Total Selfie: Generating Full-Body Selfies

Supplementary Material

(a) $d = 1$ (b) $d = 1.3$ (c) $d = 1.6$ (d) $d = 1.9$ (e) $d = 10$ (Ground-Truth)

Figure 1. Example of 4 pairs of training data rendered from one textured mesh. The left 4 columns are the input images, and the last column is their ground truth.

1. Face Undistortion

A common problem with face selfies is perspective distortion. This is caused by the camera being too close to the subject, resulting in facial features closer to the camera appearing larger and those farther appearing smaller, thereby creating an unnatural and distorted appearance.

Previous studies have addressed this issue either through single-image optimization [7, 8] or training on a combined dataset of real and unrealistic synthetic images [13]. For test-time efficiency, we follow the idea of large dataset training. This includes two steps: (1) generate high-quality paired dataset with distorted and undistorted face images. (2) Train a network on this dataset.

The goal of the first step is to render a pair of images with small and large camera-subject distance. To implement this, we adopt EG3D [1], a state-of-the-art textured 3D head generation method. EG3D utilizes a random noise vector and camera parameters to generate tri-planes, which can then be employed for volumetric rendering to produce color images and meshes. One straightforward idea for pair generation is to fix the random noise vector and adjust the camera parameters to directly render desired RGB images. However, this is not feasible as EG3D is pre-trained on a dataset with a specific camera-subject distance. Consequently, rendering images with out-of-distribution camera-subject distances results in noticeable artifacts.

Instead, we create a textured 3D head mesh using EG3D and render head images at varying distances using rasterization and the Phong shading model. Specifically, we employ EG3D to create tri-planes, which are utilized to sample the volume, producing a cube with dimensions $H \times W \times C$ containing density and color values. The surface of the head (including background) is then extracted as a mesh using the Marching Cubes algorithm [5]. For each 3D surface vertex, the vertex color is determined by assigning the color value of the nearest point on the cube. With the textured mesh in place, we proceed to render images at varying distances using conventional rendering techniques. The camera rotation matrix is fixed, and only the camera distance d is adjusted. To maintain consistent eye positions across different images of the same mesh, the focal length f is computed based on the camera distance, given by:

$$f = df_0, \tag{1}$$

where $f_0 = 2.9$ represents the pre-defined focal length for rendering images without invalid pixels (*i.e.*, ensuring that all camera rays can hit the mesh) when $d = 1$. We use PyTorch3D to render 4 input images with severe distortion by setting d to 1, 1.3, 1.6, and 1.9. Additionally, a shared ground-truth image is rendered with d set to 10. For better alignment, all rendered images are processed using the FFHQ face alignment technique proposed by [3]. Fig. 1 shows 4 training pairs derived from a single textured mesh. In total, we generate 10,000 textured meshes, each yielding four training pairs, resulting in a dataset comprising 40,000 training pairs.

The next step is to train an undistortion network using the rendered dataset. For this, we adapt an existing method called facevid2vid [9]. This method uses a source image and a driving image to synthesize a talking-head image with appearance and head pose derived from the source and driving images respectively. For our task, we made two modifications: (1) Both the source and driving images are the image with severe distortion, and the output image is the undistorted image, which will be supervised by our rendered ground-truth. (2) Instead of using shared estimators, we use different estimators (same architecture, different weights) to compute driving keypoints. This enables the network to predict the driving keypoints used for undistortion. Finally, the facevid2vid consists of a couple of face

Face Selfie Aligned Face Selfie Undistored Face Selfie

Figure 2. Results of our trained perspective undistortion network. Given a face selfie (left), we first align it (middle), and then correct the perspective distortion (right).

feature extractors that can be applied to any face image regardless of the downstream task. In order to harness this power, we choose to fine-tune the pretrained model on our dataset instead of training from scratch.

During inference, given a face selfie I_f, we initially align it and subsequently utilize the fine-tuned network for perspective undistortion. Fig. 2 shows results of perspective undistortion on the face selfies. Following [9], we set learning rate as 2e-4, and batch size as 8 for training.

2. Implementation Details

We present the implementation details, and the code will be publicly available after acceptance. Following Stable Diffusion [4], all images in our pipeline are square and share a consistent resolution of 512.

2.1. Dataset Generation

We define one training pair as $\{(S', I'_{gt} \cdot M', M'), I'_{gt}\}$, where $S' = \{I'_f, I'_u, I'_\ell, I'_s\}$ is a set of four synthetic selfies for face, upper body, lower body, and shoes respectively. I'_{gt} is the ground-truth full-body image, and M' is the mask indicating the region to be inpainted.

We employ RealisticVision [6] as the pretrained Stable Diffusion with OpenPose ControlNet v1.1 [12] to generate I'_{gt}. The guidance scale is set to 7.5, the denoising step is 20, and the ControlNet scale is 1.0. OpenPose Skeleton images, used for guidance, are detected from a subset

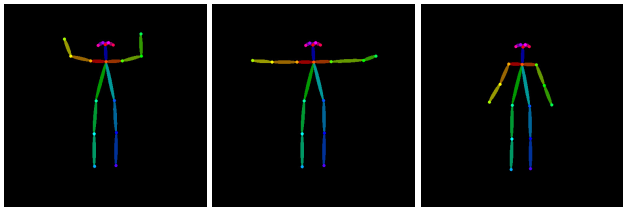


Figure 3. Examples of OpenPose skeleton images we use to generate ground truth full body image.

of the Human Bodies in the Wild dataset [2]. Fig. 3 illustrates three examples of these pose images. The text prompt used is "a [gender], [place], [upper], [lower], [shoes], standing, front-facing, RAW photo, full body shot, 8k uhd, high quality, film grain". Here, [gender] can be man or woman, [place] includes common indoor and outdoor locations such as beach, park, street, restaurant, cafe, shopping mall, *etc*. [upper] comprises shirt, hoodie, sweater, jacket, *etc*., while [lower] includes jeans, leggings, shorts, *etc*., and [shoes] covers sneakers, heels, boots, flats, etc. Additionally, we use CodeFormer [14] to enhance the details of face regions in the generated I'_{gt}.

After obtaining the face-refined I'_{gt}, we utilize the human parsing network [11] to generate the semantic map of I'_{gt}. This map is then employed to extract the bounding box of the person. The bounding box is scaled up following the strategy outlined in [10] to obtain M'.

For the real selfies utilized in detecting typical keypoints for selfie simulation, we gather sets of upper body, lower body, and shoes selfies, each comprising 10 examples. Fig. 4 illustrates one example from each pre-captured selfie category.

In total, we create a training dataset consisting of 39,816 pairs, encompassing diverse individuals, clothing types, poses, and backgrounds.

2.2. Training

We initialize all network weights using the pretrained model provided by Paint-By-Example [10], except for the adapted linear layer L (zero-initialized). For training, we set the learning rate as 1e-5 and batch size as 12. We train the model for 9 epochs, taking around 36 hours on 3 NVIDIA A100 GPUs.

2.3. Automatic Target Pose Selection

We develop an automatic selection strategy to help obtain I_r from the users' photo collection Φ. The selection criteria are based on the similarity between the clothing types in the input selfies and a candidate image in Φ. This is because the more similar the clothing type is, the more accurately the body shape (in this particular type of outfit) can be extracted from I_r.

Upper Body Lower Body Shoes

Figure 4. Examples of in-the-wild selfies used to detect typical keypoints.

(a) Upper Body (b) Lower Body (c) Shoes (d) Ref Photo

Figure 5. Visualization of automatic target pose selection. The top row shows the input selfies and candidate reference photos, and the bottom row shows their segmentation results.

Specifically, we begin by utilizing the pretrained human parsing model [11] to obtain the semantic map of selfies I_u. We filter out semantic labels that occupy less than 21 pixels and labels that do not belong to upper cloth (*e.g.*, pants, face). This results in a set of upper body labels, denoted as P_u. The same process is applied to obtain lower body labels P_ℓ and shoes labels P_s. For a full-body reference photo in Φ, we use the same network to obtain its semantic map. We filter out semantic labels that occupy less than 5 pixels and labels that do not belong to the upper body, lower body, and shoes. The resulting set is denoted as P_r. The matching score of each photo in Φ is computed by $P_r \cap (P_u \cup P_\ell \cup P_s)$. Then we rank the candidate references based on the matching score, with higher scores indicating better matches. Once reference photo I_r is selected, we obatin inpainting mask M using the bounding box of I_r, scaled up by 1.1 times by default.

Fig. 5 (a) to (d) visualizes an example of the semantic map of selfies and the selected reference photo.

2.4. Pose-Guided Generation

For generation, we set $T = 50$ and use DDIM scheduler for denoising. The ControlNet scale is set to 1.0. In cases where the target pose involves spreading arms, the mask M might sometimes become too large, potentially leading to the failure of preserving background content in I_b. To help alleviate this issue, we apply the following strategy during the denoising process. We dilate the foreground mask (the finer mask containing only the human body) by 21 pixels, denoted as $\bar{M}$. Then, at each denoising timestep t, we compute the denoised latent as:

$$z_{t-1} = \begin{cases} z^f_{t-1}, & \text{if } t \leq sT \\ z^f_{t-1} \cdot \bar{M} + z^b_{t-1} \cdot (1 - \bar{M}), & \text{if } t > sT \end{cases}, \tag{2}$$

where z^f_{t-1} is the foreground latent obtained using the selfie-conditioned inpainting model (following the same process discussed in the main paper). z^b_{t-1} is the background latent obtained by adding noise to I_b by $t-1$ steps using DDIM scheduler. We set $s = 0.4$. This enables the generation of details in the surrounding area (*e.g.*, shadows) based on the inpainting model in later timesteps (smaller t), while reasonably preserving background content in the earlier timesteps (larger t).

2.5. Fine-Tuning

We generate a "ground truth" for fine-tuning by resizing and placing a randomly selected selfie image from the set S into the mask region M of the background image I_b. The full-body inpainting prior learned by the trained model is used to determine where and how to place the selected selfie image into the masked I_b.

Specifically, we first generate full body selfie I_n without any pose as condition. This is achieved by using the same process as Pose-Guided Generation but omitting the modified ControlNet. Suppose the upper body selfie I_u is selected for augmentation. We extract the bounding box of the upper body in I_n based on the semantic map of I_n detected by the human parsing network [11]. Then we resize I_u to have the same height as this bounding box. The resize operation keeps the aspect ratio of I_u unchanged to avoid using an image with the wrong scale. Finally, we paste this resized I_u to the masked I_b, ensuring the center of resized I_u and the bounding box are the same.

In practice, we generate 20 different I_n as the candidate pool for augmentation. We then repeat the above augmentation process (resizing and pasting) 200 times, each time with I_n randomly chosen from the candidate pool, resulting in a dataset of 200 augmented images. For fine-tuning, we set the learning rate to 5e-6 and the batch size to 4. The model is fine-tuned for 400 steps, taking around 10 minutes on a single NVIDIA A40 GPU.

2.6. Appearance Refinement

To train the DreamBooth (with two concepts) using only one image for each concept, we need to augment them to avoid overfitting. Specifically, we randomly resize the face selfie I_f from a resolution of 350 to 450 and apply random zero-padding to create the augmented image with a resolu-

tion of 512. The same operation is performed for the shoes selfie I_s, but with resizing resolution from 400 to 500. We generate 50 augmented images for I_f and I_s respectively. Then, we train a DreamBooth with two concepts using these two kinds of augmented images. Specifically, we set the training text prompt for face and shoes as "a sks face" and "a hta shoes" respectively. We set the learning rate as 5e-6, batch size as 4, and fine-tune the RealisticVision Stable Diffusion model for 300 epochs, taking around 5 minutes on a single NVIDIA A40 GPU.

To perform refinement, we use the pretrained human parsing model [11] to obatin the face (shoes) region in I_o, and use SDEdit based on trained DreamBooth to edit the face (shoes). We then paste the cropped, edited image back onto I_o. The same process is applied (using the pretrained Stable Diffusion model) for the hands regions (left and right hands) since hands are often invisible or incomplete in selfies.

3. Experiments

3.1. Dataset

We provide details on the data capture process for our pipeline. In a specific site, the user is requested to capture five square images: face, upper body, lower body, shoes, and background. Firstly, the user takes a face and upper body photos using the front camera, holding the camera with either one or two hands. Subsequently, the user switches to the rear camera to capture the lower body, shoes, and background photos. The entire process usually takes less than 20 seconds.

To obtain the real photo as the "ground truth," we have another person take a full-body photo of the user in a desired pose. This process should ensure that the real photos maintain the same clothing, nearly identical facial expressions, and background.

3.2. Results

We show more results of Total Selfie in Fig. 6. Total Selfie can produce high-quality full-body shots in diverse backgrounds, poses, outfits, and expressions, all while maintaining reasonable shading and composition.

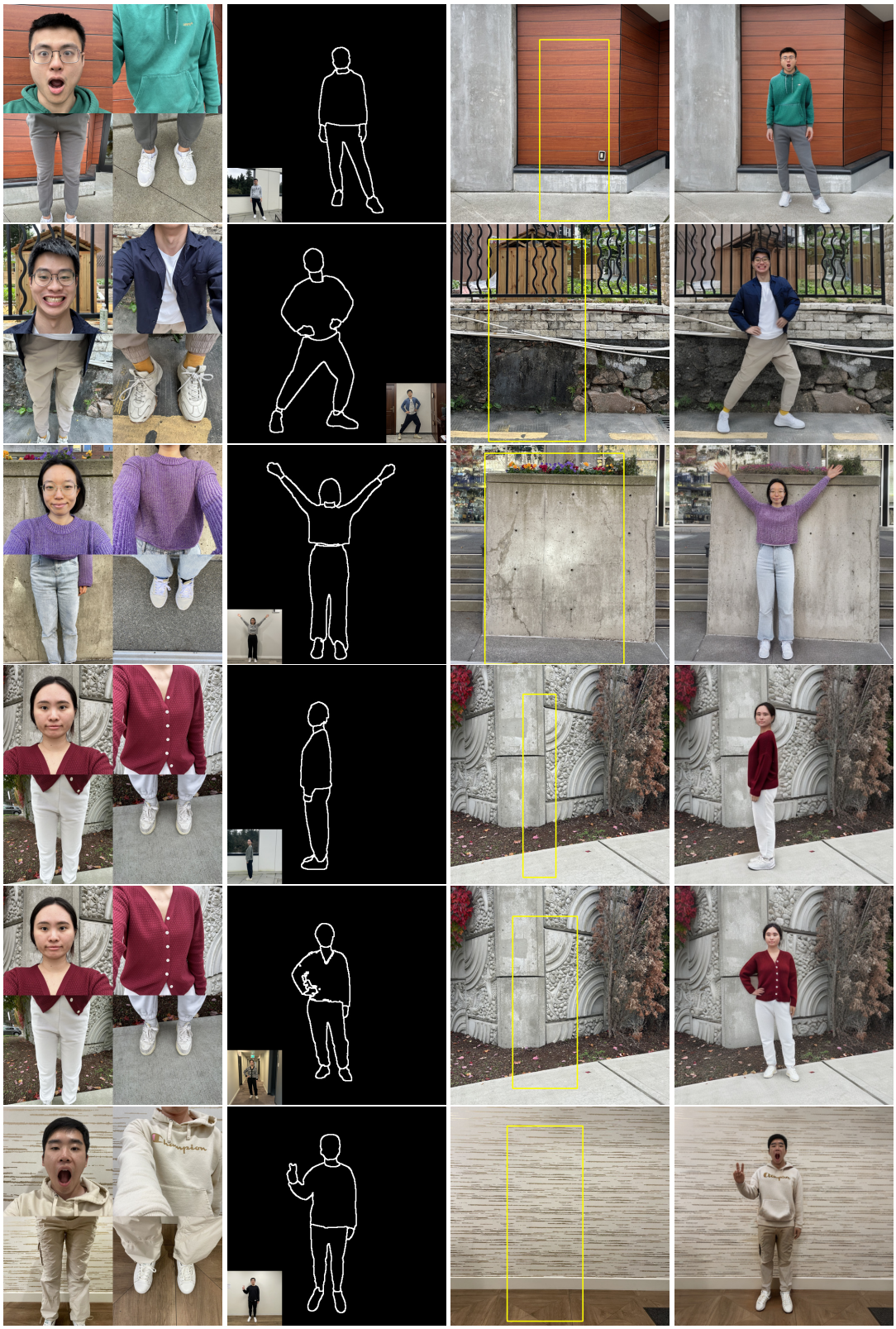
3.3. Ablation Study

Fig. 7 shows additional results of the ablation study, further demonstrating the effectiveness of our final design.

3.4. Baseline Comparison

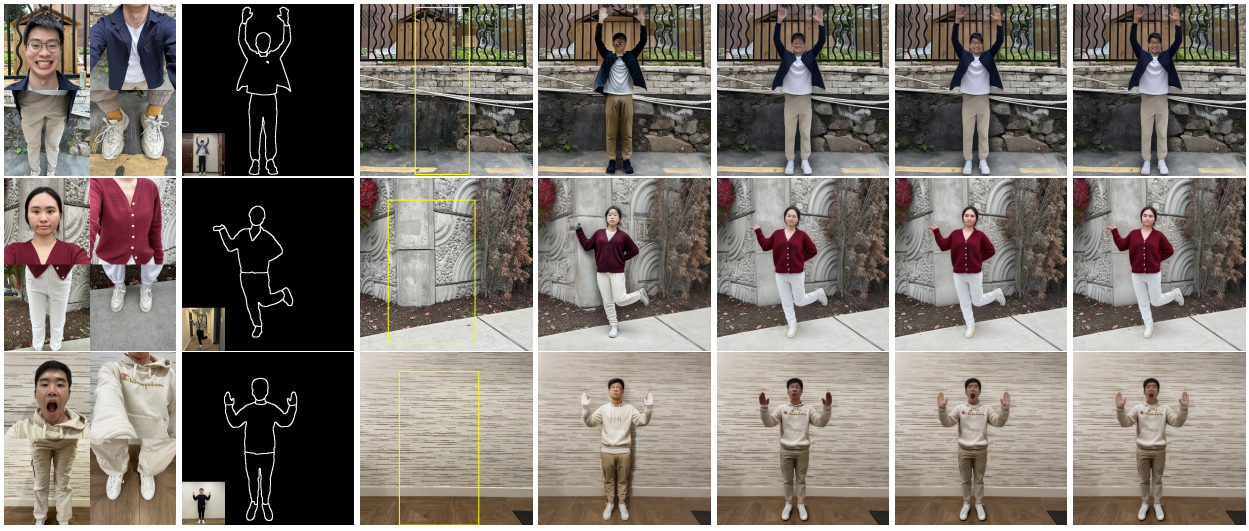
Fig. 8 shows a comparison with all baselines. Total Selfie can produce high-quality full-body shots in diverse backgrounds, poses, outfits, and expressions, all while maintaining reasonable shading and composition.

For the baseline DreamBooth, we use the prompt: "photo of a full body person, [V] face, wearing [X] top, [Y] bottom, [Z] shoes", where tokens $[\cdot]$ are unique identifiers used to train DreamBooth for a specific concept (body part).



(a) Input Selfies (b) Target Pose (c) Masked Background (d) Total Selfie

Figure 6. Results. The second column shows the Canny Edge images detected from reference images (shown as insets). Regions inside yellow box of (c) are the masked regions. Total Selfie generates realistic, full-body images of different individuals with diverse poses and expressions against a variety of backgrounds, while preserving facial expression and clothing.



(a) Input Selfies (b) Target Pose (c) Background (d) Ours-FT-AR (e) Ours-AR (f) Ours-FU (g) Ours

Figure 7. Results for different modules of our pipeline. The Canny Edge image in (b) is detected from the reference image, inset. Regions inside the bounding box (c) are to be inpainted. Generating without fine-tuning and appearance refinement (d) produces an inaccurate outfit and identity. Through fine-tuning, the pipeline (e) generates the correct outfit with reasonable shading but with the wrong identity. Without face undistortion, (f) generates a face with more perspective distortion (*i.e.*, exaggerated facial features), zoom in for details. In contrast, the full pipeline (g) yields high-quality full-body selfies.

(a) Input Selfies (b) Background (c) PBE (d) DisCo (e) LaDI-VTON (f) DreamBooth (g) Ours (h) Real Photo

Figure 8. Qualitative comparison with all baselines. For all methods (except for DisCo), we used the Canny Edge of the real photo as the target pose (inset of (h)). For DisCo, we used OpenPose Skeleton of the real photo as the target pose. Our pipeline clearly outperforms baselines in terms of photorealism and faithfulness (zoom in for details, including faces and shoes). Note that, while the selfies, background image, and real photo were captured in the same session, variations in lighting conditions, auto exposure, white balance, and other factors may result in intensity and color tone differences.

References

[1] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *arXiv*, 2021. 1

[2] Vasileios Choutas, Lea Muller, Chun-Hao P. Huang, Siyu Tang, Dimitris Tzionas, and Michael J. Black. Accurate 3d body shape regression using metric and semantic attribute. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1

[4] Sangyun Lee, Gyojung Gu, Sunghyun Park, Seunghwan Choi, and Jaegul Choo. High-resolution virtual try-on with misalignment and occlusion-handled conditions. In *European Conference on Computer Vision*, pages 204–219. Springer, 2022. 2

[5] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 1

[6] SG_161222. Realistic vision v5.1, 2023. Face and Gesture submission ID 324. Supplied as supplemental material `fg324.pdf`. 2

[7] YiChang Shih, Wei-Sheng Lai, and Chia-Kai Liang. Distortion-free wide-angle portraits on camera phones. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 1

[8] Tan Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zhengyuan Yang, Hanwang Zhang, Zicheng Liu, and Lijuan Wang. Disco: Disentangled control for referring human dance generation in real world. *arXiv preprint arXiv:2307.00040*, 2023. 1

[9] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2

[10] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. *arXiv preprint arXiv:2211.13227*, 2022. 2

[11] Lu Yang, Wenhe Jia, Shan Li, and Qing Song. Deep learning technique for human parsing: A survey and outlook. *arXiv preprint arXiv:2301.00394*, 2023. 2, 3, 4

[12] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 2

[13] Yajie Zhao, Zeng Huang, Tianye Li, Weikai Chen, Chloe LeGendre, Xinglei Ren, Ari Shapiro, and Hao Li. Learning perspective undistortion of portraits. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7849–7859, 2019. 1

[14] Shangchen Zhou, Kelvin C.K. Chan, Chongyi Li, and Chen Change Loy. Towards robust blind face restoration with codebook lookup transformer. In *NeurIPS*, 2022. 2