3D Magic Mirror: Clothing Reconstruction from a Single Image via a Causal Perspective

Zhedong Zheng[1] Jiayin Zhu[1] Wei Ji[1] Yi Yang[2] Tat-Seng Chua[1]
[1]Sea-NExT Joint Lab, National University of Singapore [2] Zhejiang University

Abstract

This research aims to study a self-supervised 3D clothing reconstruction method, which recovers the geometry shape and texture of human clothing from a single image. Compared with existing methods, we observe that three primary challenges remain: (1) 3D ground-truth meshes of clothing are usually inaccessible due to annotation difficulties and time costs; (2) Conventional template-based methods are limited to modeling non-rigid objects, e.g.*, handbags and dresses, which are common in fashion images; (3) The inherent ambiguity compromises the model training, such as the dilemma between a large shape with a remote camera or a small shape with a close camera.*

In an attempt to address the above limitations, we propose a causality-aware self-supervised learning method to adaptively reconstruct 3D non-rigid objects from 2D images without 3D annotations. In particular, to solve the inherent ambiguity among four implicit variables, i.e.*, camera position, shape, texture, and illumination, we introduce an explainable structural causal map (SCM) to build our model. The proposed model structure follows the spirit of the causal map, which explicitly considers the prior template in the camera estimation and shape prediction. When optimization, the causality intervention tool,* i.e.*, two expectation-maximization loops, is deeply embedded in our algorithm to (1) disentangle four encoders and (2) facilitate the prior template. Extensive experiments on two 2D fashion benchmarks (ATR and Market-HQ) show that the proposed method could yield high-fidelity 3D reconstruction. Furthermore, we also verify the scalability of the proposed method on a fine-grained bird dataset,* i.e.*, CUB. The code is available at* https://github.com/layumi/3D-Magic-Mirror.

1. Introduction

Nowadays, people can purchase clothing via online shopping sites, *e.g.*, Amazon and eBay. However, there remains a gap between the display images and the real product quality [7, 32]. In an attempt to minimize such a visualization gap, we study the 3D clothing reconstruction from a single image. Given a 2D clothing image and the foreground mask, we intend to reconstruct a 3D mesh, which recovers the geometry shape and texture of the target clothing. Besides, the clothing reconstruction can also be applied to many computer vision applications, including virtual reality [53], interactive system [47, 52] and 3D printing [4].

Figure 1: Motivation. Here we compare the proposed approach with prevailing template-based methods, *i.e.*, HMR [19] and ROMP [55] on a fashion dataset ATR [28]. We re-implement and visualize results with the color mapping according to the projected location. The first row is the front view, and the second row is the 3D mesh rotated with 45°. The template-based model can capture the human poses but miss non-rigid objectives, such as hairs, handbags, and dresses.

However, there remain three challenges. **First,** 3D annotations are difficult to obtain due to the annotation difficulty and time costs. There are no public large-scale 3D clothing mesh datasets for supervised learning. In contrast, the availability of the large-scale 2D fashion datasets, such as ATR [28] and Market-HQ [79], makes training data-hungry deep-learned approaches become feasible. The success has been proved in the 2D pedestrian image generation [8, 11, 36, 45, 51]. With the recent development in self-supervised learning and deep-learned models, one straightforward idea is raised whether we can leverage 2D data for 3D reconstruction, even without manual 3D annotations. **Second,** existing works [19, 29, 30] typically focus on human pose estimation and body reconstruction via parametric models, *e.g.*, a morphable body template [34]. However, pre-defined body parameters usually are not scalable to non-rigid clothing, *e.g.*, dresses, handbags, and loose clothing [16, 69], losing fine-grained clothing details. As shown in Figure 1, we re-implement two prevailing meth-

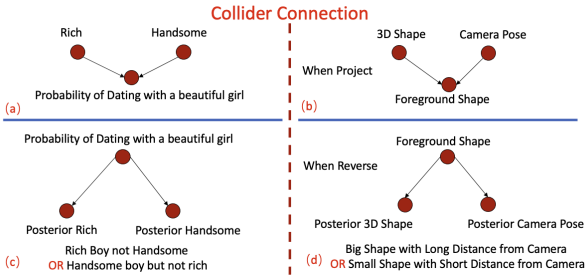

Figure 2: Explanation of the Collider Connection. Here we show a common example "Dating with Beautiful Girl" [43] in (a), which is similar to our simplified dilemma on two variants, *i.e.*, shape and camera pose in (b). Since we study an inverse problem, we also draw (c) and (d). We note that there is a compensation effect when we estimate the posterior probability. As shown in (d), given the observed foreground shape, the model needs to estimate the posterior 3D shape and posterior camera pose. There are two possible alternatives for the network to learn, *i.e.*, a big shape from a long distance or a small shape from a short distance. Therefore, it makes the model difficult to converge an answer.

ods, *i.e.*, HMR [19] and ROMP [55], which both successfully capture the human pose but miss the cloth shape. In contrast, our methods leverage a deformable model to further facilitate learning non-rigid objects. **Third,** one scientific question still remains in single image 3D reconstruction. The primary implicit variables for reconstruction are camera viewpoint, shape, texture, and illumination. Ideally, these four factors are independent. However, it remains challenging to disentangle the implicit variables in practice. One typical dilemma is the ambiguity between the camera and shape [26]. Given a 2D image, it is hard to decide the object size. There are two possible answers (see Figure 2). One large object is far from the camera or one small object is close to the camera. Despite different physical sizes, the two objects have the same projection size in photos.

In an attempt to overcome the above-mentioned challenges, this paper proposes a self-supervised learning approach to adaptively reconstruct 3D clothing from a single image without 3D mesh annotation. We study the existing works (see Figure 3) and introduce an explainable structural causal map (SCM) to build our model and guide the optimization strategy. (1) Following the spirit of the causal map, we deploy four independent 3D attribute encoders and a differentiable render for reconstruction. The encoders are to extract 3D attributes from 2D images and foreground masks, including camera viewpoint, geometric shape, texture, and illumination. Then the attributes are fed to the differentiable render to reconstruct the 3D mesh. Different from existing works [17, 26], we explicitly introduce the prior template to help the camera estimation and shape offsets estimation, which is aligned with human observation. If we foreknow the human prototype, it helps us to predict the camera position as well as the intra-class variant (such as leg movement). (2) We leverage the causality intervention tool, *i.e.*, two expectation-maximization loops, to help learn the prototype, and disentangle encoders from the confusing loss punishment. To summarize, our contributions are two-fold:

- We identify the three challenging problems in the 3D clothing reconstruction: 1) No 3D annotations; 2) Non-rigid objects; 3) Reconstruction ambiguity. In an attempt to solve these challenges, we propose a self-supervised learning method with a causality design to reconstruct the 3D clothing mesh from large-scale 2D image datasets. Following the spirit of the causal map, we re-design the encoder structure and leverage the "intervention" tool, *i.e.*, two expectation-maximization loops, to facilitate the 3D attribute encoder learning.
- Experiments on two fashion datasets verify the effectiveness of the proposed method quantitatively and qualitatively. Furthermore, experiments on the fine-grained bird dataset also show that the proposed method has good scalability to other non-rigid objects.

Table 1: Comparison with existing methods on supervisions. The proposed method harnesses relatively weak supervision for 3D reconstruction from a single image. It is also worth noting that some works take 2D input images with white background as inputs, and we also view this line of works deploying the foreground mask.

Methods	Viewpoint Annotations	Semantic Keypoint	Manual Template	Part Seg.	Foreground Mask
VPL [21]	✓				✓
CMR [20]† *	(✓)	✓	(✓)		✓
CSM [24]			✓		✓
DIB-R [6]	✓				✓
IMR [60]*			(✓)		✓
ACMR-vid [25]		✓			✓
UMR [26]				✓	✓
WLDO [3]		✓	✓		✓
Texformer [70]			✓	✓	✓
MeshInversion [77]					✓
SMR [17]					✓
Ours					✓

*: The method deploys the manual template for initialization;
†: The viewpoint annotation is optional.

2. Related Work

3D Reconstruction from Single Image. Humans can estimate 3D structures from a single image. Many works deploy a parameter-based template [19, 30, 55], which is robust but also limits the representative ability to non-rigid objects. To enable more degrees of freedom, Deephuman [82] adopts a U-Net model to reconstruct the human body and clothing voxel, but dense depths and ground-truth 3D annotations are needed. To reduce the dependency on 3D annotations, PrGANs [10] trains a generative adversarial network (GAN) to generate the 3D voxel from lots of 2D images with different viewpoints. Tulsiani *et al.* [61] further leverage multi-view photos of the identical object to reconstruct the 3D voxel model in an unsupervised manner. To avoid the dense prediction of the voxel format, Pixel2Mesh [64, 68] is a fully supervised learning work with the well-designed regularization, which reconstructs

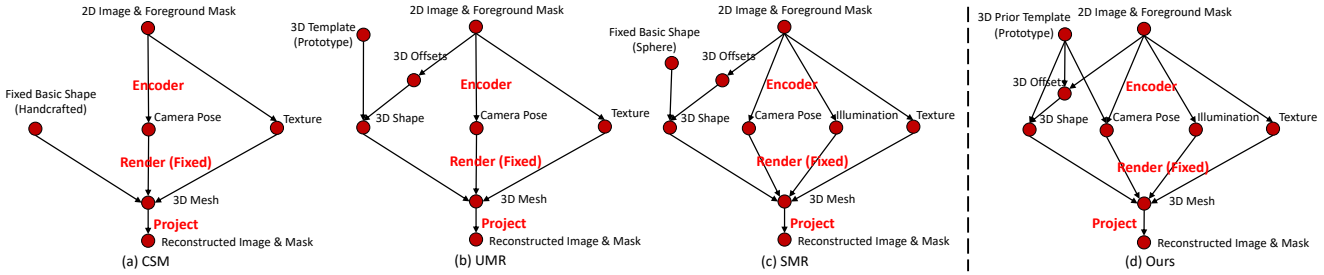

Figure 3: The Structural Causal Map (SCM). We compare the proposed method with three typical 3D reconstruction works, including CSM [24], UMR [26], and SMR [17]. Here we show the image reconstruction loop, *i.e.*, "2D $\rightarrow$ 3D $\rightarrow$ 2D". (a) Given one 2D image and the foreground mask, CSM only applies two encoders for Camera and Texture, since it assumes that the basic shape is shared, ignoring intra-class changes. (b, c) UMR and SMR further introduce the shape encoder, which is to predict the shape offset. The final shape is obtained by adding the shape offset to the shape template. (d) In this work, we argue that the predicted shape offsets are also conditioned on the prior template. Besides, the prior template also impacts the camera prediction. Therefore, we explicitly introduce the dependency with the prior prototype. It is worth noting that our prototype is initialized from an ellipsoid and iteratively updated during training, which does not require any hand-crafted initialization. (Qualitative comparison with more other methods is listed in Table 1.)

mesh by deforming an ellipsoid. VPL [21] leverages viewpoint annotation to ensure mesh reconstruction consistency via adversarial training. To invade the viewpoint annotation, one of the early works is Canonical Surface Mapping (CSM) [24], which treats 3D reconstruction as a dense keypoint estimation problem. However, CSM deploys a fixed pre-defined shape template, which largely limits intra-class shape changes for different instances. To address the shape limitation, CMR [20] first proposes to use a learnable shape template and Li *et al.* [26] further proposes UMR, which adopts a two-stage training strategy for template updating. The segmentation parsing [18] is used in UMR for better alignment. The contemporary work, IMR [60], first introduces the mapping function instead of the vertex location regression, saving computation costs. Taking a further step, SMR [17] aligns the 3D mesh via mix-up and conducting the auxiliary vertex classification, while MeshInversion [77] deploys foreground prediction. The proposed method is mainly different from existing works in three aspects: (1) **Weaker supervision.** As shown in Table 1, the proposed method demands limited supervision and mainly leverages the large-scale multi-view images to learn prior knowledge. (2) **Model design.** As shown in Figure 4, the network design follows the causal map. (3) **Optimization strategy.** To deal with the compensation effect in the loss punishment, we deploy the "intervention" tool, *i.e.*, two expectation-maximization loops, to facilitate the 3D attribute learning.

Causal Learning. Causal learning is to identify causalities from a set of empirical factors, which can be either pure observations or counterfactual inference [44]. To represent causalities, a causal model is usually defined via structural equations and graphs [42]. According to the structural causal model, manipulations can be conducted to optimize the estimated relations between variables, *e.g.*, "Do" operation is to cut certain directed edges and control the target variable [42]. One line of works using the counterfactual thought is to conduct data augmentation [5, 27, 37, 48, 67] and obtain the debiased prediction [39,59,76]. Another line of works on the generative model mainly explores implicit causal learning to discover the causal factors during training [33, 78]. CausalGAN [23] trains a generator, which is consistent with an implicit causal graph, and is able to sample from either conditional labels or interventional distributions. Similarly, CausalVAE [73] is a VAE-based causal framework, which discovers latent causal factors in data with graph constraints. Both methods implicitly harness causal mapping by learning latent code or adding one graph constraint. However, the causality learned from data is not always accurate and explainable, limiting the causality effect. Differently, our model follows the spirit of causality between semantic entities to (1) explicitly consider the causality relation between entities, *e.g.*, leveraging the prior template to help both camera encoder and shape encoder learning; and (2) explicitly apply "intervention" tools to solve the ambiguity of learning multiple variables.

Expectation Maximization (EM). EM is an iterative method to find the parameters with maximum likelihood in statistical models [38]. The EM algorithm iteratively conducts two kinds of steps: an expectation step (E-step) to obtain the expectation of latent variables and a maximization step (M-step) which computes parameters to maximize the expected likelihood based on the latent variables. Since the M-step updates parameters, it affects the E-step in the next round. In this way, the EM algorithm can keep updating until the convergence, and is usually applied to scenarios that miss the observation of implicit variables, such as Gaussian mixture model [71]. For 3D reconstruction, we also meet a similar problem to estimate the four reconstruction factors simultaneously. Inspired by EM, we propose a similar optimization strategy in our work, and this process actually is a "Do" operation in the causal map [43].

3. Method

3.1. Overview

Given a clothing image I_i and the foreground mask M_i, we aim to infer the corresponding 3D mesh with texture

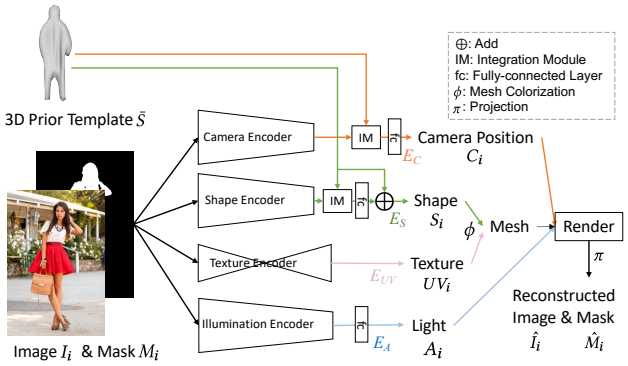

Figure 4: Overview. Here we show a "2D→3D→2D" loop. We follow the causal map in Figure 3 (d) to design the pipeline. Given one pair of clothing image I_i and the mask M_i, we deploy four independent encoders E_c, E_s, E_{UV}, and E_A for camera position, shape, texture and light estimation. We introduce the integration module (IM) to fuse the local feature from 3D prior template $\bar{S}$. Then we apply the colorize function ϕ to obtain the mesh, and utilize the render to re-project the mesh into 2D space via the project function π. Finally, we obtain the reconstructed $\hat{I}_i$ and $\hat{M}_i$. During inference, we manipulate intermediate camera attributes C_i to generate novel-view images of the target person.

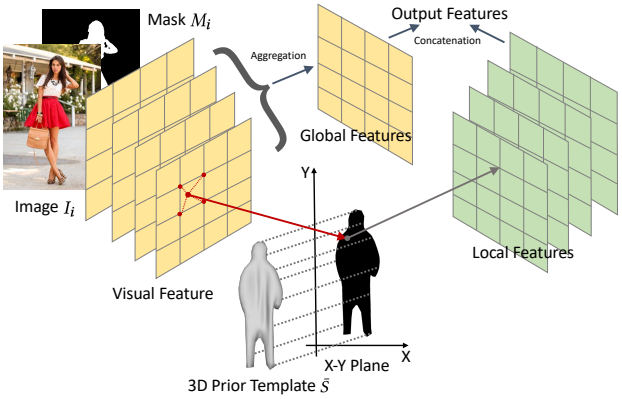

Figure 5: Integration Module (IM). We harness IM to explicitly fuse the prior spatial information from template $\bar{S}$ and the visual feature from (I_i, M_i). The visual feature is the output of the backbone, *e.g.*, HR-Net [75]. As the arrow direction, we leverage the grid sampler to obtain the local feature from the corresponding X-Y of the visual feature. We concatenate global features and local features as final outputs.

(see Figure 4), where $i \in [1, N]$ and N is the number of the samples in the dataset. We do not require any extra 3D annotations. In this work, we explicitly follow the structural causal map in Figure 3 (d) to build the whole pipeline. Generally, the spirit of causality helps us to (1) disentangle the 3D attributes from inherent correlations, such as the ambiguity between the shape and camera encoder; (2) reconsider the causal relation between entities, such as the 3D prior template (prototype) and camera position estimation. Specifically, we deploy four independent encoders, *i.e.*, **shape encoder** E_S, **camera pose encoder** E_C, **illumination encoder** E_A and **texture encoder** E_{UV}. The decoder is based on the differentiable render [9], which does not contain any learnable parameters. Therefore, we can also regard the render as a fixed decoder. Following existing works [24, 26], we also introduce a 3D prototype $\bar{S} \in \mathbb{R}^{|\bar{S}|\times 3}$, which explicitly involves the prior body structure to the network learning. The 3D prior template can be initialized with an arbitrary mesh. Without loss of generality, we apply the elliposid (contains 642 vertices and 1280 faces) to initialize the 3D prior template. Here we set 642 vertices as the default setting to illustrate the proposed approach. During training, we keep updating the prior template $\bar{S}$. When inference, the model deploys the latest 3D prior template (prototype).

3.2. Model Structure

Shape Encoder. We follow the causal map to explicitly introduce 3D prior into the encoder learning. Given a input image-mask pair I_i, M_i and 3D prior template $\bar{S}$, the shape encoder predicts the offsets $\Delta S_i \in \mathbb{R}^{642\times 3}$ for every vertex:

$$\Delta S_i = E_S(I_i, M_i, \bar{S}), \tag{1}$$

$$S_i = \bar{S} + \Delta S_i. \tag{2}$$

The final 3D shape S_i is the sum of the 3D prior template and the 3D per-vertex offsets, and $S_i \in \mathbb{R}^{642\times 3}$. Different from most existing works [17, 26], which independently estimates ΔS_i from the input image I_i and the mask M_i, our shape encoder explicitly takes the prior template into the deformation prediction as $E_S(I_i, M_i, \bar{S})$. The main idea is straightforward, since predicting shape offsets depends on foreknowing the shape prior. We explicitly provide the shape template to help the training. In particular, the shape encoder contains a CNN-based backbone, an integration module (IM) and a fully connected layer (fc).

Integration Module. As shown in Figure 5, we fuse the visual feature from both the input image/mask and the 3D prior template. We harness the integration module (IM) to extract the local visual feature according to the 2D location by projecting the 3D template to the X-Y plane. On the other hand, the global feature is generated by averaging the input visual feature (by global average pooling) and then we repeat the aggregated feature as the original size. The final ΔS_i is predicted by a fully-connected layer on the concatenated feature of both global features and local features.

Camera Pose Encoder. Similarly, the camera pose estimation also depends on the shape prior and the image. However, previous works usually ignore the causal dependency on the shape prior $\bar{S}$. When people infer the object position, *e.g.*, distance, azimuth, and elevation, it is necessary to foreknow the general object shape (general size, general shape, and symmetry to which axis). Therefore, we also deploy a basic convolutional neural network (CNN) followed by an integration module (IM) and fully-connected layers as the camera pose encoder, which can be formulated as:

$$C_i = E_C(I_i, M_i, \bar{S}), \tag{3}$$

where C_i contains four factors, *i.e.*, distance (1-dim), azimuth (1-dim), elevation (1-dim), and X-Y position offset (2-dim). "dim" is dimension. The distance is also formulated as object scale in other works [26], while the azimuth is called as the rotation degree. We notice that several existing works [17] do not include X-Y position, since they assume that the object is in the center, but we find that including X-Y position prediction actually improves the camera robustness during both training and testing. (Please see the discussion on camera attribute distribution in experiment.)

Illumination Encoder. The illumination encoder is to regress the illumination direction and strength, which can be simply formulated as a 9-channel Spherical Harmonics coefficient [6]. Therefore, we adopt a basic convolutional neural network followed by a 9-channel fully-connected layer to predict the illumination vector from the input image-mask pair:

$$A_i = E_A(I_i, M_i). \tag{4}$$

Texture Encoder. In this work, we do not predict the color for every vertex. We follow existing works [20, 24] to learn a texture flow as the UV map by a U-Net structure [49]. Given the input image I_i and the corresponding foreground mask M_i, we first predict the texture flow and then map the color according to the spatial location.

$$UV_i = E_{UV}(I_i, M_i). \tag{5}$$

Decoder (Render). Finally, the decoder, *i.e.*, render, can reconstruct the 3D mesh with color by simply combining the shape S_i and UV_i. If we want to re-project the mesh to the 2D space, we further need the camera pose C_i and the illumination direction A_i. Therefore, the reconstructed image $\hat{I}_i$ can be written as:

$$\hat{I}_i = \pi(\phi(S_i, UV_i), C_i, A_i), \tag{6}$$

where ϕ is the function to colorize the 3D mesh S_i with the UV map UV_i. π denotes the projection function mapping the mesh through camera parameters C_i with the illumination A_i. As a side product, we can also obtain the reconstructed foreground mask $\hat{M}_i$ during projection. We note that both ϕ and π are based on the physical mapping, so there do not contain any learnable parameters.

3.3. Optimization Objectives

Image Reconstruction Loss. As shown in the right part of Figure 6, we calculate the pixel level l_1 loss of the foreground area between the reconstructed image and the input:

$$\mathcal{L}_{img} = \mathbb{E}[||I_i \odot M_i - \hat{I}_i \odot \hat{M}_i||_1], \tag{7}$$

where $\odot$ denotes element-wise multiplication, and $\mathbb{E}$ denotes the expectation. $\hat{I}_i$ and $\hat{M}_i$ are the reconstructed image and mask projected from the 3D mesh. We note that the $\mathcal{L}_{img}$ focuses on the low-level input. Sometimes the generation quality is good but with small position shifts. To further ensure the generation quality from high-level activations, we also introduce the adversarial loss:

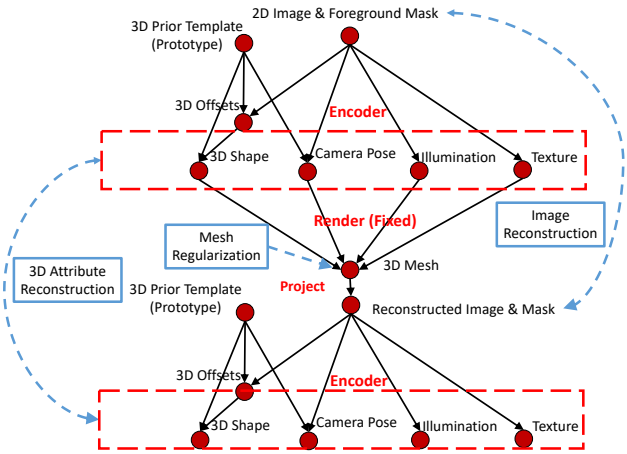

Figure 6: Optimization Objectives. Here we show three kinds of losses on the causal map, which are the image reconstruction loss, the attribute reconstruction loss and the mesh regularization.

$$\mathcal{L}_{adv} = \mathbb{E}[logD(I_i \oplus M_i) + log(1 - D(\hat{I}_i \oplus \hat{M}_i))], \tag{8}$$

where $\oplus$ means concatenation. For instance, $I_i \oplus M_i$ is a 4-channel input. D denotes a multi-layer discriminator to classify whether the input is real or generated from our render. In practice, we adopt a basic WGAN structure [1] as the discriminator. Otherwise, we introduce the IoU loss to compare the overlapping area between the generated mask with the ground-truth input mask:

$$\mathcal{L}_{IoU} = \mathbb{E}[1 - \frac{M_i \cap \hat{M}_i}{M_i \cup \hat{M}_i}]. \tag{9}$$

Attribute Reconstruction Loss. As shown in Figure 6 left part, we also conduct the 3D attribute reconstruction to ensure that the encoder and the decoder are self-consistent:

$$\mathcal{L}_{att} = \mathbb{E}[||S_i - E_S(\hat{I}_i, \hat{M}_i, \bar{S})||_1] + \mathbb{E}[||C_i - E_C(\hat{I}_i, \hat{M}_i, \bar{S})||_1] + \mathbb{E}[||A_i - E_A(\hat{I}_i, \hat{M}_i)||_1] + \mathbb{E}[||UV_i - E_{UV}(\hat{I}_i, \hat{M}_i)||_1]. \tag{10}$$

The 3D attributes predicted from the reconstructed image $\hat{I}_i$ should be the same as the predicted attribute from I_i.

Mesh Regularization. (1) Laplacian loss [64] is a regularization to prevent self-intersection of mesh faces. It encourages adjacent vertices to move in the same direction, consequently, avoiding the local part of the mesh producing outrageous deformation. For each vertex position p in the mesh shape S_i, the laplacian coordinate is $\delta_p = p - \sum_{k \in K(p)} \frac{k}{||K(p)||}$, where $K(p)$ is the neighbor vertices of p with connected edges. Specifically, the laplacian loss can be defined as $\mathcal{L}_{lpl} = \mathbb{E}[||\hat{\delta_p} - \delta_p||_2^2]$, where $\hat{\delta_p}$ and δ_p are laplacian coordinates of a vertex before and after the updation respectively; (2) Flatten loss is another regularization for keeping faces from intersecting [64]. The cosine of the angle between two adjacent faces is calculated. The flatten loss is defined as $\mathcal{L}_{flat} = \mathbb{E}[(cos(\Delta\theta_i) + 1)^2]$, where $\Delta\theta_i$ is the angle between two adjacent faces. The angle around $180°$ implies a smooth mesh surface; (3) Symmetry loss constrains mesh deformations to be reflectional symmetric in the depth [60]. It can be expressed as

$\mathcal{L}_{sym} = \mathbb{E}[||Z(p) + Z(\tilde{p})||_1]$, where Z denotes the depth of the vertex and $\tilde{p}$ is the reflected vertex of p; (4) Deformation loss [20, 26] is a regularization to prevent the mesh from deforming excessively and facilitate the average shape learning: $\mathcal{L}_{deform} = \mathbb{E}[||\Delta S||_2]$.

Total Loss. We train four encoders and the discriminator to optimize the total objective, which is a weighted sum of above-mentioned losses:

$$\begin{aligned}\mathcal{L}_{total} &= \lambda_{rec}(\mathcal{L}_{img} + \mathcal{L}_{IoU}) + \lambda_{att}\mathcal{L}_{att} + \lambda_{adv}\mathcal{L}_{adv} \\ &+ \lambda_{reg}(\mathcal{L}_{sym} + \mathcal{L}_{deform} + \lambda_{lpl}\mathcal{L}_{lpl} + \lambda_{flat}\mathcal{L}_{flat}).\end{aligned} \tag{11}$$

In practice, we refer to existing works [17,26,64] and empirically set $\lambda_{rec} = 2$, $\lambda_{att} = 1$, $\lambda_{adv} = 1 \times 10^{-5}$, $\lambda_{reg} = 0.1$, $\lambda_{lpl} = 0.1$, and $\lambda_{flat} = 0.01$.

3.4. Optimization Strategy

Following the causality-aware design, two loops are introduced during optimization as the causal "intervention" tools. While estimating one cause, the relations between the outcome and the other "colliders" are cut off. In this way, the **encoder loop** helps to disentangle the correlated relations between the four 3D attributes S_i, UV_i, C_i, A_i, while the **prototype loop** separates the prototype updating $\bar{S}$ from the shape offset estimation ΔS_i.

Encoder Loop. During the implementation, we notice one main challenge is simultaneously optimizing the four encoders. The problem mainly lies in the image reconstruction loss. For instance, even if three out of four encoders provide correct prediction, the rest provides the wrong attribute, such as incorrect shape offsets. All four encoders are penalized equally. This is one typical "collider" case in causality. Therefore, one straightforward idea is to train one encoder (predict one cause) while fixing the other three encoders (control other causes). In this way, we can effectively penalize the target encoder. In particular, we adopt one expectation-maximization loop, which also is an "invention" tool in causal learning. For example, we fix the three encoders, *e.g.*, E_c, E_A, E_{UV}, to cut the arrows from inputs to three attributes, *i.e.*, C_i, A_i and UV_i. Only the shape attribute S_i still keeps the dependency from the input image-mask pair. Hence, when the loss is back-propagating, only the shape encoder is penalized. In this way, we disentangle four encoders not only in the forward passing design (independent encoder weights) but also in the loss of back-propagation.

Prototype Loop. We also observe a common ambiguity between prototype updating and shape estimation. The problem is mostly due to Eq. 2. Since it is an addition equation, during gradient back-propagation, $\bar{S}$ and ΔS_i receive the punishment equally. It is hard to distinguish $\bar{S}$ from ΔS_i. Therefore, we adopt the causal invention tool, *i.e.*, Expectation-Maximization, again. During training, we fix the prototype (control one cause) and maximize the shape offsets likelihood (predict another cause). After every training epoch, we leverage the mean shape offsets to update $\bar{S} = \bar{S} + \mathbb{E}[\Delta S]$. In practice, different from existing works (*e.g.*, two-stage training [26]), we adopt a linear warming-up strategy [12] to update prototype slowly in the early epochs and harness the exception handling by clipping extreme deformations. In this way, we disentangle the prototype updating from the shape offsets estimation and learn the model in one go.

Table 2: Comparison with two off-the-shelf template-based methods on the human clothing ATR dataset. Since no texture mapping is contained in the template-based methods, we only compare MaskIoU (%), which reflects the "2D → 3D → 2D" reconstruction quality on the unseen test set.

Methods	HMR [19]	ROMP [55]	Ours
MaskIoU (%) ↑	69.7	70.3	81.1

4. Experiment

We evaluate the proposed approach on two fashion datasets, *i.e.*, ATR [28] and Market-HQ [79], and a widely-used bird dataset CUB [62]. Since there are no ground-truth 3D meshes, we follow existing works [17] and adopt 2D metric, *i.e.*, FID [15], SSIM [66], and MaskIoU, to evaluate the "2D → 3D → 2D" process. FID compares the distribution of two sets of images. We denote the 3D reconstruction results as FID_{recon}, the generated image with different viewpoints as FID_{novel}, following [17]. For Market-HQ, we also report FID_{90} by comparing generated side-view images with real images. Please see **supplementary material** for dataset preparation, structure and training details.

4.1. Quantitative Experiments

Comparison with Template-based Methods. We first compare with the off-the-shelf template-based methods [19, 55] in Table 2. This line of methods is based on the body template with great structure robustness, but is not well scalable to the non-rigid clothing. Since no texture mapping function is built in the template-based methods, we focus on comparing MaskIoU (%), which reflects the "2D → 3D → 2D" shape reconstruction quality. We observe that the proposed method achieves a higher MaskIoU score of 81.1% on the test set. The result is also consistent with the visualization in Figure 1. For clothing reconstruction, the proposed method is more scalable than the template-based methods, covering more regions of interest.

Comparison with Single-image Reconstruction Methods. As shown in Table 3, we compare the proposed method with other state-of-the-art approaches [2,6,17,20,25,26,74] on CUB. MeshInversion [77] deploys test time optimization, so here we do not include it. Among the existing works, SMR [17] has achieved the high-fidelity reconstruction and novel-view generation performance. In contrast, the proposed method yields a better reconstruction performance (81.8% MaskIoU and 83.5% SSIM). At the

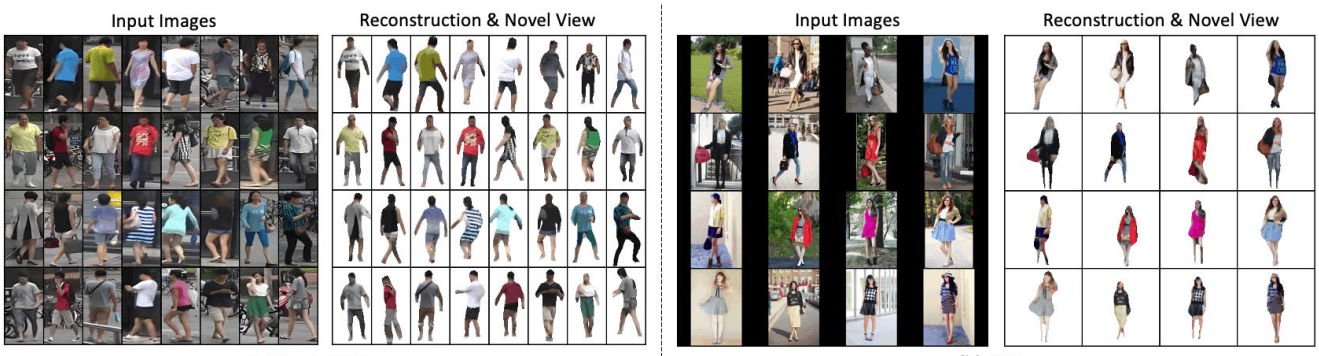

Figure 7: Novel-view 3D clothing generation from single images on the unseen test set of Market-HQ and ATR. (Please open the paper in Adobe Reader to see the mesh rotation.) Here we gradually "Do" / change the camera azimuth degree to render the human.

Table 3: Comparison with other single-image reconstruction methods on the CUB bird dataset. MaskIoU (%) and SSIM (%) reflects the front-view reconstruction quality on the unseen test set, while FID_{novel} compares the distribution difference between generated images from novel views and the original dataset.

Methods	MaskIoU (%) ↑	SSIM (%) ↑	FID_{novel} ↓
View-gen [2]	61.7	-	70.3
ShSMesh [74]	70.7	-	161.0
CMR [20]	73.8	44.6	115.1
UMR [26]	73.4	71.3	83.6
DIB-R [6]	75.7	-	-
ACMR-vid [25]	77.3	-	-
SMR [17]	80.6	83.2	79.2
Ours	**81.8**	**83.5**	**63.5**

Table 4: Ablation Study on Market-HQ and ATR. "No IM" denotes that we remove the integration module. We observe that although "No Encoder Loop" leads the model to over-fit the front-view reconstruction FID_{recon}, the side-view performance FID_{90} is extremely poor. In contrast, our full model takes a balance point between reconstruction, *i.e.*, MaskIoU, and novel-view generation, *i.e.*, FID_{novel}. (Considering the majority of ATR test set is close-frontal view, we do not report FID_{90} on ATR.)

Methods	Market-HQ					ATR			
	MaskIoU (%) ↑	SSIM (%) ↑	FID_{recon} ↓	FID_{novel} ↓	FID_{90} ↓	MaskIoU (%) ↑	SSIM (%) ↑	FID_{recon} ↓	FID_{novel} ↓
SMR†	81.0	66.1	23.6	60.0	120.5	78.5	72.9	38.5	76.7
No IM	72.0	56.4	44.6	72.5	107.7	77.3	72.4	43.0	81.6
No Prototype Loop	82.6	65.8	21.9	47.2	104.1	80.7	71.9	37.3	72.2
No Encoder Loop	82.9	**67.9**	**17.4**	49.2	176.0	76.7	71.6	**33.1**	67.0
Ours	**83.4**	66.3	21.5	**46.7**	**93.3**	**81.1**	**72.6**	35.9	**66.8**

†: For a fair comparison, we re-implement SMR with the same backbone as ours and enable XY position prediction.

meantime, for novel-view generation, our method also has achieved 63.5 FID_{novel}, surpassing SMR by a clear margin. Similarly, based on the same backbone, our method also surpasses SMR on Market-HQ and ATR (see Table 4).

4.2. Qualitative Experiments

Reconstruction and Novel-view Results. As shown in Figure 7, we reconstruct the person with non-rigid clothing. We could observe that the model not only successfully learns legs and arms, but also captures non-rigid objects, including hair, dress, and handbag. CUB results are in Fig. 10.

Exchanging Clothing. Inspired by 2D GAN-based work [81], we also show the result of changing the texture of any two persons but with a 3D mesh manner (see Figure 9). In particular, we apply the shape encoder and the texture encoder to extract the shape S_i and the UV texture map UV_j, respectively. Then we deploy the render to generate the new mesh based on S_i and UV_j. The first row and the first column are the input RGB images. The rest is the projected results of the new 3D meshes. We rotate the mesh for better 3D visualization. It verifies the robustness of our method. The learned UV map could be successfully aligned to different human meshes, even though we have not introduced any part annotations during the training process.

Manipulate Camera Attributes. Since four encoders are disentangled, the proposed method could easily manipulate 3D attributes for customization. Besides the rotation (interpolating the azimuth degree), we could also leverage the learned model to change distance, elevation, and XY position. As shown in Figure 8, we could observe that the proposed method successfully disentangles these camera parameters and could control the projected result smoothly.

4.3. Ablation Study and Further Discussion

Does the two expectation-maximization loops help the encoder learning? Yes. As shown in Table 4, we conduct two ablation studies on Market-HQ and ATR. (1) One is to stop the prototype updating, *i.e.*, No Prototype Loop, and we deploy the fixed elliposid as the basic shape. It directly limits the shape deformation, compromising the reconstruction performance. (2) Besides, we also explore training all the encoders simultaneously without the encoder loop as "No Encoder Loop". We observe that the model can easily over-fit the front-view reconstruction quality but it does not perform well in the novel view, especially when we look at the 3D mesh from the side view, *i.e.*, 90°.

Does the integration module work? Yes. We design the integration module to explicitly fuse the prior prototype as local features for learning shape. Removing the integration module, *i.e.*, No IM, leads to a performance drop in both reconstruction and novel-view generation (see Table 4).

Person Re-id. One interesting problem remains whether our learned 3D human model can facilitate downstream tasks, such as person re-id, which intends to match the

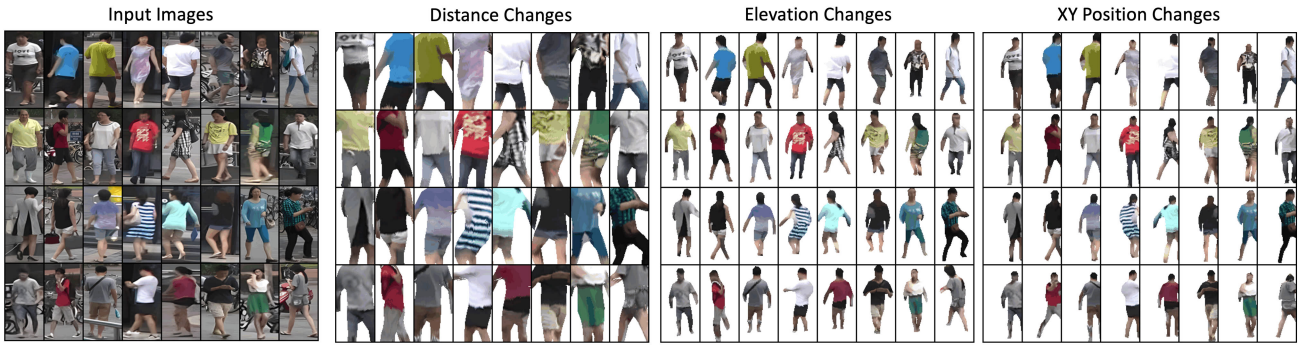

Figure 8: Novel-view 3D clothing generation from single images on the unseen test set of Market-HQ. Here we gradually "Do" / change the camera distance, elevation and XY-position to render the human. (Please open the paper in Adobe Reader to see the movement.)

Figure 9: 3D clothing changing by exchanging the 3D mesh shape and texture. (Please open Adobe Reader to see the movement.)

pedestrian from different viewpoints. We do not intend to pursue state-of-the-art performance, but verify the relative improvement of using the generated data (see Table 5). We observe that our generated 3D-aware images can facilitate re-id representation learning. More details are in **suppl**.

Camera Attribute Distribution. We observe that the camera encoder successfully captures the camera distribution in the Market-HQ test set, which is aligned with the dataset collection setting. Please check **suppl.** for details.

Limitations. There are two faces or two backs of heads on one reconstructed mesh, commonly called Janus Issue. It is because our work is still based on a single image, and the learned model only "sees" one single view of the human. Especially on ATR (most photos are close-frontal faces), it can not learn the 3D prior, *i.e.*, one person only has one face.

Table 5: The re-id performance improvement on Market-1501. We train two competitive backbones. The results suggest that our generated 3D-aware data can further facilitate representation learning.

Methods	Training Set	Rank@1	mAP
ResNet50-ibn [40]	Original	94.63	87.37
	Original+3D	95.07	87.80
HR18-Net [63]	Original	94.74	88.13
	Original+3D	95.43	88.54

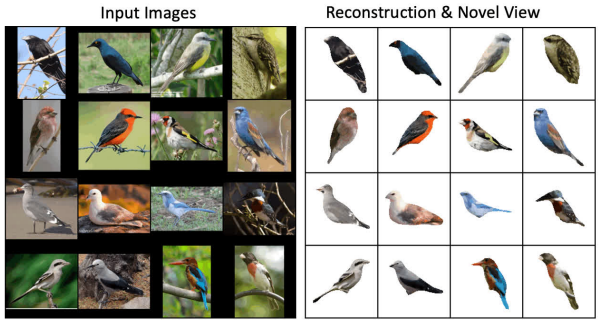

Figure 10: Novel-view 3D bird generation on the test set of CUB.

Therefore, even if we introduce WGAN discriminator [1], it can not provide 3D-aware adversarial loss. The model still largely relies on the symmetric structure to generate the back view. Hence, we think that, in the future, the large-scale multi-view image datasets [16] may help to further solve this limitation upon our work. We also tried simply replacing the ellipsoid with an SMPL template [34], but it fails due to the optimization problem on too many vertices & initial arm position (see **suppl.** discussion).

5. Conclusion

In this paper, we study the 3D clothing reconstruction task to build a "3D Magic Mirror". We follow the spirit of the structural causal map to re-design the output dependency, and leverage two expectation-maximization loops to facilitate the training process. Despite using relatively weak supervision, the proposed method is still competitive with other existing works, and shows great scalability to different non-rigid objects. In the future, we will further explore the applications to multi-modality generation [72] and 3D object re-id [13, 54, 80].

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

[2] Anand Bhattad, Aysegul Dundar, Guilin Liu, Andrew Tao, and Bryan Catanzaro. View generalization for single image textured 3d models. In *CVPR*, 2021.

[3] Benjamin Biggs, Ollie Boyne, James Charles, Andrew Fitzgibbon, and Roberto Cipolla. Who left the dogs out: 3D animal reconstruction with expectation maximization in the loop. In *ECCV*, 2020.

[4] Michael P Chae, Warren M Rozen, Paul G McMenamin, Michael W Findlay, Robert T Spychal, and David J Hunter-Smith. Emerging applications of bedside 3d printing in plastic surgery. *Frontiers in surgery*, 2:25, 2015.

[5] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *CVPR*, 2020.

[6] Wenzheng Chen, Jun Gao, Huan Ling, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *NeurIPS*, 2019.

[7] Yanbei Chen, Shaogang Gong, and Loris Bazzani. Image search with text feedback by visiolinguistic attention learning. In *CVPR*, 2020.

[8] Jianglin Fu, Shikai Li, Yuming Jiang, Kwan-Yee Lin, Chen Qian, Chen Change Loy, Wayne Wu, and Ziwei Liu. Stylegan-human: A data-centric odyssey of human generation. In *ECCV*, 2022.

[9] Clement Fuji Tsang, Maria Shugrina, Jean Francois Lafleche, Towaki Takikawa, Jiehan Wang, Charles Loop, Wenzheng Chen, Krishna Murthy Jatavallabhula, Edward Smith, Artem Rozantsev, Or Perel, Tianchang Shen, Jun Gao, Sanja Fidler, Gavriel State, Jason Gorski, Tommy Xiang, Jianing Li, Michael Li, and Rev Lebaredian. Kaolin: A pytorch library for accelerating 3d deep learning research. https://github.com/NVIDIAGameWorks/kaolin, 2022.

[10] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *3DV*, 2017.

[11] Yixiao Ge, Zhuowan Li, Haiyu Zhao, Guojun Yin, Shuai Yi, Xiaogang Wang, et al. Fd-gan: Pose-guided feature distilling gan for robust person re-identification. *NeurIPS*, 2018.

[12] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv:1706.02677*, 2017.

[13] Ke Han, Yan Huang, Shaogang Gong, Liang Wang, and Tieniu Tan. 3d shape temporal aggregation for video-based clothing-change person re-identification. In *ACCV*, 2022.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017.

[16] Fangzhou Hong, Zhaoxi Chen, Yushi Lan, Liang Pan, and Ziwei Liu. Eva3d: Compositional 3d human generation from 2d image collections. *arXiv:2210.04888*, 2022.

[17] Tao Hu, Liwei Wang, Xiaogang Xu, Shu Liu, and Jiaya Jia. Self-supervised 3d mesh reconstruction from single images. In *CVPR*, 2021.

[18] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *CVPR*, 2019.

[19] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.

[20] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018.

[21] Hiroharu Kato and Tatsuya Harada. Learning view priors for single-view 3d reconstruction. In *CVPR*, 2019.

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

[23] Murat Kocaoglu, Christopher Snyder, Alexandros G Dimakis, and Sriram Vishwanath. Causalgan: Learning causal implicit generative models with adversarial training. *arXiv:1709.02023*, 2017.

[24] Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. In *ICCV*, 2019.

[25] Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Online adaptation for consistent mesh reconstruction in the wild. *NeurIPS*, 2020.

[26] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3d reconstruction via semantic consistency. In *ECCV*, 2020.

[27] Yicong Li, Xiang Wang, Junbin Xiao, Wei Ji, and Tat-Seng Chua. Invariant grounding for video question answering. In *CVPR*, 2022.

[28] Xiaodan Liang, Si Liu, Xiaohui Shen, Jianchao Yang, Luoqi Liu, Jian Dong, Liang Lin, and Shuicheng Yan. Deep human parsing with active template regression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(12):2402–2414, Dec 2015.

[29] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021.

[30] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. *ICCV*, 2021.

[31] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *NeurIPS*, 2018.

[32] Si Liu, Zheng Song, Guangcan Liu, Changsheng Xu, Hanqing Lu, and Shuicheng Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *CVPR*, 2012.

[33] Weiyang Liu, Zhen Liu, Liam Paull, Adrian Weller, and Bernhard Schölkopf. Structural causal 3d reconstruction. In *ECCV*, 2022.

[34] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015.

[35] Yawei Luo, Zhedong Zheng, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Macro-micro adversarial network for human parsing. In *ECCV*, 2018.

[36] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. *NeurIPS*, 2017.

[37] Chengzhi Mao, Augustine Cha, Amogh Gupta, Hao Wang, Junfeng Yang, and Carl Vondrick. Generative interventions for causal learning. In *CVPR*, 2021.

[38] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[39] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *CVPR*, 2021.

[40] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, pages 464–479, 2018.

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *NeurIPS 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[42] Judea Pearl. *Causality*. Cambridge university press, 2009.

[43] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

[44] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[45] Xuelin Qian, Yanwei Fu, Tao Xiang, Wenxuan Wang, Jie Qiu, Yang Wu, Yu-Gang Jiang, and Xiangyang Xue. Pose-normalized image generation for person re-identification. In *ECCV*, 2018.

[46] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018.

[47] Haolin Ren, Zheng Wang, Zhixiang Wang, Lixiong Chen, Shin'ichi Satoh, and Daning Hu. An interactive design for visualizable person re-identification. In *ACM MM*, 2020.

[48] Qibing Ren, Yiting Chen, Yichuan Mo, Qitian Wu, and Junchi Yan. Dice: Domain-attack invariant causal learning for improved data privacy protection and adversarial robustness. In *SIGKDD*, 2022.

[49] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[50] Tao Ruan, Ting Liu, Zilong Huang, Yunchao Wei, Shikui Wei, and Yao Zhao. Devil in the details: Towards accurate single and multiple human parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4814–4821, 2019.

[51] Kripasindhu Sarkar, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. Humangan: A generative model of human images. In *3DV*, 2021.

[52] Zijun Sha, Zelong Zeng, Zheng Wang, Yoichi Natori, Yasuhiro Taniguchi, and Shin'ichi Satoh. Progressive domain adaptation for robot vision person re-identification. In *ACM MM*, 2020.

[53] Keng Hua Sing and Wei Xie. Garden: A mixed reality experience combining virtual reality and 3d reconstruction. In *ACM CHI*, 2016.

[54] Xiaoxiao Sun and Liang Zheng. Dissecting person re-identification from the viewpoint of viewpoint. In *CVPR*, 2019.

[55] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Black Michael J., and Tao Mei. Monocular, one-stage, regression of multiple 3d people. In *ICCV*, 2021.

[56] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *ECCV*, pages 480–496, 2018.

[57] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, 2020.

[58] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NeurIPS*, 2017.

[59] Bing Tian, Yixin Cao, Yong Zhang, and Chunxiao Xing. Debiasing nlu models via causal intervention and counterfactual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11376–11384, 2022.

[60] Shubham Tulsiani, Nilesh Kulkarni, and Abhinav Gupta. Implicit mesh reconstruction from unannotated image collections. *arXiv:2007.08504*, 2020.

[61] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017.

[62] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[63] 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.

[64] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018.

[65] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *International Conference on Computer Vision Workshops (ICCVW)*.

[66] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[67] Hao Wei, Shuhui Wang, Xinzhe Han, Zhe Xue, Bin Ma, Xiaoming Wei, and Xiaolin Wei. Synthesizing counterfactual samples for effective image-text matching. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 4355–4364, New York, NY, USA, 2022. Association for Computing Machinery.

[68] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *ICCV*, 2019.

[69] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J Black. Icon: implicit clothed humans obtained from normals. In *CVPR*, 2022.

[70] Xiangyu Xu and Chen Change Loy. 3d human texture estimation from a single image with transformers. In *ICCV*, 2021.

[71] Guorong Xuan, Wei Zhang, and Peiqi Chai. Em algorithms of gaussian mixture model and hidden markov model. In *ICIP*, 2001.

[72] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning unified representation of language, image and point cloud for 3d understanding. *arXiv:2212.05171*, 2022.

[73] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Disentangled representation learning via neural structural causal models. In *CVPR*, 2021.

[74] Yufei Ye, Shubham Tulsiani, and Abhinav Gupta. Shelf-supervised mesh prediction in the wild. In *CVPR*, 2021.

[75] Changqian Yu, Bin Xiao, Changxin Gao, Lu Yuan, Lei Zhang, Nong Sang, and Jingdong Wang. Lite-hrnet: A lightweight high-resolution network. In *CVPR*, 2021.

[76] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. *NeurIPS*, 2020.

[77] Junzhe Zhang, Daxuan Ren, Zhongang Cai, Chai Kiat Yeo, Bo Dai, and Chen Change Loy. Monocular 3d object reconstruction with gan inversion. In *ECCV*, 2022.

[78] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. Causerec: Counterfactual user sequence synthesis for sequential recommendation. In *SIGIR*, 2021.

[79] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015.

[80] Zhedong Zheng, Xiaohan Wang, Nenggan Zheng, and Yi Yang. Parameter-efficient person re-identification in the 3d space. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2022. doi:10.1109/TNNLS.2022.3214834.

[81] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *CVPR*, 2019.

[82] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. Deephuman: 3d human reconstruction from a single image. In *ICCV*, 2019.

[83] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020.

A. Dataset Preparation

ATR. ATR is a large-scale fashion dataset [28]. It contains $17,700$ human body images, with 18 detailed semantic annotations for every image. The dataset is split into $16,000$ training images, 700 validation images, and $1,000$ test images. In this work, we do not use any part segmentation annotations but only leverage the binary foreground mask as the weak annotation. It is because foreground masks are easy to obtain, which is closer to our application in real-world scenarios.

Market-HQ. We build a high-resolution variable of the Market-1501 dataset [79] based on academic usage. The original Market-1501 is with a relatively low resolution of 128×64. We first apply Real-ESRGAN [65] to upsample the images to 512×256. To acquire the foreground mask, we have tried the off-the-shelf human parsing model [35, 50], but such models suffer from the dataset domain gap. Instead, we apply HMR [19] to obtain the pseudo foreground mask via 2D projection. As a result, Market-HQ contains $12,936$ images of 751 persons for training and $3,386$ test images of other 750 persons. There is no overlapping human appearing in both the training and test set.

CUB-200-2011. CUB-200-2011 is one of the prevailing datasets for single-view 3D reconstruction tasks [62]. It contains images from 200 subcategories of birds, of which $5,994$ are for training and $5,794$ are for testing. 2D foreground masks and keypoints are provided. In this work, we only deploy 2D foreground masks.

B. Implementation Details

Our model consists of four encoders, one render, and one discriminator. Four encoders and the discriminator are implemented based on Pytorch [41], and the differentiable render is from Kaolin [9]. In the following part, we use channel $\times$ height $\times$ width to indicate the size of feature maps. For training Market-HQ, all input images are resized to 128×64 from a high resolution and the image quality is still better than that of images in the original dataset. For ATR, all input image is resized as 160×96. For CUB Bird, all images are padded to a square input and then resized to 128×128. During the evaluation, some generated results are up-sampled to 256×256 for a fair comparison. Random horizontal flipping is used for data augmentation. We apply Adam [22] to optimize the encoders with a mini-batch of 48 and set the basic learning rate as 5×10^{-5}, and $(\beta_1, \beta_2) = (0.95, 0.99)$. The number of training epochs is set as 600. Besides, since the backbone model, *i.e.*, HRNet-Lite [75], has been pre-trained on ImageNet, we do not intend the backbone to update too fast and set the learning rate of the backbone as $0.05\times$ the basic learning rate. We also modified the first layer of HRNet-Lite for 4-channel input (RGB image + foreground mask). In particular, we apply the mean filter weights of the original first convolutional layer to initialize the fourth channel filters. Next, we illustrate the network architecture in detail.

C. Network Architectures

The proposed method consists of the camera encoder E_C, shape encoder E_S, texture encoder E_{UV}, illumination encoder E_A and discriminator D. Following the common practice in GANs, we mainly adopt convolutional layers and residual blocks [14] to construct them. The model can be applied to different input scales. Taking the Market dataset as an example, we utilize the input size as $4 \times 128 \times 64$ for illustration. MMPool denotes the gem pooling [46], which is a weighted sum of average pooling and max pooling.

(1) Shape Encoder: Table 6 shows the architecture of E_s. We apply the backbone network, *i.e.*, HRNet-Lite-v2 [75], to extract the visual feature. Then we apply the Integration Module (IM) to fuse the visual feature and 3D prior template. In the IM block, we concatenate the local feature (2048-dim), global feature (2048-dim), neighbor difference feature (2048-dim), and the template coordinate (3-dim), so the output size of the IM is 6147 dimensions. In practice, we simply subtract the local feature of every vertex with the mean neighbor feature as the neighbor difference feature, where the mean neighbor feature is the mean local feature of the connected vertex. After each convolutional layer, we generally apply the batch normalization layer and LReLU (negative slope set to 0.2). Finally, we obtain 1926-dim ($1926 = 642 \times 3$) output, which is the XYZ biases for 642 vertices as ΔS.

(2) Camera Position Encoder: Table 7 shows the architecture of encoder E_c. We deploy residual blocks and convolutional layers to build the model. For better location estimation, we follow CoordConv [31] and concatenate the grid. ResBlock_half denotes the downsampling block with residual connection from [14]. Since we concatenate the downsampled input, the output size of ResBlock_half is doubled. The final IM simply concatenates the local feature and global feature, so the output size is $576 \times 2 \times 2$. We then flatten the feature and apply three independent MLPs (each contains 2 fully-connected layers) for $azimuths_y$ & $azimuths_x$ (2-dim), elevation & distance (2-dim), and XY-biases (2-dim). The final azimuth is $\arctan(azimuths_y / azimuths_x)$, which is one dimension.

(3) Illumination Encoder: As shown in Table 8, we deploy one convolutional neural network to predict the illumination (9-dim). We apply both the batch normalization layer and LReLU (negative slope set to 0.2) after every convolutional layer.

(4) Texture Encoder: We deploy a U-Net structure in [17] and adopt light-weight ResNet34 [14] as U-Net encoder. We also adopt the BiFPN structure [57] to facilitate the

Table 6: Architecture of the shape encoder E_S.

Layer	Parameters	Output Size
Input	-	$4 \times 128 \times 64$
Backbone (HRNet-Lite-v2)	16 M	$2048 \times 4 \times 2$
IM Module	-	$6147 \times 642 \times 1$
Conv1d	[1×1, 256]	$256 \times 642 \times 1$
Conv1d	[1×1, 3]	$3 \times 642 \times 1$
FC	[1926, 1926]	1926

Table 7: Architecture of the camera position encoder E_C.

Layer	Parameters	Output Size
Input	-	$4 \times 128 \times 64$
AddCoords2d	-	$6 \times 128 \times 64$
Conv	[5×5, 36]	$36 \times 64 \times 32$
ResBlock_half	[3×3, 36 3×3, 36] ×1	$72 \times 32 \times 16$
ResBlock	[3×3, 72 3×3, 72] ×1	$72 \times 32 \times 16$
ResBlock_half	[3×3, 72 3×3, 72] ×1	$144 \times 16 \times 8$
ResBlocks	[3×3, 144 3×3, 144] ×3	$144 \times 16 \times 8$
ResBlock_half	[3×3, 144 3×3, 144] ×1	$288 \times 8 \times 4$
ResBlocks	[3×3, 288 3×3, 288] ×6	$288 \times 8 \times 4$
IM	-	$576 \times 8 \times 4$
MMPool	-	$576 \times 2 \times 2$
MLPs × 3	[2304, 128 128, 2] ×3	128×3 2×3

communication between different layers when decoding. Due to the limitation of the table, we could not show the skip connections and the entire up-sampling process of decoder. Therefore, here we only show the main components in Table 9. The final UV map is symmetric, so we concatenate the mirrored output.

(5) Discriminator: We deploy one convolutional neural network to obtain the real/fake prediction (see Table 10). We only apply LReLU (negative slope set to 0.2) after every convolutional layer.

D. More Ablation Studies

Person Re-id Implementation and Discussion. One interesting problem remains whether our learned 3D human model can facilitate downstream tasks, such as person re-identification (re-id), which intends to match the pedestrian from different viewpoints. To verify this point, we conduct a preliminary experiment via dataset augmentation. In particular, we randomly mix up the texture from two different

Table 8: Architecture of the illumination encoder E_A.

Layer	Parameters	Output Size
Input	-	$4 \times 128 \times 64$
Conv	[5×5, 32]	$32 \times 64 \times 32$
Conv	[5×5, 64]	$64 \times 32 \times 16$
Conv	[5×5, 96]	$96 \times 16 \times 8$
Conv	[5×5, 192]	$192 \times 8 \times 4$
Conv	[5×5, 96]	$96 \times 4 \times 2$
MMPool	-	$96 \times 1 \times 1$
FC	[96, 48]	48
FC	[48, 9]	9

Table 9: Architecture of the texture encoder E_{UV}.

Layer	Parameters	Output Size
Input	-	$4 \times 128 \times 64$
Conv (ResNet-34)	[7×7, 64]	$64 \times 64 \times 32$
MaxPooling		$64 \times 32 \times 16$
ResBlock (ResNet-34)	[3×3, 64 3×3, 64] ×3	$64 \times 32 \times 16$
ResBlocks (ResNet-34)	[3×3, 128 3×3, 128] ×4	$128 \times 16 \times 8$
ResBlocks (ResNet-34)	[3×3, 256 3×3, 256] ×6	$256 \times 8 \times 4$
ResBlock (ResNet-34)	[3×3, 512 3×3, 512] ×3	$512 \times 4 \times 2$
Conv	[3×3, 256]	$256 \times 4 \times 2$
ResBlock	[3×3, 256 3×3, 256] ×1	$256 \times 4 \times 2$
Upsample	-	$256 \times 8 \times 4$
Conv	[3×3, 128]	$128 \times 8 \times 4$
ResBlock	[3×3, 128 3×3, 128] ×1	$128 \times 8 \times 4$
Upsample	-	$128 \times 16 \times 8$
Conv	[3×3, 64]	$64 \times 16 \times 8$
ResBlock	[3×3, 64 3×3, 64] ×1	$64 \times 16 \times 8$
Upsample	-	$64 \times 32 \times 16$
Conv	[3×3, 64]	$64 \times 32 \times 16$
ResBlock	[3×3, 64 3×3, 64] ×1	$64 \times 32 \times 16$
Upsample	-	$64 \times 64 \times 32$
Conv	[3×3, 32]	$32 \times 64 \times 32$
ResBlock	[3×3, 32 3×3, 32] ×1	$32 \times 64 \times 32$
Upsample	-	$32 \times 128 \times 64$
Conv	[3×3, 2]	$2 \times 128 \times 64$
Tanh	-	$2 \times 128 \times 64$
Project	Grid Sampler	$3 \times 128 \times 64$
Concat	Flipping	$3 \times 256 \times 64$

Table 10: Architecture of the discriminator D.

Layer	Parameters	Output Size
Input	-	$4 \times 128 \times 64$
Conv1	$\begin{bmatrix} 1\times1, 16 \end{bmatrix}$	$16 \times 128 \times 64$
Conv2	$\begin{bmatrix} 3\times3, 16 \end{bmatrix}$	$16 \times 128 \times 64$
Conv3	$\begin{bmatrix} 3\times3, 32 \end{bmatrix}$	$32 \times 64 \times 32$
Conv4	$\begin{bmatrix} 3\times3, 32 \end{bmatrix}$	$32 \times 64 \times 32$
Conv5	$\begin{bmatrix} 3\times3, 48 \end{bmatrix}$	$48 \times 32 \times 16$
Conv6	$\begin{bmatrix} 3\times3, 48 \end{bmatrix}$	$48 \times 32 \times 16$
Conv7	$\begin{bmatrix} 3\times3, 64 \end{bmatrix}$	$64 \times 16 \times 8$
Conv8	$\begin{bmatrix} 3\times3, 64 \end{bmatrix}$	$64 \times 16 \times 8$
Conv9	$\begin{bmatrix} 3\times3, 64 \end{bmatrix}$	$64 \times 8 \times 4$
Conv10	$\begin{bmatrix} 3\times3, 64 \end{bmatrix}$	$64 \times 8 \times 4$
Conv11	$\begin{bmatrix} 3\times3, 64 \end{bmatrix}$	$64 \times 4 \times 2$
Conv12	$\begin{bmatrix} 3\times3, 64 \end{bmatrix}$	$64 \times 4 \times 2$
Conv13	$\begin{bmatrix} 3\times3, 48 \end{bmatrix}$	$48 \times 2 \times 1$
Conv14	$\begin{bmatrix} 3\times3, 32 \end{bmatrix}$	$32 \times 2 \times 1$
Conv15	$\begin{bmatrix} 1\times1, 1 \end{bmatrix}$	$1 \times 2 \times 1$
Mean	-	1

Table 11: Architecture of the IM Module.

Data	Layer	Parameters	Output Size
Input Feature		-	$2048 \times 4 \times 2$
Template		-	$\mathbf{3} \times 642 \times 1$
Local	Grid Sampler	-	$\mathbf{2048} \times 642 \times 1$
Global	Pooling	-	$2048 \times 1 \times 1$
	Repeat	-	$\mathbf{2048} \times 642 \times 1$
Neighbor (Optional)	Matrix Multiplication	- -	$\mathbf{2048} \times 642 \times 1$
Output	Concat	-	$6147 \times 642 \times 1$

real identities to form one new virtual pedestrian identity class, and then render five projections of the 3D mesh from -60°, -30°, 0°, 30°, and 60° rotation. As a result, we obtain 77,566 training images of 12,991 identities (751 real identities + 12,240 virtual identities), which enlarge about 6 times images compared with the original Market training set. We train the model with the common setting, such as dropout, random erasing, and horizontal flipping [56, 83]. The final feature dimension is set as 2048 instead of 512 to preserve more visual clues. For a fair comparison, we try our best to tune the baseline and select the most competitive hyper-parameter to report a reliable and competitive baseline result (see Table 5). For the ResNet50-ibn baseline, we train the model with a learning rate of 0.004, batch size of 16, and dropout of 0.75. Similarly, for the HRNet-w18 baseline, we train the model with a learning rate of 0.005, batch size of 16, and dropout of 0.75. On the other hand, since we add many inter-class variants via 3D generation, we train the baseline on our generated dataset (Original+3D) with a smaller dropout rate of 0.2. We also set a larger batch size for faster training. For ResNet50-ibn, we train the model on the generated data with a learning rate of 0.045, and batch-size of 192. Since the parameter of HRNet-w18 is larger than ResNet50-ibn, due to the GPU memory limitation, we set batchsize as 160 with a learning rate of 0.055.

Camera Attribute Distribution. We observe that the camera encoder successfully captures the camera distribution in the Market-HQ test set. As shown in Figure 11, most samples are $2 \sim 4$ unit distances from the camera, and most persons are in the center of the figure with 0 X-Offsets and 0 Y-Offsets. Most samples appear in 0°, 180° or -180°, which means that most people are facing towards or backward to the camera. It is aligned with the dataset setup since cameras are set up in front of the supermarket entrance or exit. The elevation of most samples is from -10 to 10, which is also aligned with the data collection setup, *i.e.*, six horizontal-view cameras. Besides, we also show the distribution of mean shape offset ΔS. We observe that most deformations are relatively small since we have introduced the 3D prior template. In a summary, the learned attribute statistics verify that we disentangle the camera hyper-parameter, *e.g.*, scale changes and position offsets, from the shape encoder.

SMPL Initialization. Actually, our method is compatible with SMPL initialization, but it is worth noting that we still need to carefully consider several engineering problems. (1) Directly using SMPL as a general prototype? We failed. Most people in our datasets are walking with arm close to their bodies. The model does not converge, since the rising arm is in the canonical SMPL model. (2) Why not use SMPL model estimation for every individual people instead of a global prototype? Yes. It is a great idea. We apply the state-of-the-art ROMP [55] to obtain the SMPL mesh for every image as the initial shape. However, another problem arises. The SMPL model contains too many vertexes (12,943) than our basic ellipsoid (642). It also arises the optimization problem during training. The model also does not converge. In the future, we would like to consider other down-sampling strategies. (3) In the future, we may also try LBS and inverse LBS functions like [16] to conduct canonical mapping, and it may align the representation to deal with the optimization problem. However, these techniques may be beyond our work. Therefore, we leave them as future work.

About EMA. We do not report the result with the weight moving average for a fair comparison with other methods in the main paper. EMA [58] actually can further boost our performance, and we conduct EMA for the last 100 training epochs. As shown in Table 12, there are significant improvements in SSIM and MaskIoU, while FID is fluctuating. Actually, it is close to our observation. No matter whether we apply EMA, we observe that the visual results are still close to the ones without EMA. EMA successfully

Table 12: Ablation Study of EMA on Market-HQ and ATR.

Methods	Market-HQ					ATR			
	MaskIoU (%) ↑	SSIM (%) ↑	FID_{recon} ↓	FID_{novel} ↓	FID_{90} ↓	MaskIoU (%) ↑	SSIM (%) ↑	FID_{recon} ↓	FID_{novel} ↓
Ours	83.4	66.3	21.5	46.7	93.3	81.1	72.6	35.9	66.8
Ours +EMA	87.1	74.0	20.0	47.3	94.3	84.3	80.3	32.6	65.4

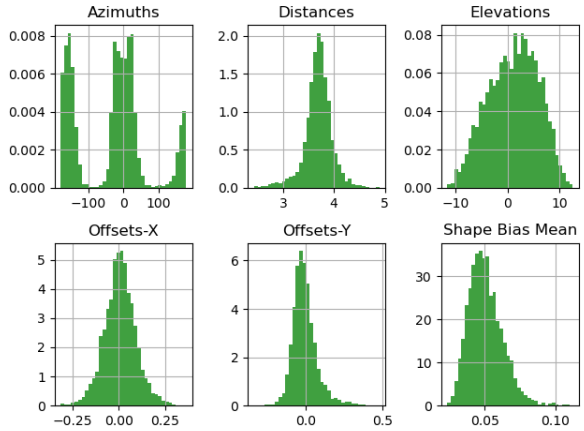

Figure 11: Histogram of 3D Camera Attributes C on Market-HQ. Here we show the distribution of azimuths, distances, elevations, Offsets-X and Offsets-Y. Besides, we also provide the distribution of the mean shape offset ΔS over the test set.

replaces some visual changes with a more stable prediction.