Distribution-Aligned Diffusion for Human Mesh Recovery

Lin Geng Foo[1] Jia Gong[1] Hossein Rahmani[2] Jun Liu[1†]
[1]Singapore University of Technology and Design [2]Lancaster University
{lingeng_foo,jia_gong}@mymail.sutd.edu.sg, h.rahmani@lancaster.ac.uk, jun_liu@sutd.edu.sg

Abstract

Recovering a 3D human mesh from a single RGB image is a challenging task due to depth ambiguity and self-occlusion, resulting in a high degree of uncertainty. Meanwhile, diffusion models have recently seen much success in generating high-quality outputs by progressively denoising noisy inputs. Inspired by their capability, we explore a diffusion-based approach for human mesh recovery, and propose a Human Mesh Diffusion (HMDiff) framework which frames mesh recovery as a reverse diffusion process. We also propose a Distribution Alignment Technique (DAT) that injects input-specific distribution information into the diffusion process, and provides useful prior knowledge to simplify the mesh recovery task. Our method achieves state-of-the-art performance on three widely used datasets. Project page: https://gongjia0208.github.io/HMDiff/.

1. Introduction

Monocular 3D human mesh recovery, where the 3D mesh vertex locations of a human are predicted from a single RGB image, is an important task with applications across virtual reality [17], sports motion analysis [1], and healthcare [54]. The field has received a lot of attention in recent years [6, 7, 32, 33, 8, 22], which has led to significant progress, but monocular 3D human mesh estimation still remains very challenging. Human body shapes are not only complex and contain many fine details, but also inherently exhibit *depth ambiguity* (when recovering 3D information from single 2D images) and *self-occlusion* (where body parts can be occluded by other body parts) [25, 6, 53]. In particular, the depth ambiguity and self-occlusion in this task often bring much uncertainty to the recovery of 3D mesh vertices, and places a huge burden on the model to handle this inherent uncertainty [30, 39, 8, 22].

At the same time, denoising diffusion probabilistic models (*diffusion models*) [16, 49] have recently seen much success in generative tasks, such as image [36], video [48] and text [31] generation, where they have been capable of producing highly realistic and good-quality samples. Specifically, diffusion models [16, 49] progressively "denoise" a noisy input – which is uncertain – into a high-quality output from the desired data distribution (e.g., natural images), through estimating the gradients of the data distribution [52] (also known as the score function). This progressive denoising helps break down the large gap between distributions (i.e., from a highly uncertain and noisy distribution to a desired target distribution), into smaller intermediate steps [52], which assists the model in converging towards generating the target data distribution smoothly. This gives diffusion models a strong ability to recover high-quality outputs from uncertain and noisy input data.

For the monocular 3D mesh recovery task, we also seek to recover a high-quality mesh prediction from uncertain and noisy input data, and so we leverage diffusion models to effectively tackle this task. To this end, we propose a novel diffusion-based framework for monocular 3D human mesh recovery, called Human Mesh Diffusion (**HMDiff**), where we frame the mesh recovery task as a *reverse diffusion process* which recovers a high-quality mesh by progressively denoising noisy and uncertain inputs.

Intuitively, in our HMDiff's approach, we can regard the mesh vertices as particles in the context of thermodynamics. At the start, the particles (representing the ground truth mesh vertices) are systematically arranged to form a high-quality human mesh, then these particles stochastically disperse throughout the space and eventually degrade into noise, leading to high uncertainty. This process (i.e., particles becoming more dispersed and noisy) is the *forward diffusion process*. Conversely, for human mesh recovery, we aim to perform the opposite of this process, i.e., the *reverse diffusion process*. Starting from a noisy and uncertain input distribution, we want to progressively denoise and reduce the uncertainty of the input to obtain a target human mesh distribution containing high-quality samples.

Correspondingly, our HMDiff framework consists of both the forward process and the reverse process. Specifically, the forward process is performed during training to generate samples of intermediate distributions that are used

† Corresponding author

as step-by-step supervisory signals to train our diffusion model g. On the other hand, the reverse process is a crucial part of our mesh recovery pipeline, which is used during both training and testing. In the reverse process, we first initialize a noisy distribution H_K, and use our diffusion model g to progressively transform H_K into a high-quality human mesh distribution (H_0) over K diffusion steps.

However, we face challenges in adopting a diffusion-based approach to tackle human mesh recovery. Firstly, it is difficult to directly produce complicated 3D mesh outputs with a single RGB image as input; as shown in previous works [7, 58, 42, 29, 8, 22, 41, 59], it is important to leverage some prior knowledge (e.g., pose information, segmentation maps) as input to guide the mesh recovery process, which is not performed in the standard diffusion process. Secondly, it is difficult to predict an accurate mesh output that corresponds to the RGB image, using only the standard diffusion process [16]. This is because we aim to predict a mesh that exactly corresponds to the input (i.e., it is an *input-specific* prediction), but the reverse process starts from the Gaussian distribution (H_K), which is *input-agnostic* and does not contain helpful input-specific distribution information. To tackle the above issues, we can extract useful *input-specific distribution information* to guide the reverse diffusion process, e.g., we can potentially follow some existing works [22, 8] to extract a pose heatmap from the image (which encodes rich semantic and uncertainty information [37, 28, 14]) and use it to initialize the input (H_K) for our diffusion process.

Using an input-specific distribution (e.g., a pose heatmap) to initialize the input (H_K) to the diffusion process has two benefits. Firstly, it will provide input-specific information to the diffusion process, which helps in generating an accurate mesh prediction (H_0) that corresponds to the input. Secondly, using an input-specific distribution (e.g., a pose distribution extracted from a pose estimator) allows us to effectively leverage prior knowledge [7, 58, 42, 29, 8, 22, 41, 59], which makes the mesh recovery task easier. Due to these reasons, we ideally want an input-specific distribution as input (H_K) to our reverse diffusion process. However, it is not feasible to directly initialize the starting distribution (H_K) with an input-specific distribution, because the standard reverse diffusion process is *theoretically formulated* and *constrained* to start from the (input-agnostic) Gaussian distribution.

Hence, we further propose a Distribution Alignment Technique (**DAT**) to inject input-specific distribution information to the diffusion process, which *narrows down the target space for the diffusion process* towards the specific mesh distribution that corresponds to the input image. Specifically, we initialize an input-specific distribution from the input image via a pose estimator [57], and use it to guide the initial diffusion steps towards the diffusion target H_0. This allows us to start the reverse process from Gaussian noise, while infusing the diffusion process with input-specific information, leading to faster convergence (i.e., fewer diffusion steps) and better performance.

2. Related Work

Human Mesh Recovery (HMR) aims to recover the 3D human mesh from a given input. Traditionally, HMR has been successful with the aid of information from depth sensors [46, 40] and inertial measurement units [18, 56]. Recently, there has been much research attention on the monocular setting [53], which is more convenient and widely applicable. Yet, monocular HMR is very challenging due to depth ambiguity, occlusions and complex pose variations [32, 53, 4]. Many works [20, 42, 41, 25, 13, 47, 4] tackle monocular HMR via a *model-based* approach, where networks are trained to regress the parameters of a human parametric model (e.g., SMPL [35], SCAPE [3]). To further improve performance, some works propose to leverage various forms of prior knowledge as guidance, including pose [7, 58], pose heatmaps [8, 22, 29], or segmentation maps [41, 59]. Recently, some works adopt a *model-free* approach [7, 32, 33, 27, 39, 6], where the full 3D human body shape is directly predicted. In this line of work, several models have been explored, including Convolutional Neural Networks (CNNs) [39], Graph Convolutional Networks (GCNs) [27, 7] and Transformers [32, 33, 6]. Differently, here we explore a diffusion-based framework to handle the uncertainty in HMR. To the best of our knowledge, this is the first work to use diffusion models to tackle monocular HMR. Our HMDiff framework effectively recovers human 3D mesh and achieves state-of-the-art performance.

Denoising Diffusion Probabilistic Models (Diffusion Models) [49, 16] effectively enable us to learn to sample from a desired data distribution, by iteratively "denoising" random noise into a high-quality sample from the desired data distribution through estimating the gradients (i.e., score function) of the data distribution [52, 9]. Diffusion models have been effective at image generation [16, 50], and have been explored for various other generation tasks such as video generation [48], and text generation [31]. Recently, several works also explore applying diffusion models in prediction tasks [10, 12, 43] and image-based inverse problems [9, 21, 51]. In contrast to these studies, monocular HMR presents a more difficult challenge, requiring a dense (mesh) output with only a single input image. Thus, to simplify the task of monocular HMR, we take inspiration from previous works [9, 21, 51] that estimate the posterior, and adopt a similar approach to guide the initial stages of the diffusion process with our DAT. Our DAT aligns the initial mesh distribution towards an extracted input-specific pose distribution (bridged by a mesh-to-pose function), resulting in faster convergence and better performance.

3. Background on Diffusion Models

Overall, diffusion models [16, 50] are probabilistic generative models that learn to transform random noise $h_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ into a desired sample h_0 by denoising h_K in a recurrent manner. Diffusion models have two opposite processes: the *forward process* and the *reverse process*.

Specifically, in the forward process, a "ground truth" sample h_0 with low uncertainty is gradually diffused over K steps ($h_0 \rightarrow h_1 \rightarrow ... \rightarrow h_K$) towards becoming a sample h_K with high uncertainty. Samples are obtained from the intermediate steps along the way, which are used during training as the step-by-step supervisory signals for the diffusion model g. To start the reverse process, noisy and uncertain samples h_K are first initialized according to a standard Gaussian distribution. Next, the diffusion model g is used in the reverse process ($h_K \rightarrow h_{K-1} \rightarrow ... \rightarrow h_0$) to progressively reduce the uncertainty of h_K and transform h_K into a sample with low uncertainty (h_0). The diffusion model g is optimized using the samples from intermediate steps (generated in the forward process), which guide it to smoothly transform the noisy and uncertain samples h_K into high-quality samples h_0. We go into more detail below.

Forward Process. Firstly, the forward diffusion process generates a set of intermediate noisy samples $\{h_k\}_{k=1}^{K-1}$ that will be used to aid the diffusion model in learning the reverse diffusion process during training. Since Gaussian noise is added between each step, we can formulate the posterior distribution $q(h_{1:K}|h_0)$ as:

$$q(h_{1:K}|h_0) := \prod_{k=1}^{K} q(h_k|h_{k-1}) \tag{1}$$

$$q(h_k|h_{k-1}) := \mathcal{N}_{pdf}\left(h_k|\sqrt{\frac{\alpha_k}{\alpha_{k-1}}}h_{k-1}, (1-\frac{\alpha_k}{\alpha_{k-1}})\mathbf{I}\right), \tag{2}$$

where $\mathcal{N}_{pdf}(h_k|\cdot)$ is the likelihood of sampling h_k conditioned on the given parameters, while $\alpha_{1:K} \in (0,1]^K$ is a fixed decreasing sequence that controls the noise scaling at each diffusion step. We can then formulate the posterior $q(h_k|h_0)$ for the diffusion process from h_0 to step k as:

$$\begin{aligned} q(h_k|h_0) :=& \int q(h_{1:k}|h_0)\mathrm{d}h_{1:k-1} \\ =& \mathcal{N}_{pdf}(h_k|\sqrt{\alpha_k}h_0, (1-\alpha_k)\mathbf{I}). \end{aligned} \tag{3}$$

Hence, we can express h_k as a linear combination of the source sample h_0 and random noise z, where each element of z is sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, as follows:

$$h_k = \sqrt{\alpha_k}h_0 + \sqrt{(1-\alpha_k)}z. \tag{4}$$

Thus, by setting a decreasing sequence $\alpha_{1:K}$ such that $\alpha_K \approx 0$, the distribution of h_K will converge to a standard Gaussian ($h_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$). Intuitively, this implies that the source signal h_0 will eventually be corrupted into Gaussian noise h_K, which matches with the non-equilibrium thermodynamics phenomenon of the diffusion process [49]. This facilitates the training of the reverse process, as the generated samples $\{h_k\}_{k=1}^{K}$ effectively bridge the gap between the standard Gaussian noise h_K and the source sample h_0.

Reverse Process. Next, to approximate the reverse diffusion process which transforms Gaussian noise h_K to a high-quality sample h_0, a diffusion model g (which is often a deep network) is optimized using the generated samples $\{h_k\}_{k=1}^{K}$ and the source sample h_0. The diffusion model g can be interpreted to be a score-based model [52] that estimates the *score function* $\nabla_h \log p(h)$ of the data distribution, where the iterative steps are performing denoising score matching [55] over multiple noise levels. The standard formulation of the reverse diffusion step by DDPM [16] can be formally expressed as:

$$h_{k-1} = \sqrt{\frac{\alpha_k}{\alpha_{k-1}}}\left(h_k - \frac{\alpha_{k-1}-\alpha_k}{\alpha_{k-1}\sqrt{1-\bar{\alpha}_k}}g(h_k, k)\right) + \sigma_k z_k, \tag{5}$$

where σ_k is a hyperparameter and z_k is Gaussian noise.

Therefore, during inference, Gaussian noise h_K can be sampled, and the reverse diffusion step introduced in Eq. 5 can be recurrently performed, allowing us to generate a high-quality sample h_0 with the trained diffusion model g.

4. Method

In this section, we first formulate our HMDiff framework (in Sec. 4.1). An overview of our framework is depicted in Fig. 1. Then, in Sec. 4.2, we propose DAT to inject input-specific distribution information into the diffusion process. Lastly, we design a diffusion network for HMR in Sec. 4.3 that can effectively model the relationship between vertices.

4.1. Human Mesh Diffusion (HMDiff)

Monocular 3D HMR is a very challenging task, due to the inherent depth ambiguity in recovering 3D information from single 2D images, as well as self-occlusion where some body parts may be occluded by other body parts. These issues often result in high uncertainty during 3D mesh recovery [25, 6, 53, 30]. Thus, in order to alleviate the uncertainty, we propose to leverage diffusion models, which have a strong capability in recovering high-quality outputs from noisy and uncertain data.

To this end, we propose HMDiff, a diffusion-based framework for HMR, which consists of a *forward process* and a *reverse process*. The forward process gradually adds noise and uncertainty to the ground truth mesh samples, eventually corrupting the mesh vertices into Gaussian noise. On the other hand, the reverse process learns to *reverse the effects of the forward process*, and learns to model the step-by-step reduction of noise and uncertainty, gaining the ability to progressively denoise the noisy inputs to recover a high-quality mesh. Specifically, we frame the HMR task as a reverse diffusion process, while the forward process plays a crucial role during training, as described below.

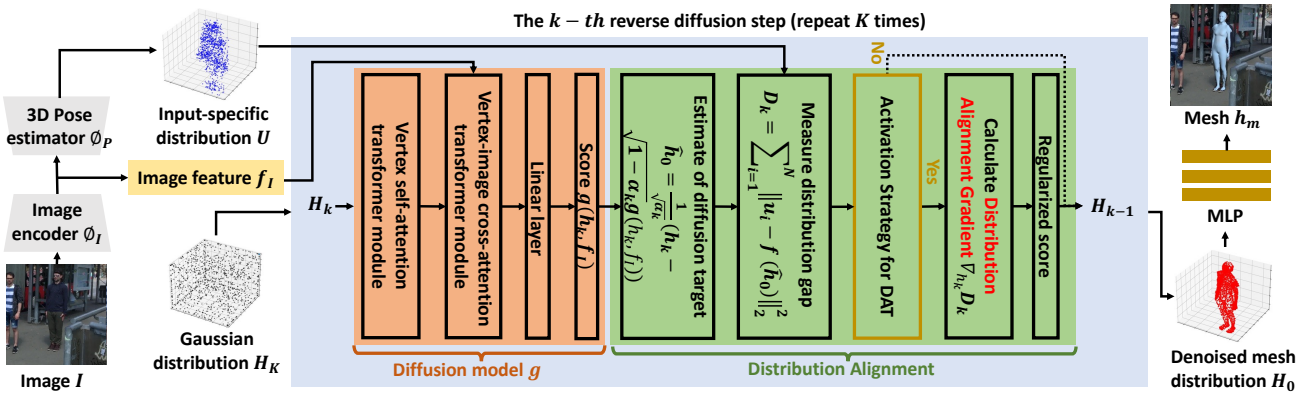

Figure 1. Illustration of the proposed Human Mesh Diffusion (HMDiff) framework with the Distribution Alignment Technique (DAT). Given an RGB image I, we first extract an image feature f_I and input-specific distribution U, by using pre-trained models ϕ_I and ϕ_P. Then, a Transformer-based diffusion model g takes in U, f_I and noise from a Gaussian distribution H_K, and iteratively performs K denoising diffusion steps to eventually obtain a denoised distribution H_0. We use our DAT technique to guide the diffusion process by computing a Distribution Alignment Gradient, which effectively injects input-specific information to guide the denoising step. Lastly, after obtaining the high-quality mesh distribution H_0, we take the center of H_0 and feed it into an MLP to obtain the final prediction h_m.

Forward Process. To train the diffusion model g to bridge the large gap between the target human mesh distribution H_0 and the standard Gaussian distribution H_K, we first generate a set of intermediate distributions $\{H_1, H_2, ..., H_{K-1}\}$ in the forward process. We obtain the intermediate distributions via Eq. 4, and use them as step-by-step supervisory signals to train the diffusion model g. Throughout the forward process, we apply noise to the vertex coordinates while keeping the topology between vertices fixed. To be more precise, we set the topology according to a predefined adjacency matrix (following [6, 7]) and keep it fixed.

Reverse Process. In the reverse process, we aim to recover an accurate and high-quality human mesh distribution H_0 from a noisy and uncertain input H_K (initialized as a standard Gaussian distribution) through a step-by-step process where the uncertainty is reduced. To achieve this, we design a diffusion model g, which predicts the score function in each k-th reverse diffusion step, conditioned on step index k. Then, we can use diffusion model g to perform reverse diffusion by recurrently taking reverse diffusion steps. Specifically, the reverse process consists of K steps that produce mesh distributions $p_k(h_k)$ at each k-th step, which are trained such that $p_k(h_k)$ match $q(h_k|\cdot)$ of the forward process (i.e., Eq. 2 and Eq. 3) at the k-th step. Overall, the reverse process is trained to produce $p_0(h_0)$ at step 0, which corresponds to the high quality mesh distribution H_0.

However, rather than taking the standard steps (Eq. 5) for reverse diffusion, here we instead use a different formulation [50] for each reverse diffusion step. This is because the reformulated reverse diffusion step (i.e., Eq. 6) includes the $\hat{h}_0$ term, which plays an important role in our DAT, and allows us to inject input-specific distribution information (defined as U) into the diffusion process. Importantly, the presence of $\hat{h}_0$ provides a good way for us to update h_k such that it is more aligned towards U yet stays within H_k, which will be explained in detail in Sec. 4.2. Specifically, we adopt the following formulation [50] for the k-th reverse diffusion step:

$$h_{k-1} = \sqrt{1-\alpha_{k-1}-\sigma_k^2} \cdot g(h_k, k) + \sqrt{\alpha_{k-1}} \cdot \hat{h}_0 + \sigma_k z_k, \quad (6)$$

where h_k is a sample at the k-th step ($h_k \sim H_k$), α_k, σ_k are hyperparameters and z_k is standard Gaussian noise. We remark that this formulation for reverse diffusion is widely applicable, and as noted in [50], is equivalent to DDPM [16] when we set $\sigma_k = \sqrt{(1-\alpha_{k-1})/(1-\alpha_k)}\sqrt{1-\alpha_k/\alpha_{k-1}}$ for all k. Following [50], we can further define $\hat{h}_0$ as:

$$\hat{h}_0 = \frac{1}{\sqrt{\alpha_k}}(h_k - \sqrt{1-\alpha_k} \cdot g(h_k, k)). \quad (7)$$

Intuitively, $\hat{h}_0$ represents an estimation of the target (h_0) by using the sample at the current step (h_k) and "reverse diffusing" in one step, without taking k steps. Remarkably, including $\hat{h}_0$ in the reverse diffusion step helps to improve the efficiency, as the DDIM acceleration technique [50] can be applied during testing that allows the model to skip diffusion steps [50]. Furthermore, to facilitate the reverse diffusion process, we also extract an image feature f_I from the image, and feed it to the diffusion model g at each step. More details can be found in Sec. 4.3.

Overall, after recurrently applying Eq. 6 on h_K for K iterations, we can get h_0, where $h_0 \sim H_0$ and H_0 is the denoised (high-quality) mesh distribution. In practice, we can run this reverse diffusion process in parallel N times, and obtain N samples of h_0 to represent H_0.

Through the reverse diffusion process introduced above, we can generate a high-quality mesh output, but it is dif-

ficult to predict an accurate mesh that matches the RGB image. This is because we start the reverse diffusion from the Gaussian distribution (H_K) which is *input-agnostic* and does not contain characteristics specific to the input image. Thus, in order to recover an accurate human mesh, we ideally should perform the reverse diffusion from an *input-specific distribution* extracted from the input image, e.g., where H_K can be an extracted pose heatmap which contains rich and meaningful semantic and uncertainty information. This also has the additional benefit of allowing us to leverage on prior knowledge (from pre-trained extractors), which helps to simplify the challenging 3D human mesh recovery task. However, it is not feasible to directly initialize the starting distribution (H_K) with an input-specific distribution, since the standard reverse process is *theoretically formulated* and *trained* to start from the input-agnostic Gaussian distribution instead. Thus, we propose a way to extract an input-specific distribution from the image and use it to guide the diffusion process, as described next.

4.2. Distribution Alignment Technique (DAT)

We develop DAT to inject input-specific distribution information to the diffusion process, which *narrows down the target space for the diffusion process* towards the specific mesh distribution that accurately matches with the input image. This enables us to start the reverse process from Gaussian noise, while infusing the diffusion process with input-specific distribution information. An illustration can be seen in Fig. 1.

The Big Picture. In the reverse diffusion process, we start the diffusion from the Gaussian distribution (H_K), that is *input-agnostic* and has a large gap from the desired diffusion target H_0, which makes it difficult to predict an accurate target distribution that exactly corresponds to the input image. To overcome this, we take inspiration from previous approaches [9, 21, 51] that estimate the posterior, and aim to adopt a similar approach to guide the initial stages of the diffusion process. Specifically, we can extract an input-specific distribution U that is much closer to H_0 as compared to the Gaussian distribution, e.g., in this paper, U is a pose heatmap which contains rich semantic and uncertainty information [37, 28, 14] of the input image. Thus, as U is close to H_0, we want to align our initial (input-agnostic) distribution H_K towards the extracted distribution U at the initial stages of the diffusion process.

Intuitively, this usage of the input-specific distribution U *narrows down the target* of the diffusion process to the neighbouring area around U, where the rest of the diffusion process can further reduce the uncertainty to obtain the desired target mesh distribution H_0. This greatly reduces the difficulty of the 3D mesh recovery process. Therefore, we modify the diffusion process in a manner that fulfills this intuition, which we formulate below.

Theoretical Formulation. First, we formally introduce some definitions. We are given an input-specific distribution U and Gaussian noise $h_K \sim H_K$, and want to produce an output $h_0 \sim H_0$ after K iterations of reverse diffusion. We also define the relationship between the samples from the target distribution H_0 and the input-specific distribution U as: $f(h_0) + n = u$, where f is a function (e.g., a linear mesh-to-pose function if U is a human pose distribution), $u \sim U$, and n is some error which is defined since the distribution U can be imprecise and uncertain. Moreover, since diffusion model g can be interpreted to be a learned score estimator [52, 16] to estimate the score function (distribution gradient) $\nabla_{h_k} \log p_k(h_k)$, the reverse diffusion step in Eq. 6 can be reformulated as:

$$h_{k-1} = \sqrt{1-\alpha_{k-1}-\sigma_k^2} \cdot \nabla_{h_k} \log p_k(h_k) + \sqrt{\alpha_{k-1}} \cdot \hat{h}_0 + \sigma_k z_k \tag{8}$$

Intuitively, Eq. 8 shows that the evolution between each step (e.g., $h_k \rightarrow h_{k-1}$) is dependent on the gradient of the data distribution $\nabla_{h_k} \log p_k(h_k)$ – which is called the score function – that pushes h_k to h_{k-1}. Importantly, here we want this gradient to take into account the information from distribution U and align towards it. In other words, instead of estimating the gradient of the data distribution $\nabla_{h_k} \log p_k(h_k)$, we instead want an estimate that is conditioned on U, i.e., $\nabla_{h_k} \log p_k(h_k|U)$, which allows us to *effectively inject input-specific distribution information into the diffusion process*.

Next, we aim to find a way to compute $\nabla_{h_k} \log p_k(h_k|U)$. From Bayes' rule, we can get $\nabla_{h_k} \log p(h_k|U) = \nabla_{h_k} \log p_k(h_k) + \nabla_{h_k} \log p(U|h_k)$, where we already have the first term (i.e., the original score function). Hence, in order to compute the gradient conditioned on input U, we would only need to find a way to compute the second term, i.e., $\nabla_{h_k} \log p(U|h_k)$. Intuitively, to directly compute the gradient $\nabla_{h_k} \log p(U|h_k)$, we need to find a differentiable function that connects U and h_k and compute its gradient, i.e., minimize the gap between the noisy h_k and the distribution U. Such a gradient will directly update h_k to be closer to U, thereby injecting input-specific information directly into h_k at each step k.

Why we use $\hat{h}_0$. However, it is not feasible to directly apply the above-mentioned gradient ($\nabla_{h_k} \log p(U|h_k)$) on h_k. This is because the distribution H_k at every step k is unique and different from each other, thus the diffusion model g is trained conditioned on the step index k (see Eq. 5), to learn how to bring a sample specifically from H_k to H_{k-1}. Hence, if we forcefully align h_k to be closer to U, it can be pulled away from H_k and be different from what the diffusion model was trained to denoise at step k, and thus disrupt the diffusion process. Therefore, although we want to perform an alignment of h_k to U, but at the same

time we also want to keep the sample h_k within the distribution H_k, where the diffusion model is trained to perform well.

Next, we observe that our aim is to eventually predict an accurate mesh sample h_0 – *not to align h_k itself to U*. Thus, we propose an indirect approach to align h_k. For every h_k, we can first compute an estimate of h_0 (i.e., $\hat{h}_0$) as an intermediate prediction via Eq. 7 which is conditioned on k. Next, we can compute some gradients that will make $\hat{h}_0$ into a better prediction (i.e., by minimizing the gap with U), then propagate those gradients back from $\hat{h}_0$ to h_k. As a result, these gradients *give guidance on how to align h_k, such that after k diffusion steps the prediction $\hat{h}_0$ is pulled closer to U (and the target H_0)*, which fulfills our objective. Through this method, we can inject suitable information at every step k, updating h_k with input-specific information to be aligned with the target H_0, while keeping the sample within the distribution H_k.

Specifically, we first estimate $\hat{h}_0$ as a function of h_k (i.e., in Eq 7); we slightly abuse notation to denote this as $\hat{h}_0(h_k)$, to make it clear that $\hat{h}_0$ is estimated as a function of h_k. Using $\hat{h}_0(h_k)$, we can approximate $p(U|h_k)$ by computing $p(U|\hat{h}_0(h_k))$, which can be used to compute the gradients to update h_k.

Distribution Alignment Gradient. Next, we show how the gradient $\nabla_{h_k} \log p(U|\hat{h}_0(h_k))$ can be computed in practice to obtain a Distribution Alignment Gradient that can inject input-specific distribution information into the diffusion process. Specifically, we can interpret $-\log p(U|\hat{h}_0)$ to be the negative log-posterior of observing U given $\hat{h}_0$, which tends to have higher magnitude as the difference between U and $\hat{h}_0$ gets larger, and we can derive a gradient by trying to minimize the gap D_k between them. However, U is a pose distribution, while $\hat{h}_0$ is in mesh format, making it difficult to directly compute the gap D_k. Thus, we introduce a mesh-to-pose function f to map $\hat{h}_0$ to a corresponding pose, and minimize the gap between U and $f(\hat{h}_0)$. To efficiently calculate D_k, we sample N elements from U (i.e., $u \sim U$), and calculate the sum of L_2 norms between u and $f(\hat{h}_0)$, as follows:

$$\nabla_{h_k} \log p(U|h_k) \approx -\nabla_{h_k} D_k, \tag{9}$$

$$D_k = \sum_{i=1}^{N} ||u_i - f(\hat{h}_0(h_k))||_2^2, \; u_i \sim U, \tag{10}$$

where $-\nabla_{h_k} D_k$ is our Distribution Alignment Gradient. In practice, we also add a hyperparameter γ as a coefficient to $-\nabla_{h_k} D_k$. This Distribution Alignment Gradient enables us to align the diffusion steps towards the distribution U (and the diffusion target H_0) at the initial stages of the diffusion process. This narrows down the target space for the diffusion process towards the specific mesh distribution that corresponds to the input image, which greatly reduces the difficulty of the 3D mesh recovery process. However, the role of DAT diminishes in importance towards the end of the diffusion process, after having aligned h_k towards U and H_0. Thus, we design an Activation Strategy to decide when we apply the gradient, as explained next.

Algorithm 1: The DAT Reverse Diffusion Process

Input: input-specific distribution U, number of samples N, decreasing sequence $\alpha_{1:K}$, sequence $\sigma_{1:K}$, diffusion model g, threshold r, DAT weight γ.

Sample a noise h_K, where $h_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$act = True$ // Initialize activation status
for $k = K$ *to* 1 **do** // Reverse diffusion process
  $s \leftarrow g(h_k, k)$ // Estimate score
  $\hat{h}_0 \leftarrow \frac{1}{\sqrt{\alpha_k}}(h_k - \sqrt{1-\alpha_k} \cdot s)$ // Estimate $\hat{h}_0$
  $D_k \leftarrow \sum_{i=1}^{N} ||u_i - f(\hat{h}_0)||_2^2$ // Measure the gap
  $R_k \leftarrow \frac{D_k}{D_K}$ // Measure the relative gap
  if $R_k \geq r$ *and* $act == True$ **then** // with DAT
    $h_{k-1} \leftarrow \sqrt{1-\alpha_{k-1}-\sigma_k^2} \cdot s + \sqrt{\alpha_{k-1}} \cdot \hat{h}_0 - \gamma \nabla_{h_k} D_k + \sigma_k z_k$
  else // without DAT
    $act \leftarrow False$
    $h_{k-1} \leftarrow \sqrt{1-\alpha_{k-1}-\sigma_k^2} \cdot s + \sqrt{\alpha_{k-1}} \cdot \hat{h}_0 + \sigma_k z_k$

Activation Strategy for DAT. Our DAT seeks to infuse input-specific characteristics into the diffusion process, and align the initial (input-agnostic) Gaussian distribution H_K towards an input-specific distribution U, which is close to the target H_0. However, as the extracted input-specific distribution U contains uncertainty and noise, we do not want to overly align H_k towards U in the later parts of the diffusion process, when H_k is relatively more accurate and certain. Therefore, we want to use our proposed DAT only in the early reverse diffusion steps, while not using it in the later steps when H_k converges to a more compact and high-quality distribution. To achieve that, we propose activating (or deactivating) DAT based on the measured gap (D_k) between h_k and U.

Specifically, at the start, we measure the initial gap D_K (between U and h_K) based on Eq. 10. Then, at each step k, we measure the gap D_k, and calculate the relative distribution gap value $R_k = \frac{D_k}{D_K}$. At the start (when $k = K$), $R_k = 1$, and it starts to shrink as H_k slowly gets aligned to U. We activate DAT as long as the relative distribution gap value R_k is more than a specified threshold r. On the other hand, when $R_k < r$ for some k-th step, we terminate DAT for the steps thereafter, because the distribution H_k has become rather aligned to U and the target H_0.

In summary, the detailed reverse diffusion process with our DAT can be seen in Algorithm 1.

4.3. Network Architecture

As shown in Fig. 1, our full pipeline consists of a CNN backbone ϕ_I with a 3D pose estimator head ϕ_P, a diffusion model g and a DAT component. We present the details of each of them below, with more details in Supplementary.

CNN backbone ϕ_I. We follow previous works [6, 33, 32] to adopt HRNet [57] as our CNN backbone ϕ_I, that

extracts a context feature $f_c \in \mathbb{R}^{2048\times7\times7}$ from the input image I, which is sent into the pose estimator head ϕ_P. Moreover, to produce an image feature $f_I \in \mathbb{R}^{128\times49}$ to feed into diffusion model g, we also flatten f_c and send it into an average pooling layer.

Pose estimator head ϕ_P is used to initialize the input-specific distribution U, and is a lightweight module consisting of three de-convolutional layers. ϕ_P takes in context features $f_c \in \mathbb{R}^{2048\times7\times7}$ extracted from the CNN backbone ϕ_I, and generates an xy heatmap $E_{x,y} \in \mathbb{R}^{J\times56\times56}$ and a depth heatmap $E_z \in \mathbb{R}^{J\times56\times56}$, where J is the number of joints. Since a heatmap is naturally a distribution, we initialize a 3D pose distribution U based on these heatmaps to guide the reverse diffusion process.

Diffusion model g facilitates the step-by-step denoising during the reverse process, as shown in the red block of Fig. 1. At the start of step k, we are given a noisy 3D mesh input $h_k \in \mathbb{R}^{V\times3}$ where V is the number of vertices. We first encode each v-th vertex ID to an embedding $E^v_{ID} \in \mathbb{R}^{64}$ via a linear layer and generate a diffusion step embedding $E^k_d \in \mathbb{R}^{61}$ for the k-th step via the sinusoidal function. Then, we construct V tokens, where each token $x_v \in \mathbb{R}^{128}$ represents the v^{th} vertex, and each token is constructed by concatenating E^v_{ID}, E^k_d, and the 3D coordinates of the v-th vertex in h_k. These tokens are sent into our diffusion network g, which consists of a single vertex self-attention layer, a single vertex-image cross-attention layer, and a linear layer. Refer to Supplementary for more details.

Distribution Alignment Technique. In this paper, we define the function f in DAT as a linear operation (i.e., a matrix) defined using the human *body* model SMPL [35] (or human *hand* model MANO [45]), that regresses 3D joint locations from the estimated mesh sample ($\hat{h}_0$).

4.4. Training

Learning Reverse Diffusion Process. As introduced in previous sections, our diffusion model g is optimized to iteratively denoise H_K to get H_0, i.e., $H_K \rightarrow H_{K-1} \rightarrow ... \rightarrow H_0$. To achieve this, we first generate "ground truth" intermediate distributions $\{H_1, H_2, ..., H_{K-1}\}$ via the *forward diffusion process*, where we take a ground truth mesh distribution H_0 and gradually add noise to it based on Eq. 4. Then, during model training, we follow previous works [16, 50] to formulate *diffusion reconstruction loss* $\mathcal{L}_{Diff}$ as follows: $\mathcal{L}_{Diff} = \sum_{k=1}^{K} \|(h_{k-1} - h_k) - g(h_k, k, f_I)\|_2^2$, where $h_k \sim H_k$ and $h_{k-1} \sim H_{k-1}$.

Learning Mesh Geometry. To recover an accurate human mesh, we also optimize our diffusion model with the geometric constraints of human mesh. Specifically, at each diffusion step, we obtain the estimate of diffusion target $\hat{h}_0$ via Eq. 7 and then follow previous work [7] to optimize our model via 4 kinds of losses to constrain the mesh geometry: 3D Vertex Regression Loss $\mathcal{L}_v$, 3D Joint Regression Loss $\mathcal{L}_j$, Surface Normal Loss $\mathcal{L}_n$, and Surface Edge Loss $\mathcal{L}_e$. See Supplementary for more details.

Total loss. Combining the losses described above, we define the total loss for training our HMDiff framework as follows: $\mathcal{L}_{total} = \mathcal{L}_{Diff} + \lambda_v\mathcal{L}_v + \lambda_j\mathcal{L}_j + \lambda_n\mathcal{L}_n + \lambda_e\mathcal{L}_e$.

5. Experiments

Datasets. We follow previous works [6, 33, 32] to evaluate our method on the following datasets. **3DPW** [56] consists of outdoor images with both 2D and 3D annotations. The training set has 22K images, and the test set has 35K images. **Human3.6M** [19] is a large-scale indoor dataset that has 3.6M images labelled with 2D and 3D annotations. Following the setting in previous works [32, 27, 20], we train our models using subjects S1, S5, S6, S7 and S8 and test using subjects S9 and S11. **FreiHAND** [61] consists of hand actions with 3D annotations. It has approximately 32.5K training images and 3.9K testing images.

Specifically, we follow previous works [6, 33, 32] to first train our model with the training sets of Human3.6M [19], UP-3D [29], MuCo-3DHP [38], COCO [34] and MPII [2], and then evaluate the model on Human3.6M. Moreover, we follow [6, 33, 32] to fine tune the model on 3DPW [56] training set and evaluate our model on its test dataset. For FreiHAND [61], following [32], we optimize our model on its training set and test our model on its evaluation set.

Evaluation Metrics. We follow the evaluation metrics from previous works [32, 27, 20, 33, 7, 6]. Mean-Per-Vertex-Error (**MPVE**) [42] measures the Euclidean distance (in mm) between the predicted vertices and the ground truth vertices. Next, Mean-Per-Joint-Position-Error (**MPJPE**) [19] is a metric for evaluating human 3D pose [26, 20, 7], and measures the Euclidean distance (in mm) between the predicted joints and the ground truth joints. **PA-MPJPE**, or Reconstruction Error [60], measures MPJPE after using Procrustes Analysis (PA) [11] to perform 3D alignment. We also report **PA-MPVPE** on FreiHAND, which measures MPVE after performing 3D alignment with PA. On FreiHAND, we also report the F-score [23], which is the harmonic mean of recall and precision between two sets of points, given a specified distance threshold. Following previous works [32, 33, 7, 6], we report the F-score at 5mm and 15mm (**F@5 mm** and **F@ 15mm**) to evaluate accuracy at fine and coarse scales respectively.

Implementation Details. During training, we set the total number of diffusion steps (K) at 200 and generate the decreasing sequence $\alpha_{1:200}$ via a linear interpolation function (more details in Supplementary). We also set the number of samples N to 25. Following previous works [6, 33, 32], we obtain the coarse human mesh with 431 vertices from the original SMPL human mesh (or a human hand mesh with 195 vertices from the original MANO hand mesh) via a GCN model [44] for training. We set the learning rate at

0.0001 and adopt the Adam optimizer to optimize our diffusion model g. The models ϕ_I and ϕ_P are pre-trained and then frozen during training of g. For DAT, we set the activation threshold r at 0.05, and γ at 0.2. During testing, we adopt the DDIM acceleration technique [50], and take 40 steps to complete the whole reverse diffusion process instead of 200.

5.1. Comparison with State-of-the-art Methods

We compare our method with existing state-of-the-art HMR methods on Human3.6M and 3DPW datasets. As shown in Tab. 1, our proposed method can outperform previous works on all metrics. This shows that our proposed method (i.e., HMDiff framework with DAT) can effectively recover a high-quality human mesh from a single image.

To further demonstrate the capability of our model to tackle hand mesh reconstruction, we also conduct experiments on FreiHAND dataset [61]. As shown in Tab. 2, our method outperforms previous state-of-the-art methods, showing its generalizability in this setting as well.

Table 1. Comparison results on 3DPW and Human3.6M.

Method	3DPW			Human3.6M	
	MPVE↓	MPJPE↓	PA-MPJPE↓	MPJPE↓	PA-MPJPE↓
Kanazawa et al. [20]	-	-	81.3	88.0	56.8
GraphCMR [27]	-	-	70.2	-	50.1
SPIN [26]	116.4	-	59.2	-	41.1
Pose2Mesh [7]	-	89.2	58.9	64.9	47.0
I2LMeshNet [39]	-	93.2	57.7	55.7	41.1
VIBE [24]	99.1	82.0	51.9	65.6	41.4
METRO [32]	88.2	77.1	47.9	54.0	36.7
Mesh Graphormer [33]	87.7	74.7	45.6	51.2	34.5
FastMETRO [6]	84.1	73.5	44.6	52.2	33.7
Ours	**82.4**	**72.7**	**44.5**	**49.3**	**32.4**

Table 2. Comparison results on FreiHAND. The results with an asterisk (*) are reported by [7].

Method	PA-MPVPE↓	PA-MPJPE↓	F@5 mm↑	F@15 mm↑
Hasson et al. [15] *	13.2	-	0.436	0.908
Boukhayma et al. [5] *	13.0	-	0.435	0.898
FreiHAND [61] *	10.7	-	0.529	0.935
Pose2Mesh [7]	7.8	7.7	0.674	0.969
I2LMeshNet [39]	7.6	7.4	0.681	0.973
METRO [32]	6.7	6.8	0.717	0.981
Mesh Graphormer [33]	5.9	6.0	0.764	0.986
FastMETRO [6]	-	6.5	-	0.982
Ours	**5.7**	**5.6**	**0.781**	**0.986**

5.2. Ablation Study

We also conduct extensive ablation experiments on 3DPW. **See Supplementary for more experiments and analysis.**

Impact of Diffusion Process. We evaluate the efficacy of the diffusion process by comparing against two baseline models: (1) **Baseline A** possesses the same structure as our diffusion network, but the mesh recovery is conducted in a single step without diffusion. (2) **Baseline B** has nearly the same architecture as our diffusion network, but we stack the network multiple times to approximate the computational cost of our method. We remark that both baselines are optimized to directly predict the human mesh with a forward pass, instead of learning the step-by-step reverse diffusion process. We report the results of our proposed method and the baselines in Tab. 3. The performance of both baselines are much worse than ours, suggesting that much of the performance improvement comes from the designed diffusion pipeline.

Table 3. Evaluation of diffusion pipeline.

Method	MPVE	MPJPE	PA-MPJPE
Baseline A	104.0	93.2	57.6
Baseline B	97.1	85.9	52.7
Ours	82.4	72.7	44.5

Impact of DAT. We also verify the impact of our DAT by comparing against three baselines: (1) **Image Feature** where the standard diffusion process is used (without DAT), with image feature f_I as an input feature. (2) **Pose Feature** where the standard diffusion process is used (without DAT), with pose heatmap U as an input feature. (3) **Both Features** where we concatenate the pose heatmap with the image features f_I before feeding them to the diffusion model g. For each method, we tune the DDIM [50] acceleration rate to perform the reverse diffusion process with different diffusion steps, and report the number of steps used where they obtain the best performance. Results are reported in Tab. 4, where our method significantly outperforms all baselines while using fewer steps. This is because, as compared to these baselines, our DAT can explicitly constrain the initial steps to align towards the input-specific distribution, which improves the performance as well as the speed of convergence to the target H_0.

Table 4. Evaluation of impact of DAT.

Method	MPVE	MPJPE	PA-MPJPE	Steps needed
Image Feature	94.9	83.3	51.3	200
Pose Feature	92.7	81.4	50.5	200
Both Features	92.3	81.1	49.9	100
Ours (with DAT)	82.4	72.7	44.5	40

Impact of DAT components. We study the impact of various DAT components by comparing against the following: (1) **Ours w/ H_U** starts the reverse process from a noisy distribution H_U, generated by upsampling the prior pose distribution U to a mesh distribution. (2) **Direct L_2** baseline directly aligns h_k to the prior pose distribution U via an L_2 loss which directly modifies h_k (i.e., disrupts the diffusion process). (3) **Ours w/ u_c** uses a single 3D pose u_c as prior knowledge (detected from the heatmap), instead of the 3D pose distribution U. (4) **Ours w/o AS** activates DAT over the whole reverse process, i.e., without using the Activation Strategy. As shown in Tab. 5, our method outperforms all baselines, showing the efficacy of our design choices.

Table 5. Evaluation of DAT components.

Method	MPVE	MPJPE	PA-MPJPE
Ours w/ H_U	93.5	82.1	50.4
Direct L_2	91.3	78.0	50.0
Ours w/ u_c	89.9	76.5	49.2
Ours w/o AS	85.3	72.9	46.6
Ours	82.4	72.7	44.5

Visualization of difficult samples. Some qualitative re-

Figure 2. Visualization of body and hand mesh outputs using our method. Our method effectively recovers the mesh even under ambiguity (e.g., due to heavy occlusions), and produces high-quality results. (More visualization results are shown in Supplementary.)

sults of our method on Human3.6M and 3DPW are shown in Fig. 2. We observe that our method can handle challenging cases, e.g., under heavy occlusions or with noisy background, showing its strong ability to handle uncertainty.

Inference Speed. Here, we conduct experiments on a single GeForce RTX 3090 card and compare the speed of our proposed method (HMDiff with DAT) with existing methods in terms of the accuracy metric (MPJPE) and inference speed (FPS) in Tab. 6. Our method can achieve a competitive speed compared with the current SOTA [6] while significantly outperforming it (as shown in Tab. 1 and Tab. 2).

Table 6. Comparison of inference speed.

Method	Human3.6M(MPJPE)	FPS
[6]	52.2	23
Ours	49.3	18

6. Conclusion

We present HMDiff, a novel diffusion-based framework that frames mesh recovery as a reverse diffusion process to tackle monocular HMR. We inject input-specific information via our proposed DAT during the diffusion process to obtain improved performance. Extensive experiments show that the proposed method achieves state-of-the-art performance on three widely used benchmark datasets.

7. Acknowledgements

This work was supported by the Singapore Ministry of Education (MOE) AcRF Tier 2 under Grant MOE-T2EP20222-0009, the National Research Foundation Singapore through AI Singapore Programme under Grant AISG- 100E-2020-065, and SUTD SKI Project under Grant SKI 2021_02_06

References

[1] Jake K Aggarwal and Quin Cai. Human motion analysis: A review. *Computer vision and image understanding*, 73(3):428–440, 1999. 1

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014. 7

[3] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005. 2

[4] Benjamin Biggs, David Novotny, Sebastien Ehrhardt, Hanbyul Joo, Ben Graham, and Andrea 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. 2

[5] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019. 8

[6] Junhyeong Cho, Kim Youwang, and Tae-Hyun Oh. Cross-attention of disentangled modalities for 3d human mesh recovery with transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, pages 342–359. Springer, 2022. 1, 2, 3, 4, 6, 7, 8, 9

[7] 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 *European Conference on Computer Vision*, pages 769–787. Springer, 2020. 1, 2, 4, 7, 8

[8] Hongsuk Choi, Gyeongsik Moon, JoonKyu Park, and Kyoung Mu Lee. Learning to estimate robust 3d human mesh from in-the-wild crowded scenes. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1475–1484, 2022. 1, 2

[9] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. 2, 5

[10] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023. 2

[11] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40:33–51, 1975. 7

[12] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17113–17122, 2022. 2

[13] Peng Guan, Alexander Weiss, Alexandru O Balan, and Michael J Black. Estimating human shape and pose from a single image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1381–1388. IEEE, 2009. 2

[14] Chuchu Han, Xin Yu, Changxin Gao, Nong Sang, and Yi Yang. Single image based 3d human pose estimation via uncertainty learning. *Pattern Recognition*, 132:108934, 2022. 2, 5

[15] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11807–11816, 2019. 8

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 2, 3, 4, 5, 7

[17] Yinghao Huang, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V Gehler, Javier Romero, Ijaz Akhter, and Michael J Black. Towards accurate marker-less human shape and pose estimation over time. In *2017 international conference on 3D vision (3DV)*, pages 421–430. IEEE, 2017. 1

[18] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018. 2

[19] 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*, 36(7):1325–1339, 2013. 7

[20] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7122–7131, 2018. 2, 7, 8

[21] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in Neural Information Processing Systems*, 35:23593–23606, 2022. 2, 5

[22] Rawal Khirodkar, Shashank Tripathi, and Kris Kitani. Occluded human mesh recovery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1715–1725, 2022. 1, 2

[23] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 7

[24] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5253–5263, 2020. 8

[25] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael 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. 1, 2, 3

[26] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2252–2261, 2019. 7, 8

[27] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4501–4510, 2019. 2, 7, 8

[28] Jogendra Nath Kundu, Siddharth Seth, Pradyumna YM, Varun Jampani, Anirban Chakraborty, and R. Venkatesh Babu. Uncertainty-aware adaptation for self-supervised 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20448–20459, June 2022. 2, 5

[29] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6050–6059, 2017. 2, 7

[30] Gun-Hee Lee and Seong-Whan Lee. Uncertainty-aware human mesh recovery from video by learning part-based 3d dynamics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12375–12384, 2021. 1, 3

[31] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022. 1, 2

[32] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021. 1, 2, 6, 7, 8

[33] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *Proceedings of the IEEE/CVF International*

Conference on Computer Vision, pages 12939–12948, 2021. 1, 2, 6, 7, 8

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 7

[35] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 2, 7

[36] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 1

[37] Zhengxiong Luo, Zhicheng Wang, Yan Huang, Liang Wang, Tieniu Tan, and Erjin Zhou. Rethinking the heatmap regression for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13264–13273, 2021. 2, 5

[38] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. In *2018 International Conference on 3D Vision (3DV)*, pages 120–130. IEEE, 2018. 7

[39] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *European Conference on Computer Vision*, pages 752–768. Springer, 2020. 1, 2, 8

[40] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 2

[41] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *2018 international conference on 3D vision (3DV)*, pages 484–494. IEEE, 2018. 2

[42] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 459–468, 2018. 2, 7

[43] Duo Peng, Ping Hu, Qiuhong Ke, and Jun Liu. Diffusion-based image translation with label guidance for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2

[44] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European conference on computer vision (ECCV)*, pages 704–720, 2018. 7

[45] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017. 7

[46] Daeyun Shin, Zhile Ren, Erik B Sudderth, and Charless C Fowlkes. 3d scene reconstruction with multi-layer depth and epipolar transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2172–2182, 2019. 2

[47] Leonid Sigal, Alexandru Balan, and Michael Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. *Advances in neural information processing systems*, 20, 2007. 2

[48] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 1, 2

[49] 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, 2, 3

[50] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2, 3, 4, 7, 8

[51] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. 2, 5

[52] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2, 3, 5

[53] Yating Tian, Hongwen Zhang, Yebin Liu, and Limin Wang. Recovering 3d human mesh from monocular images: A survey. *arXiv preprint arXiv:2203.01923*, 2022. 1, 2, 3

[54] Philip Treleaven and Jonathan Wells. 3d body scanning and healthcare applications. *Computer*, 40(7):28–34, 2007. 1

[55] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011. 3

[56] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard 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. 2, 7

[57] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020. 2, 6

[58] Zhenbo Yu, Junjie Wang, Jingwei Xu, Bingbing Ni, Chenglong Zhao, Minsi Wang, and Wenjun Zhang. Skeleton2mesh: Kinematics prior injected unsupervised human mesh recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8619–8629, 2021. 2

[59] Andrei Zanfir, Eduard Gabriel Bazavan, Mihai Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Neural descent for visual 3d human pose and shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14484–14493, 2021. 2

[60] Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):901–914, 2018. 7

[61] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2019. 7, 8