3D Pose Estimates and Diffusion Method

Aolong Li [1 *] **Wenda Li** [1 *] **Tianyu Sun** [1 *]

Abstract

The adoption of diffusion models in generative AI has gained popularity. This paper delves into a pose estimation framework aimed at enhancing probabilistic 3D human pose estimation using an improved diffusion model. Our approach involves leveraging information from 2D key points and systematically introducing noise to each key point. Subsequently, we employ a trained diffusion model to denoise the image. The resulting denoiser is then utilized for making inferences and generating 3D poses. Several methods are proposed to enhance the propagation of the diffusion model, building upon previous efforts to achieve superior results. Code is available at https://github.com/local-ring/E533-3D-Pose-Estimates-and-Diffusion-Method.

1. Introduction

In the realm of computer vision, predicting the three-dimensional coordinates of human joints from images or video content, known as 3D human pose estimation, has garnered significant attention. This technology is pivotal for applications in augmented reality, self-driving, metaverse, etc. Traditionally, this estimation process involves two steps: detecting 2D poses with a 2D pose detector and then converting these into 3D poses, a process termed as 2D-to-3D lifting. However, the challenge of accurately predicting 3D poses from single-camera imagery persists, mainly due to depth ambiguity and potential occlusions leading to high levels of uncertainty.

Emerging as a powerful tool in image generation, diffusion models demonstrate an ability to generate samples from random noise by progressively reducing noise. This technique of gradual noise reduction simplifies the bridging of significant gaps between the highly noisy distribution and a certain distribution, guiding models to smoothly create samples matching the desired data distribution.

The 3D pose prediction model draws inspiration from the robust capabilities of diffusion models, particularly in handling high uncertainty scenarios, to address the challenges in 3D pose estimation.

We can conceptualize the task of estimating 3D poses as a reverse diffusion process, progressively refining a distribution of uncertain 3D poses distribution towards a distribution with lower uncertainty. The reverse diffusion process takes an initial 2D pose, which is indeterminate in 3D space, aiming to reduce indeterminacy and yield a distribution of high-quality 3D pose solutions.

The framework comprises two contrasting processes: a forward process creating supervisory signals for model training and a reverse process for both training and estimation. In the forward process, we gradually diffuse a ground truth 3D pose distribution x_0 with low indeterminacy towards a high-indeterminacy distribution, which mirrors the inherent uncertainty of 3D poses distribution x_T. We acquire samples from each intermediate stage of the distribution throughout the forward diffusion process, served as the supervisory signal, enabling the model to smoothly transition from the uncertain and variable distribution to a state of higher accuracy and determinacy in the 3D pose estimation.

The reverse process then employs a diffusion model to transform this uncertain distribution into accurate 3D pose predictions. However, challenges arise due to the starting point of the reverse diffusion process being an estimated 2D pose with inherent 3D uncertainty, unlike the random noise starting point in the well-established image generation models. The divergence stems from the variability in the underlying uncertainty distribution of each 3D pose, prohibiting a uniform approach to converge the output of the forward diffusion steps to a single consistent Gaussian noise, as seen in prior works. Additionally, the complexity and irregularity of the uncertainty distribution in 3D poses defy characterization by a single Gaussian model. Another significant hurdle is performing precise 3D pose estimation using just the x_T distribution as input. The objective here transcends the generation of any plausible 3D pose; the aim is to predict accurate 3D poses that align with the estimated 2D poses, necessitating additional contextual information for accuracy.

To surmount these challenges, firstly, the indeterminate

[*]Equal contribution [1]Department of Mathematics, Indiana University, Bloomington, Indiana. Correspondence to: Aolong Li (aolli@iu.edu), Wenda Li (wli8@iu.edu), Tianyu Sun (ts19@iu.edu).

3D pose distribution, x_T, is initialized based on extracted heatmaps (in the preprocessed data), capturing the essence of the desired 3D pose's underlying uncertainty. Secondly, during the forward diffusion phase, the ground truth 3D pose distribution, x_0, is subjected to added noise modeled by a Gaussian Mixture Model (GMM), which accurately represents the uncertainty distribution x_T. This process gradually shapes the indeterminate 3D pose distributions to mirror x_T after T-step iterations. Thirdly, the reverse diffusion process is meticulously conditioned on contextual information extracted from the input video or frame. This approach aims to exploit the spatial-temporal dynamics between frames and joints. To effectively utilize this context information and achieve progressive denoising for accurate 3D pose predictions, we have designed a specialized Graph Convolutional Network (GCN)-based diffusion model.

2. Related Work

2.1. Diffusion Model

Diffusion models form a family of generative models that iteratively add noise to the observed data and subsequently reconstruct the original data by reversing this process. Unlike VAEs or GANs, diffusion models emphasize a reversible generative process. This property facilitates effective denoising, making them well-suited for handling noisy data and producing high-quality reconstructions. The diffusion models first become noticed when the DDPM(Ho et al., 2020) is released. Then DDIM(Song et al., 2022) is introduced to improve the image generation speed. And (Dhariwal & Nichol, 2021) is introduced to incorporate class information to each step to improve the training result. Later on, EDM(Karras et al., 2022) is introduced to use a second-order sampling method to improve the resulting quality.

2.2. 3D Human Pose Estimation

Our task mainly focuses on predicting 3D human pose from 2D key joint locations. The most popular dataset is the Human 3.6m dataset (Ionescu et al., 2014). Past works include CNN-based frameworks (Chen et al., 2021), transformer seq-to-seq frameworks(Zhang et al., 2022), and GCNs(Zhao et al., 2019). Notably, several works (Gong et al., 2023)(Shan et al., 2023) incorporate GCNs into the seq-to-seq framework and make hypotheses on several results to obtain the final predictions. In our papers, we proposed a method based on the Diffpose(Gong et al., 2023) but with better diffusion modeling. And we showed that our improvements could outperform the original result obtained by the original paper.

3. Background on Diffusion Models

(can be moved to the appendix part if necessary-AL) Diffusion models are a class of probabilistic generative models learning how to transform noise x_T to a sample x_0 by recurrently denoising. It consists of two processes, a diffusion (forward) process $x_0 \to x_1 \to \cdots \to x_T$ and a reverse process $x_T \to x_{T-1} \to \cdots \to x_0$.

3.1. Diffusion Process

The diffusion process q generates contaminated samples $x_1, x_2, \cdots, x_T$ by adding different levels of Gaussian noise to the original signal x_0 (the ground truth 3D pose distribution in our case) at each timestep $t \in [0, T]$ by

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$

where each conditional distribution is Gaussian, that is,

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$$

where β_t is the noise variance schedule and $\mathbf{I}$ denotes the identity matrix. If we do not care too much about the intermediate step, we can generate x_T from x_0 without iterations by:

$$\begin{aligned} q(x_t|x_0) &:= \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t\mathbf{I}) \\ &= \sqrt{\bar{\alpha}_t}x_0 + \epsilon\sqrt{1-\bar{\alpha}_t} \end{aligned}$$

where $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$ and $\alpha_t := 1 - \beta_t$. The second identity is for reparametrization trick and ϵ is the Gaussain noise, that is, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. Note that with reasonable choice of noise schedule β_t (fixed numbers between 0 and 1, or cosine noise variance schedule[1]), $\bar{\alpha}_t$ will converge to zero and therefore the distribution of $q(x_T)$ will be Gaussian distribution $(0, \mathbf{I})$, as the T is sufficiently large.

As mentioned in the introduction, the final distribution is in general not a standard Gaussian distribution $(0, \mathbf{I})$. We need to increase the number of random variables to the number of joints because the mean of locations of all joints cannot be in the origin at the same time. Moreover, we need to increase the number of kernels/components because the initial region of each joint for different pose can be different (for example, a standing person and a laying person). We can use K components of Gaussian distribution of $3J$ random variables where K is the hyperparameter we could tune, and J is number of joints and for each joint the coordinates consisting of three random variables (x, y, z).

We can use the EM algorithm to find the Gaussian Mixture Model parameters to fit the "ground truth" distribution x_T

[1] $\beta_t = 1 - \frac{\alpha_t}{\alpha_{t-1}}$ where $\alpha_t = \frac{f(t)}{f(0)}$ and $f(t) = \cos\left(\frac{t/T+s}{1+s}\frac{\pi}{2}\right)^2$, and we clip β_t between 0.01 and 0.99 to avoid singularity

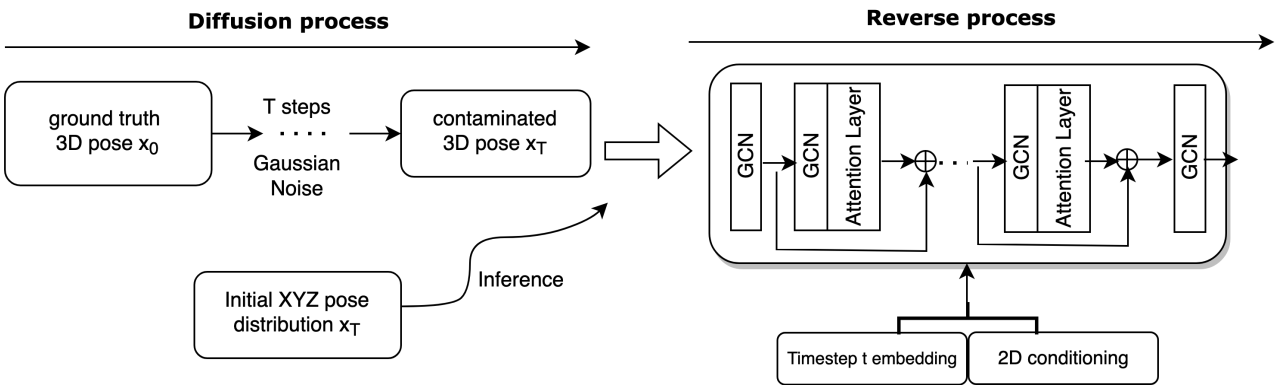


Figure 1. Overview of the proposed method. For reverse process, since the model will map distribution to distribution, in practice, we take N samples from the distribution x_T, and we denoise the input T steps recurrently. Finally, we take the mean of the output of n samples we feed to obtain the deterministic prediction.

which is generated from the heatmap. Suppose we find the GMM parameters, including the mean value $\mu_m \in \mathbf{R}^{3J}$, the covariance matrix $\Sigma_m \in \mathbf{R}^{3J\times 3J}$ and π_m the weight of the m-th Gaussian component, we can modify the above equation by

$$q(x_t|x_0) = \mu^G + \sqrt{\bar{\alpha}_t}(x_0 - \mu^G) + \epsilon^G\sqrt{1-\bar{\alpha}_t}$$

where μ^G is the random variable $\mu^G = \sum \mathbf{1}_m\mu_m$, ϵ^G is the random variable $\epsilon^G \sim \mathcal{N}(0, \sum(\mathbf{1}_m\Sigma_m))$ and $\mathbf{1}_m$ is the indicator variable for the m-th component. That is, in practice, to draw a sample for $q(x_t|x_0)$, first, we need to determine which $\mathbf{1}_m$ is taking value 1 with the given probability π_m, and then we can compute the sample values for the mean and variance. Last, we can draw sample of $q(x_t|x_0)$ with those values. Similar with the argument of basic diffusion process, we can show that the distribution will converge to the desired GMM distribution as $T \to \infty$.

3.2. Reverse Process g

We use a neural network parameterized by θ to implement the reverse process p. Like the diffusion process, we will iterate $p_\theta(x_{t-1}|x_t)$ step by step until we get our prediction of x_0. Indeed, by Bayes' law, we have

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t\mathbf{I})$$

where $\tilde{\mu}_t$ and $\tilde{\beta}_t$ are the variance and mean of the Gaussian distribution respectively. Note that $\tilde{\mu}_t$ depends on the x_0, and x_0 is the distribution we want to predict, so all we know is the distribution is a Gaussian and now we need to approximate $q(x_{t-1}|x_t, x_0)$ by the neural network

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \tilde{\mu}_\theta(x_t, t), \tilde{\beta}_t\mathbf{I})$$

where

$$\tilde{\mu}_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$$

where ϵ_θ is the noise predicted by the neural network, supervised by the loss $\mathcal{L} = \mathbf{E}_{x_t,t,\epsilon}[||\epsilon_\theta(x_t, t) - \epsilon||^2]$ where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

3.3. Ecidating the Design Space (EDM)

Usually, the reverse process can be done using a DDPM/DDIM architecture. Here, we adopted the idea from Elucidating the Design Space (EDM)(Karras et al., 2022) to implement a second-order Runge Kutta method to solve the ODE and obtain x at the next time step. The idea is that we first nudge t_i by some γ_i to obtain a new time $\hat{t}_i = t_i + \gamma_i t_i$. Then we inject noise at $\hat{t}_i$ level to x_i to obtain $\hat{x}_i$. Then we solve the ODE of the diffusion process to obtain x_{i+1} from $\hat{x}_i$. Then we add a second-order correction term to reduce the error term of the diffusion process.

3.4. Network Architecture

In summary, our diffusion model mainly consists of two passes: forward diffusion that diffuse the images to the distribution of 3D joints, Φ_T and backward reverse process that denoise the noisy 2D joints samples from Φ_T, Φ_{T-1}, ..., to Φ_0, our prediction.

3.4.1. Reverse Process g

To encode the skeleton joints informations, especially with the spatial relatioships between joints, we model the human joints as a graph and choose graphical convotional neural network(GCN) and its modifications to transformer (Zhao et al., 2022). For first and final layer, we used GCN layers to do embeddings.

The combination includes GCN layers with self-attention layers. To take advantage of the particular simple graph structure of human joints, we used 4 GCN attention blocks(2

GCN layer followed by a self-attention) to encode the joints information: first block to describe the coupling of two joints, second block to describe the global coupling of joints in a limb, third block to describe the coupling of limbs, and final block plus the residual connection to connect the local information to the global body joints.

4. Experiments

4.1. Datasets and Evaluation Metrics

We train and test the models on the widely used dataset for 3D human pose estimation: Human3.6M (Ionescu et al., 2014). It consists of 3.6 million 3D human poses and corresponding images of 17 scenarios (discussion, smoking, taking photo, etc.) performed by 11 professional actors (6 male, 5 female). Especially, we used the GMM format data preprocessed by previous work (Gong et al., 2023). We train on five subjects (S1, S5, S6, S7, S8) and test on (S9, S11) subjects. We report the mean per joint position error (MPJPE) and Procrustes MPJPE (P-MPJPE). The former computes the Euclidean distance between the predicted joint positions and the ground truth positions. The latter is the MPJPE after the predicted results are aligned to the ground truth via a rigid transformation.

4.2. Implementation Details

We set the number of pose sample N from distributions as 5, and the number of reverse diffusion steps as 50. For forward diffusion, we fit the corrupted distribution via GMM model with 5 kernels ($M = 5$) of 17 random variables ($K = 17$), which is equal to the number of key joints.

4.2.1. Tensor Shapes

Through GCN embedding, the input x_t of shape $[:, J, 3]$ is embedded to a $i(x_t) = [:, J, 128]$, and then to encode timestep t, we used sinusoidal embedding (Vaswani et al., 2017) to get $i(t)$ of shape $[:, J, 128]$, then we add them $e(x_t) = i(t) + i(x_t)$, as in (Vaswani et al., 2017). Then $e(x_t)$ is fed to GCN attention blocks and then mapped by the final GCN layer to get an output x_{t-1}. We iterate this T steps to get the prediction x_0.

4.2.2. Loss Choice

From the input X, we first extract 2D heatmaps from pretrained network (Chen et al., 2017), then we compute the z-direction distribution with 3D joints data from 2D heatmaps. EM should do the trick of estimating the GMM parameters of $\hat{\Phi}_T$. Now we sample N(we chose 5) samples to get $(\hat{X}_1^k, \hat{X}_2^k, \cdots, \hat{X}_n^k)$, $k = 1, \cdots, N$. At last, we formulate the loss

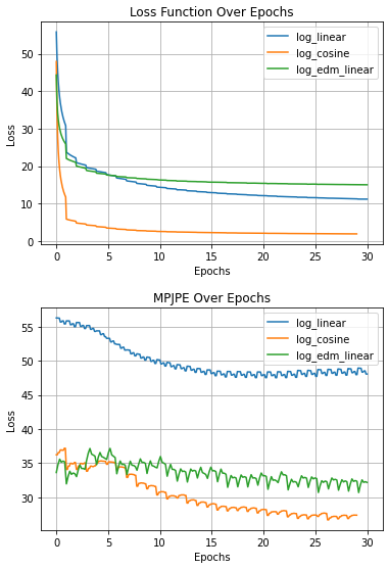
$$\mathcal{L} = \sum_{i=1}^{N} \sum_{t=1}^{T} MSE(g_{\theta,t}(X_t^i) - X_{t-1}^i)$$

4.3. Qualitative Results

Because of the enormous dataset and limited computing resources, we were unable to fully replicate the original research paper(Gong et al., 2023) to all the epochs. We stopped at 30 epochs and compared the result using linear beta scheduling, cosine beta scheduling, and EDM(Karras et al., 2022) to solve for the diffusion process. The result is shown from the table below.

Linear beta schedule	47.503
Cosine beta schedule	26.6827
EDM	30.6844

We also plotted the loss function and MPHPE over epochs to show how different schedules or diffusion methods affected loss function decay. The result is shown below.



We can observe that using a cosine beta schedule has a much higher impact than a linear beta schedule. And while using EDM in the diffusion process could largely reduce error and achieve better results, it is not as impactful as using a cosine beta schedule.

Acknowledgements

We appreciate the suggestions provided by Professor Kim.

Most of the utility functions to preprocess data are from (Diffpose). And the structure of the denoiser is based on (Diffpose). Our team improved the beta scheduling in the diffusion generative model by using a cosine schedule because we don't want to use information that is too noisy(D3DP). And we modified and implemented the idea from (EDM) instead of the traditional (DDIM) to generate steps in the diffusion model.

(Diffpose) https://github.com/GONGJIA0208/Diffpose

(EDM) https://github.com/NVlabs/edm/tree/main

(DDIM) https://github.com/ermongroup/ddim

(D3DP) https://github.com/paTRICK-swk/D3DP

References

Chen, T., Fang, C., Shen, X., Zhu, Y., Chen, Z., and Luo, J. Anatomy-aware 3d human pose estimation with bone-based pose decomposition, 2021.

Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., and Sun, J. Cascaded pyramid network for multi-person pose estimation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7103–7112, 2017. URL https://api.semanticscholar.org/CorpusID:4703058.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis, 2021.

Gong, J., Foo, L. G., Fan, Z., Ke, Q., Rahmani, H., and Liu, J. Diffpose: Toward more reliable 3d pose estimation, 2023.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models, 2020.

Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. 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, jul 2014.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models, 2022.

Shan, W., Liu, Z., Zhang, X., Wang, Z., Han, K., Wang, S., Ma, S., and Gao, W. Diffusion-based 3d human pose estimation with multi-hypothesis aggregation, 2023.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Zhang, J., Tu, Z., Yang, J., Chen, Y., and Yuan, J. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video, 2022.

Zhao, L., Peng, X., Tian, Y., Kapadia, M., and Metaxas, D. N. Semantic graph convolutional networks for 3d human pose regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2019. doi: 10.1109/cvpr.2019.00354. URL http://dx.doi.org/10.1109/CVPR.2019.00354.

Zhao, W., Wang, W., and Tian, Y. Graformer: Graph-oriented transformer for 3d pose estimation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20406–20415, 2022. doi: 10.1109/CVPR52688.2022.01979.