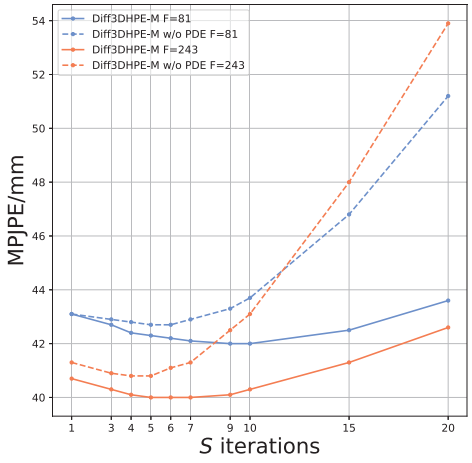

Figure 5. Performance changes when Diff3DHPE-M uses different number of reverse diffusion iterations on Human3.6M, CPN input.

Table 1. Results on Human3.6M with CPN detected 2D keypoints. The best and second-best results of the same number of frames setting are in **Bold** and underlined, respectively. ↓: lower is better. Diff3DHPE-M: Diff3DHPE with MixSTE backbone.

MPJPE↓		Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Chen *et al*. [4] (F=81)	TCSVT2021	42.1	43.8	41.0	43.8	46.1	53.5	42.4	43.1	53.9	60.5	45.7	42.1	46.2	32.2	33.8	44.6
Poseformer [51] (F=81)	ICCV2021	41.5	44.8	39.8	42.5	46.5	51.6	42.1	42.0	53.3	60.7	45.5	43.3	46.1	31.8	32.2	44.3
MixSTE [48] (F=81)	CVPR2022	**39.8**	43.0	**38.6**	**40.1**	43.4	50.6	40.6	41.4	**52.2**	56.7	43.8	**40.8**	43.9	**29.4**	30.3	42.4
P-STMO-S [37] (F=81)	ECCV2022	41.7	44.5	41.0	42.9	46.0	51.3	42.8	41.3	54.9	61.8	45.1	42.8	43.8	30.8	30.7	44.1
Diff3DHPE-M (F=81)		40.2	**42.7**	**38.6**	40.8	**42.6**	**50.0**	**40.3**	**40.2**	52.5	**55.1**	**43.6**	41.3	**42.9**	29.5	**29.5**	**42.0**
SRNet [47] (F=243)	ECCV2020	46.6	47.1	43.9	41.6	45.8	49.6	46.5	40.0	53.4	61.1	46.1	42.6	43.1	31.5	32.6	44.8
Liu *et al*. [24] (F=243)	CVPR2020	41.8	44.8	41.1	44.9	47.4	54.1	43.4	42.2	56.2	63.6	45.3	43.5	45.3	31.3	32.2	45.1
Shan *et al*. [38] (F=243)	MM2021	40.8	44.5	41.4	42.7	46.3	55.6	41.8	41.9	53.7	60.8	45.0	41.5	44.8	30.8	31.9	44.3
MixSTE [48] (F=243)	CVPR2022	37.6	40.9	37.3	39.7	42.3	49.9	40.1	39.8	51.7	55.0	**42.1**	39.8	41.0	27.9	27.9	40.9
P-STMO [37] (F=243)	ECCV2022	38.9	42.7	40.4	41.1	45.6	49.7	40.9	39.9	55.5	59.4	44.9	42.2	42.7	29.4	29.4	42.8
Diff3DHPE-M (F=243)		**37.3**	**40.6**	**36.3**	**38.0**	**41.8**	**46.6**	**38.3**	**39.4**	**51.3**	**53.0**	**42.1**	**39.5**	**40.7**	**27.3**	**28.2**	**40.0**
P-MPJPE↓		Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Chen *et al*. [4] (F=81)	TCSVT2021	33.1	35.3	33.4	35.9	36.1	41.7	32.8	33.3	42.6	49.4	37.0	32.7	36.5	25.5	27.9	35.6
Poseformer [51] (F=81)	ICCV2021	34.1	36.1	34.4	37.2	36.4	42.2	34.4	33.6	45.0	52.5	37.4	33.8	37.8	25.6	27.3	36.5
MixSTE [48] (F=81)	CVPR2022	32.0	34.2	31.7	33.7	34.4	39.2	32.0	31.8	42.9	46.9	35.5	32.0	34.4	23.6	25.2	33.9
Diff3DHPE-M (F=81)		**31.1**	**32.9**	**30.5**	**32.5**	**32.7**	**38.4**	**30.7**	**30.0**	**41.3**	**43.4**	**35.2**	**31.2**	**33.2**	**23.1**	**24.1**	**32.7**
Liu *et al*. [24] (F=243)	CVPR2020	32.3	35.2	33.3	35.8	35.9	41.5	33.2	32.7	44.6	50.9	37.0	32.4	37.0	25.2	27.2	35.6
Shan *et al*. [38] (F=243)	MM2021	32.5	36.2	33.2	35.3	35.6	42.1	32.6	31.9	42.6	47.9	36.6	32.1	34.8	24.2	25.8	35.0
MixSTE [48] (F=243)	CVPR2022	30.8	33.1	30.3	31.8	33.1	39.1	31.1	30.5	42.5	44.5	34.0	30.8	32.7	22.1	**22.9**	32.6
P-STMO [37] (F=243)	ECCV2022	31.3	35.2	32.9	33.9	35.4	39.3	32.5	31.5	44.6	48.2	36.3	32.9	34.4	23.8	23.9	34.4
Diff3DHPE-M (F=243)		**29.8**	**33.0**	**29.2**	**30.8**	**32.0**	**36.7**	**29.5**	**30.0**	**40.4**	**41.6**	**33.8**	**30.6**	**32.2**	**22.0**	23.1	**31.6**

Table 2. Results on Human3.6M with ground truth 2D keypoints. The best and second-best results of the same number of frames setting are in **Bold** and underlined, respectively. ↓: lower is better. Diff3DHPE-M: Diff3DHPE with MixSTE backbone.

MPJPE↓		Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Poseformer [51] (F=81)	ICCV2021	30.0	33.6	29.9	31.0	30.2	33.3	34.8	31.4	37.8	38.6	31.7	31.5	29.0	23.3	23.1	31.3
MixSTE [48] (F=81)	CVPR2022	25.6	27.8	24.5	25.7	**24.9**	29.9	28.6	27.4	29.9	29.0	26.1	25.0	25.2	18.7	**19.9**	25.9
Diff3DHPE-M (F=81)		**25.5**	**24.8**	**23.6**	**22.4**	25.4	**26.2**	**25.9**	**24.5**	**27.7**	**27.1**	**25.2**	**22.7**	**23.9**	**18.6**	**19.9**	**24.2**
SRNet [47] (F=243)	ECCV2020	34.8	32.1	28.5	30.7	31.4	36.9	35.6	30.5	38.9	40.5	32.5	31.0	29.9	22.5	24.5	32.0
Liu *et al*. [24] (F=243)	CVPR2020	34.5	37.1	33.6	34.2	32.9	37.1	39.6	35.8	40.7	41.4	33.0	33.8	33.0	26.6	26.9	34.7
Shan *et al*. [38] (F=243)	MM2021	29.5	30.8	28.8	29.1	30.7	35.2	31.7	27.8	34.5	36.0	30.3	29.4	28.9	24.1	24.7	30.1
MixSTE [48] (F=243)	CVPR2022	21.6	22.0	20.4	21.0	20.8	24.3	24.7	21.9	26.9	24.9	21.2	21.5	20.8	14.7	**15.7**	21.6
P-STMO [37] (F=243)	ECCV2022	28.5	30.1	28.6	27.9	29.8	33.2	31.3	27.8	36.0	37.4	29.7	29.5	28.1	21.0	21.0	29.3
Diff3DHPE-M (F=243)		**20.4**	**20.8**	**19.4**	**19.3**	**20.5**	**21.6**	**21.6**	**20.9**	**24.2**	**23.3**	**21.1**	**19.3**	**19.4**	**14.5**	16.1	**20.2**

Table 3. Results on MPI-INF-3DHP. The best and second-best results are in **Bold** and underlined, respectively. ↑: higher is better. ↓: lower is better. Diff3DHPE-M: Diff3DHPE with MixSTE backbone.

Method		PCK↑	AUC↑	MPJPE↓
PoseFormer [51] (F=9)	ICCV2021	88.6	56.4	77.1
MixSTE [48] (F=27)	CVPR2022	94.4	66.5	54.9
Chen *et al*. [4] (F=81)	TCSVT2021	87.9	54.0	78.8
P-STMO [37] (F=81)	ECCV2022	97.9	75.8	32.2
Hu *et al*. [15] (F=96)	MM2021	97.9	69.5	42.5
Diff3DHPE-M (F=27)		**99.1**	**84.8**	**19.6**

4.4.2 Effect of Noisy 2D Keypoints

To evaluate the robustness of Diff3DHPE under more challenging conditions, we design two types of artificial noise. The first type of noise is Gaussian noise with 0 mean and standard deviation $\sigma = 0.005, 0.01, 0.05, 0.1$, and 0.5. This noise is used to simulate inaccurate 2D pose detectors. The second type of noise is randomly setting 2D keypoints to $(0, 0)$ with probabilities of $dr_j = 0, 0.1, 0.2, 0.4$, and 0.8. This random drop is used to simulate occlusions in the input. We train models with CPN input on Human3.6M and separately added the two types of noise to the CPN input during the test stage. To ensure a fair comparison, We train models with the normalized CPN input on Human3.6M and separately add the two types of noise to the normalized CPN input during the test stage. To ensure a fair comparison, we train the baseline MixSTE with only L2 loss of 3D pose prediction error and normalize the training target 3D pose ground truth to $[-1, 1]$, the same as our Diff3DHPE models. Additionally, we keep the same architecture for the baselines as their corresponding backbone models used in Diff3DHPE to reduce the effect of using different loss functions, including temporal consistency loss [48].

Fig. 6 illustrates that the performances of all models drop dramatically as the magnitude of the noise increases. However, our Diff3DHPE can still effectively overcome the impact of large-magnitude noise in most scenarios compared to the baseline model.

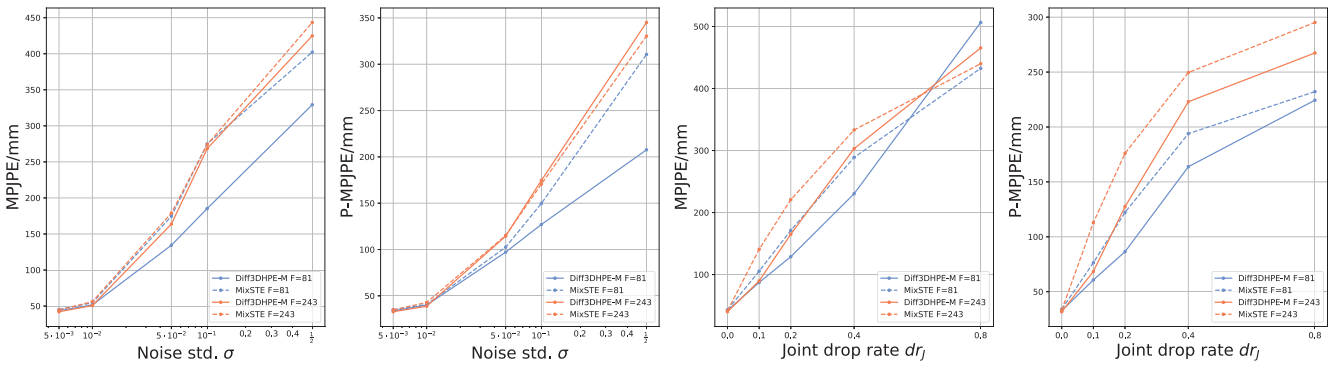

Figure 6. Performance changes when Diff3DHPE-M and the baseline model are tested with CPN input distorted by two types of artificial noise. 1. Gaussian noise with 0 mean and std σ. 2. Randomly set 2D keypoints to $(0, 0)$ with probability dr_j.

4.4.3 Effect of Diff3DHPE in seq2seq and seq2frame Styles

To demonstrate the versatility of Diff3DHPE, we apply it to both *seq2seq* and *seq2fame* settings using MixSTE and PoseFormer as backbone models for Diff3DHPE-M and Diff3DHPE-P, respectively. It is worth noting that in Diff3DHPE-P, we only initialize Gaussian noise for the 3D pose of the central frame. This noisy 3D pose is then duplicated and concatenated with each 2D pose along the channel dimension in the input sequence. We use the same training protocol for baseline models described in Section 4.4.2.

Table 4 demonstrates that the Diff3DHPE models outperform their corresponding baselines by up to 12.7% ($15.1mm$ vs. $17.3mm$). These results empirically demonstrate the capability of Diff3DHPE to collaborate with other GNN-based models in both *seq2seq* and *seq2frame* prediction methods.

We also conduct training on DDIM-M models. These models utilize the Diff3DHPE with a MixSTE backbone and incorporate the original DDIM method. As Table 4 shows, DDIM-M achieves better P-MPHPE than the baseline but has downgraded MPJPE performance. Meanwhile, DDIM-M uses more than four times iterations compared to Diff3DHPE-M but fails to surpass its accuracy, which proves the efficiency of our alternative design of the diffusion model.

4.5. Effect of Random Seeds

To show the effect of random seeds, we train MixSTE and Diff3DHPE-M with four more random seeds. The means and standard deviations of experiments using five random seeds are listed in Table 4 with † mark. We conduct the two-sample t-test based on the results. The p-values of $F = 81$ and $F = 243$ are $8.1e^{-4}$ and $2.5e^{-5}$, respectively. All p-values are smaller than 0.05, which indicates the gain from Diff3DHPE is significant.

Table 4. Ablation study of seq2seq and seq2frame models on Human3.6M. The best results are in **Bold**. ↓: lower is better. DDIM-M: Diff3DHPE with MixSTE backbone and using original DDIM method. Diff3DHPE-M: Diff3DHPE with MixSTE backbone. Diff3DHPE-P: Diff3DHPE with PoseFormer backbone. S: the number of reverse diffusion steps. $s.d$: standard deviation. $*$: we train the baselines with only L2 loss of 3D pose prediction error and normalize the training target 3D pose ground truth to $[-1, 1]$.†: we train the models with five different random seeds.

Model	CPN F=81			GT F=81		
	S	MPJPE±s.d.↓	P-MPJPE±s.d.↓	S	MPJPE↓	P-MPJPE↓
MixSTE*	N/A	43.2	34.0	N/A	26.4	20.2
DDIM-M	40	44.2	33.4	-	-	-
Diff3DHPE-M	**9**	**42.0**	**32.7**	5	**24.2**	**18.5**
MixSTE*†	N/A	43.6±0.3	34.1±0.1	-	-	-
Diff3DHPE-M†	9	**41.9±0.6**	**32.6±0.3**	-	-	-
Model	CPN F=243			GT F=243		
	S	MPJPE±s.d.↓	P-MPJPE±s.d.↓	S	MPJPE↓	P-MPJPE↓
MixSTE*	N/A	41.7	33.1	N/A	22.4	17.3
DDIM-M	80	42.2	32.3	-	-	-
Diff3DHPE-M	**5**	**40.0**	**31.6**	6	**20.2**	**15.1**
MixSTE*†	N/A	41.9±0.3	33.3±0.2	-	-	-
Diff3DHPE-M†	5	**40.1±0.4**	**31.6±0.2**	-	-	-
Model	CPN F=81			GT F=81		
	S	MPJPE↓	P-MPJPE↓	S	MPJPE↓	P-MPJPE↓
PoseFormer*	N/A	46.1	36.2	N/A	34.9	26.6
Diff3DHPE-P	5	**45.3**	**35.3**	5	**31.9**	**23.4**

5. Conclusion

This paper introduces a novel approach for the 3D human pose estimation task named Diff3DHPE, which is based on a Transformer-based diffusion model. The proposed model is designed to reduce the number of iteration steps without compromising performance. Additionally, the over-smoothing issue in Transformer is addressed by incorporating a PDE-based graph diffusion design. The experimental results demonstrate the effectiveness of Diff3DHPE and show that it can be combined with various GNN designs in both *seq2seq* and *seq2frame* settings. Overall, Diff3DHPE provides a promising solution to the challenging task of 3D human pose estimation.

Acknowledgement. TZ was supported in part by the Swiss National Science Foundation via the Sinergia grant CRSII5-180359.

References

[1] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2272–2281, 2019. 1

[2] Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In *International Conference on Machine Learning*, pages 1407–1418. PMLR, 2021. 2, 3, 5

[3] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation = 2d pose estimation + matching. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5759–5767, 2017. 1, 2

[4] Tianlang Chen, Chen Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. Anatomy-aware 3d human pose estimation with bone-based pose decomposition. *IEEE Trans. Cir. and Sys. for Video Technol.*, 32(1):198–209, 2022. 6, 7

[5] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 6

[6] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *Computer Vision – ECCV 2020*, pages 769–787. Springer International Publishing, 2020. 1

[7] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2262–2271, 2019. 1, 2

[8] Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. Learning 3d human pose from structure and motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1, 2

[9] Moritz Einfalt, Katja Ludwig, and Rainer Lienhart. Uplift and upsample: Efficient 3d human pose estimation with uplifting transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2903–2913, January 2023. 1

[10] Mark I Freidlin and Alexander D Wentzell. Diffusion processes on graphs and the averaging principle. *The Annals of probability*, pages 2215–2245, 1993. 3

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014. 3

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 3

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997. 2

[14] Mir Rayat Imtiaz Hossain and James J. Little. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2

[15] Wenbo Hu, Changgong Zhang, Fangneng Zhan, Lei Zhang, and Tien-Tsin Wong. Conditional directed graph convolution for 3d human pose estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 602–611. Association for Computing Machinery, 2021. 7

[16] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. 2, 3, 5

[17] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022. 3

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] Chen Li and Gim Hee Lee. Generating multiple hypotheses for 3d human pose estimation with mixture density network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[20] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian conference on computer vision*, pages 332–347, 2015. 2

[21] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13147–13156, June 2022. 1

[22] Mude Lin, Liang Lin, Xiaodan Liang, Keze Wang, and Hui Cheng. Recurrent 3d pose sequence machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[23] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. A comprehensive study of weight sharing in graph networks for 3d human pose estimation. In *Computer Vision – ECCV 2020*, pages 318–334, 2020. 1

[24] Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, and Vijayan Asari. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 6, 7

[25] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1

[26] Naoki Masuda, Mason A Porter, and Renaud Lambiotte. Random walks and diffusion on networks. *Physics reports*, 716:1–58, 2017. 3

[27] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian

Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017. 2, 5

[28] Francesc Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2

[29] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 4

[30] Hoang Nt and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019. 3

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 6

[32] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[33] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael 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 (CVPR)*, June 2019. 1, 2, 6

[34] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *International Conference on Learning Representations*, 2023. 3

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022. 3

[36] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–14, 2022. 3

[37] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation. *arXiv preprint arXiv:2203.07628*, 2022. 1, 6, 7

[38] Wenkang Shan, Haopeng Lu, Shanshe Wang, Xinfeng Zhang, and Wen Gao. Improving robustness and accuracy via relative information encoding in 3d human pose estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 3446–3454. Association for Computing Machinery, 2021. 7

[39] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1, 3, 4

[40] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 3, 4

[41] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 1, 3

[43] Jingbo Wang, Sijie Yan, Yuanjun Xiong, and Dahua Lin. Motion guided 3d pose estimation from videos. In *Computer Vision – ECCV 2020*, pages 764–780, 2020. 1, 2

[44] Lele Wu, Zhenbo Yu, Yijiang Liu, and Qingshan Liu. Limb pose aware networks for monocular 3d pose estimation. *IEEE Transactions on Image Processing*, 31:906–917, 2022. 1

[45] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*, 2019. 2

[46] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018. 2

[47] Ailing Zeng, Xiao Sun, Fuyang Huang, Minhao Liu, Qiang Xu, and Stephen Lin. Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. In *Computer Vision – ECCV 2020*, pages 507–523. Springer International Publishing, 2020. 7

[48] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13232–13242, June 2022. 1, 2, 3, 6, 7

[49] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[50] Ce Zheng, Wenhan Wu, Taojiannan Yang, Sijie Zhu, Chen Chen, Ruixu Liu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *ArXiv*, abs/2012.13392, 2019. 1, 6

[51] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11656–11665, October 2021. 1, 2, 3, 6, 7

[52] Kun Zhou, Xiaoguang Han, Nianjuan Jiang, Kui Jia, and Jiangbo Lu. Hemlets pose: Learning part-centric heatmap triplets for accurate 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 2

[53] Zhiming Zou and Wei Tang. Modulated graph convolutional network for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11477–11487, October 2021. 2