FEAT: Fashion Editing and Try-On from Any Design

Supplementary Material

A. Collecting Evaluation Dataset

Table 1. Modality summary of the evaluation dataset.

Category	Modality	Count	Source
Tops	Sketch	300	VITON-HD
Tops/Bottoms/Dress	Sketch	600 (200 ea.)	DressCode
Bags/Belts/Shoes/Scarves	Sketch	100 (25 ea.)	DressCode
All	Text	1000	GPT-4V
All	Image	1000	WikiArt, MJ

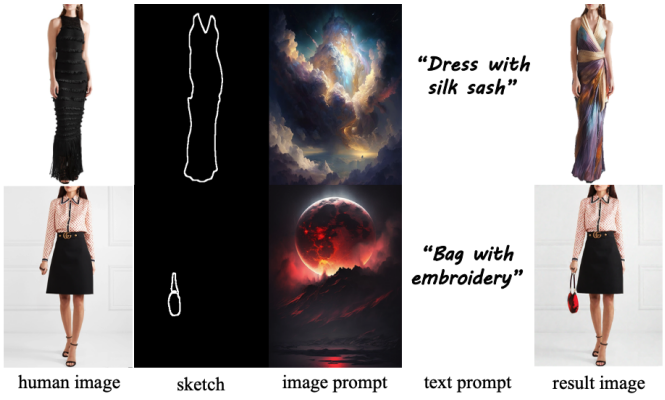

Figure 1. Examples of evaluation dataset.

Existing virtual try-on (VTON) datasets are primarily limited to garments and do not sufficiently cover the multimodal information required for our task. To enable both quantitative and qualitative evaluation across diverse scenarios, we construct a new evaluation dataset with three input modalities: sketch, image, and text. Our dataset construction is based on two representative VTON benchmarks: VITON-HD [4] and DressCode [8]. VITON-HD focuses on tops, while DressCode includes tops, bottoms, and dresses. From these datasets, we selected garment and accessory items and designed a semi-automated annotation framework to extract the corresponding multimodal inputs.

We collected 300 sketches from VITON-HD and 600 from DressCode for garment categories. Additionally, we curated 100 accessory sketches (25 each for bag, belt, shoes, and scarf) from DressCode, resulting in a total of 1,000 sketches used for our experiments. The overall statistics and sources of each modality component in our evaluation dataset are summarized in Tab. 1. The final dataset includes sketch, image, and text conditions for a wide range of garments and accessories. Visual examples are shown in Fig. 1. This dataset serves as a core component of our evaluation, and the detailed construction procedure is described in the following sections.

A.1. Extracting Sketches

Extracting Garment Sketches To extract garment sketches, we first use CATVTON [5], a state-of-the-art VTON model, to synthesize try-on images from person images in the VITON-HD [4] and DressCode [8] datasets. We then apply HumanParsing [7] to segment the clothing regions from the generated results and extract edge maps from the masked regions to obtain the final garment sketches.

Extracting Accessory Sketches We categorize fashion accessories into four groups: bags, belts, shoes, and scarves. Accessory sketches are extracted from full-body images in the DressCode [8] dataset. For each category, we apply HumanParsing [7] to segment the target accessory region. The segmented mask is used with IOPaint to inpaint and remove the accessory naturally. The inpainted image is used as the person input, and the segmented accessory region is used as the sketch condition.

A.2. Curating Style Images and Text Prompts

To incorporate a broad range of design sources, we randomly collect style images from WikiArt [11] and Midjourney [1]. For more precise and diverse control conditions, we generate text prompts using GPT-4V [6] based on the meta-prompting scheme introduced in GPTEval3D [12]. The prompt generation process is guided by three key components: (1) the type of garment item, (2) decorative design elements such as ribbons, embroidery, metal buckles, or chain details, and (3) affective attributes such as "sophisticated", "elegant", or "dreamlike", which appear in approximately 20% of the prompts. These constraints yield prompts with varying levels of complexity and creativity, encompassing both minimal and highly ornamented descriptions. The resulting text conditions span diverse visual and emotional characteristics. Detailed prompting instructions are visualized in Fig. 10.

B. GPT-4V Evaluation

B.1. GPTEval3D Framework

GPTEval3D [12] is a framework originally proposed for evaluating text-to-3D generation models. It leverages GPT-4V to automatically generate meta-prompts and performs pairwise comparisons between two 3D outputs based on user-defined criteria. Unlike traditional quantitative metrics that often rely on single indicators such as text simi-

larity, this approach enables holistic, human-centered evaluation encompassing both visual realism and multimodal alignment. We adopt the core evaluation procedure of GPTEval3D—including meta-prompt construction, GPT-4V-based pairwise comparison, and Elo score computation—but adapt it to the VTON task. Specifically, instead of 3D models, our input comprises:

1. an input person image,
2. an image prompt,
3. a sketch,
4. a text prompt,
5. output i,
6. output j.

The evaluation criteria are also redefined to focus on multimodal consistency, identity preservation, and visual realism.

B.2. Elo Score

What is Elo Score? The Elo rating system is a relative evaluation method originally designed to quantify chess player skill levels. Its core principle assumes that the outcome of a match is determined by the rating difference between two players. For instance, a 400-point rating gap corresponds to roughly a 90% win probability for the higher-rated player. This structure enables robust estimation of relative performance using only pairwise comparison results, even in the absence of absolute ground truth—making it well-suited for generative model evaluation.

Formulation The probability that model i beats model j is defined using the same logistic function as in the traditional Elo rating system:

$$\Pr(\text{“}i \text{ beats } j\text{”}) = \frac{1}{1 + 10^{(\sigma_j - \sigma_i)/400}}. \quad (1)$$

where σ_i denotes the Elo score of model i. Given the number of wins A_{ij} of model i over model j from pairwise comparisons, the Elo scores $\boldsymbol{\sigma}$ are estimated by minimizing the following negative log-likelihood:

$$\boldsymbol{\sigma} = \arg\min_{\boldsymbol{\sigma}} \sum_{i \neq j} A_{ij} \log\left(1 + 10^{(\sigma_j - \sigma_i)/400}\right). \quad (2)$$

This loss function quantifies how well the Elo score differences between models explain the observed win/loss frequencies A_{ij}. Intuitively, a larger score difference should correspond to a higher win probability. The optimization adjusts the Elo scores $\boldsymbol{\sigma}$ to minimize the discrepancy between predicted probabilities and actual pairwise outcomes.

Application to VTON We adopt the GPT-4V-based Elo evaluation framework from GPTEval3D [12], modifying the input structure to suit the VTON task. Each evaluation sample consists of: (1) an input person image, (2) an image prompt, (3) a sketch, (4) a text prompt, and (5), (6) two VTON output images to be compared. GPT-4V is provided with these five images and the text prompt, and is asked to perform a pairwise comparison by answering the question: *“Which of the two virtual try-on results better satisfies the given conditions?”*

B.3. GPT-4V Evaluation Protocol

To assess the quality of multimodal VTON outputs, we define a structured evaluation protocol using GPT-4V. This protocol consists of (i) three evaluation criteria and (ii) a standardized prompt format.

Evaluation Criteria The evaluation criteria are defined as follows:

- **Multimodal Prompt Consistency:** Measures how accurately the generated output reflects the multimodal input conditions. Direct copying of objects or backgrounds from the reference image is considered a failure.
- **Identity Preservation:** Assesses how well non-fashion aspects—such as face, body shape, and skin tone—match the input person image.
- **Realism:** Evaluates overall visual realism based on lighting consistency, texture quality, and absence of artifacts.

Prompt Format GPT-4V receives five input images and is instructed to compare the two VTON results (A and B) according to the three criteria above. An example prompt used during evaluation is shown in Fig. 9.

The output is returned in the following format:

```
Final answer: <criterion 1>
<criterion 2> <criterion 3>
```

Here, the number 1 indicates a preference for A, 2 for B, and 3 denotes a tie.

C. Collecting Garment and Accessory Datasets

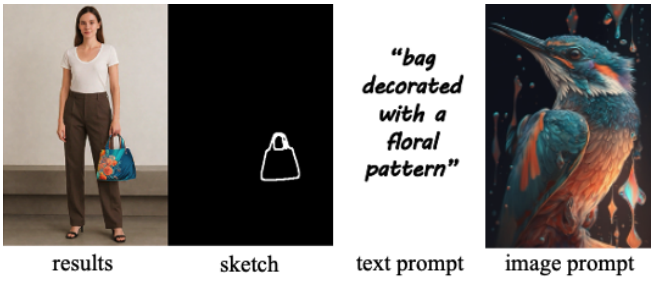

Figure 2. Examples of garment and accessory datasets.

While the DressCode [8] dataset allows for result visualization, it prohibits modification or redistribution as a separate dataset. To overcome its garment-centric limitations, we construct a new multimodal dataset that extends to accessory items and includes sketch, image, and text modalities. Using GPT-4V [6] and the semi-automated annotation

tools described in Sec. A, we collect multimodal input for each item. Specifically, based on the text prompts generated in Sec. A.2, we generate full-body images of human figures wearing the described fashion items via GPT-4V. We then extract the corresponding sketch using the method outlined in Sec. A.1, and randomly sample image prompts from Midjourney. In total, we construct 500 multimodal examples comprising:

- 100 samples each for the garment categories (tops, bottoms, dresses), and
- 50 samples each for the accessory categories (bag, belt, shoes, scarf),

with each item paired with text, sketch, and image inputs. Using these three conditioning modalities, we apply our proposed method, *FEAT*, to generate the final try-on results.

Visual examples of the final dataset are shown in Fig. 2, and the complete dataset will be publicly released.

D. Experimental Details

D.1. Performance Evaluation

Metrics For fair comparison, all outputs are downsampled to the resolution used by MGD [3] (512×384) prior to metric computation. For the Sketch score, we first obtain a mask of the try-on region using HumanParsing [7]. We then extract edge maps from the generated images with a pretrained edge detector (Hugging Face Hub) and compute the Chamfer Distance (CD) [10] against the input sketch within the masked region.

Implementation Details To extract control cues, we use ControlNet-Depth and IP-Adapter from SDXL [9]. Following DreamStyler [2], we adopt depth maps as the default condition in ControlNet. For a fair comparison, the image-scale parameter of all models is fixed at 0.5. In the *OFR* module, the orthogonal projection scale is set to $\alpha = 0.8$ to reliably attenuate residual garments prior to try-on.

D.2. Additional Quantitative Evaluation

Table 2 reports quantitative results under a style-only setting, where the content and style scales are set to 0 and 0.5, respectively, thus excluding content cues and retaining only style information. Similar to the main quantitative results in the paper, *FEAT* consistently outperforms all baselines across both garment and accessory datasets, further demonstrating its overall effectiveness.

D.3. User Study

We additionally conducted a user study under a style-only setting, where content information from the image prompt is removed while style cues are preserved. This experiment evaluates how well the proposed method harmonizes sketch, text, and image-based style signals and whether it can still generate natural and coherent outputs. The study followed the same protocol described in the main paper: 21 participants were presented with 57 comparison cases, and for each case, they selected the most appropriate result based on (i) *Realism* and (ii) *Multimodal Consistency*. Across all three prompt configurations (Sketch+Image, Sketch+Text, and Sketch+Image+Text), *FEAT* achieved the highest selection rate(Fig. 3). Notably, with content removed from the image prompt, our method more reliably leveraged structural cues from the sketch and semantic attributes specified by text. Even when guided solely by style information, *FEAT* preserved high realism and consistency, demonstrating its ability to balance the three modalities and produce natural outputs.

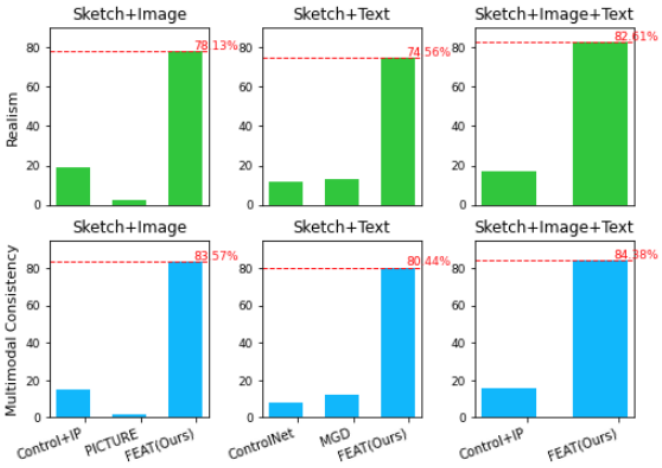

Figure 3. Results of the user study using style information only across various multimodal settings.

E. Method Supplement

E.1. Selective Dual Injection (SDI)

The image prompt i is injected through the decoupled cross-attention mechanism of the IP-Adapter [13]. Specifically, after passing through the CLIP image encoder, the extracted image features are injected into the content block and style block. To achieve this, we introduce an additional cross-attention layer exclusively within the block to integrate image features. Given the image feature $\phi(i)$ extracted by the CLIP image encoder, the output of the newly added cross-attention layer, Z', is computed as follows:

$$Z' = Softmax\left(\frac{QK'^{\top}}{\sqrt{d}}\right)V', \tag{3}$$

where $K' = \phi(i) \cdot W'_k$, $V' = \phi(i) \cdot W'_v$ represent the key and value extracted from the image features. The query Q remains identical to the one used in the cross-attention operation with text features. Subsequently, the output from the cross-attention operation with image features is simply

Table 2. Quantitative comparisons on garment and accessory datasets across various multimodal settings.

	Garment Dataset					Accessory Dataset				
	GPT-4V ↑	Sketch ↓	Image ↑	Text ↑	Human ↑	GPT-4V ↑	Sketch ↓	Image ↑	Text ↑	Human ↑
Sketch + Image										
ControlNet+IP-Adapter	1080.56	10.42	0.33	-	17.22%	1091.69	36.83	0.21	-	16.67%
PICTURE	778.09	14.54	0.31	-	2.20%	784.17	125.37	0.19	-	1.6%
FEAT (ours)	**1232.03**	**4.67**	**0.37**	-	**80.59%**	**1228.17**	**12.41**	**0.24**	-	**81.18%**
Sketch + Text										
ControlNet	902.70	8.09	-	27.81	7.10%	1149.35	6.14	-	23.58	13.05%
MGD	1027.48	7.32	-	28.35	18.43%	814.02	38.31	-	22.65	6.42%
FEAT (ours)	**1148.15**	**5.16**	-	**29.12**	**74.47%**	**1210.31**	**4.00**	-	**25.18**	**80.52%**
Sketch + Image + Text										
ControlNet+IP-Adapter	896.67	9.04	0.30	27.48	17.22%	892.57	13.25	0.17	24.91	15.08%
FEAT (ours)	**1103.32**	**4.39**	**0.38**	**28.80**	**82.78%**	**1107.41**	**3.87**	**0.23**	**25.26**	**84.92%**

added to the output from text-based cross-attention. The final output Z is thus defined as follows:

$$Z = \mathrm{Attn}(Q, K, V) + \mathrm{Attn}(Q, K', V'). \tag{4}$$

where (K', V') denote image-conditioned keys/values in a general form. In practice, we use block-specific image features, yielding (K'_c, V'_c) for content blocks and (K'_s, V'_s) for style blocks. Given that the two forms of cross-attention operate independently, the contribution of image features to the overall attention output can be explicitly controlled. Recognizing that each block serves a distinct functional role, we allow block-wise modulation of cross-attention strength. To formalize this, we assign separate scaling factors to the content and style cross-attention pathways, yielding:

$$Z_{\text{content}} = \mathrm{Attn}(Q, K, V) + \lambda_c \,\mathrm{Attn}(Q, K'_c, V'_c), \tag{5}$$

$$Z_{\text{style}} = \mathrm{Attn}(Q, K, V) + \lambda_s \,\mathrm{Attn}(Q, K'_s, V'_s), \tag{6}$$

where a content scale λ_c is applied to the content blocks and a style scale λ_s to the style block, enabling image features to be incorporated at differentiated intensities across blocks.

E.2. Content-Subtractive Proxy Embedding (CSPE)

Because direct subtraction in the CLIP embedding space can be sensitive to scale discrepancies between inputs, we thus adopt a more stable formulation. Let the original image embedding be $\mathbf{f} = \phi(i)$ and the content-proxy embedding be $\mathbf{u} = \phi(i_c)$. We first apply ℓ_2 normalization to obtain $\tilde{\mathbf{f}} = \mathbf{f}/|\mathbf{f}|$ and $\tilde{\mathbf{u}} = \mathbf{u}/|\mathbf{u}|$. Subtraction is then performed in the normalized space, after which the norm of the original embedding is restored to maintain consistent scaling:

$$\mathbf{e}_{\text{style}} = \|\mathbf{f}\| \, \frac{\tilde{\mathbf{f}} - \tilde{\mathbf{u}}}{\|\tilde{\mathbf{f}} - \tilde{\mathbf{u}}\|}. \tag{7}$$

This procedure mitigates instability arising from scale variation within the CLIP embedding space and yields a more reliable and consistent extraction of stylistic information.

E.3. Orthogonal Fashion Removal (OFR)

We further detail the formulation of OFR. The garment mask M^p is first downsampled to the VAE latent resolution to obtain $\bar{M}^p$. Gaussian blurring and gamma correction are then applied to construct a soft garment mask $\tilde{M}^p \in [0,1]^{C\times H\times W}$. Using this mask, garment-related components in the latent representation z^p are attenuated by removing the projection onto the garment direction only within the masked region:

$$\tilde{z} = z^p \odot (1 - \tilde{M}^p) + z_{\text{proj}} \odot \tilde{M}^p. \tag{8}$$

The final latent representation $\tilde{z}$ is thus updated, while non-garment regions outside the mask retain the original latent values z^p. The resulting $\tilde{z}$ is then used as the initial latent for the subsequent denoising process.

F. Additional Visual Results

In this section, we present results generated under varying random seeds, examples across diverse domains, and visual comparisons under multiple multimodal configurations.

Fig. 4 demonstrates that users can easily explore multiple design candidates under identical conditions, highlighting the practical applicability of our method.

Fig. 5 shows that *FEAT* maintains stable generation quality across markedly different domains, including animations, 2D game characters, 3D game characters, and sculptures. This robustness stems from *FEAT*'s training-free design, which does not rely on domain-specific finetuning and therefore avoids overfitting to any particular data distribution. Notably, the third example (3D game characters) il-

lustrates *FEAT*'s ability to produce precise and natural garment edits even in the presence of complex backgrounds, highlighting its strong and reliable editing capability.

Finally, Figs. 6, 7, and 8 compare our results with baselines under multiple multimodal input conditions, including text, sketches, and images. These results show that *FEAT* not only integrates diverse input modalities in a coherent manner but also naturally resolves residual garment artifacts commonly observed in prior methods. Overall, *FEAT* maintains consistent editing quality in multimodal settings while offering both high controllability and strong visual realism.

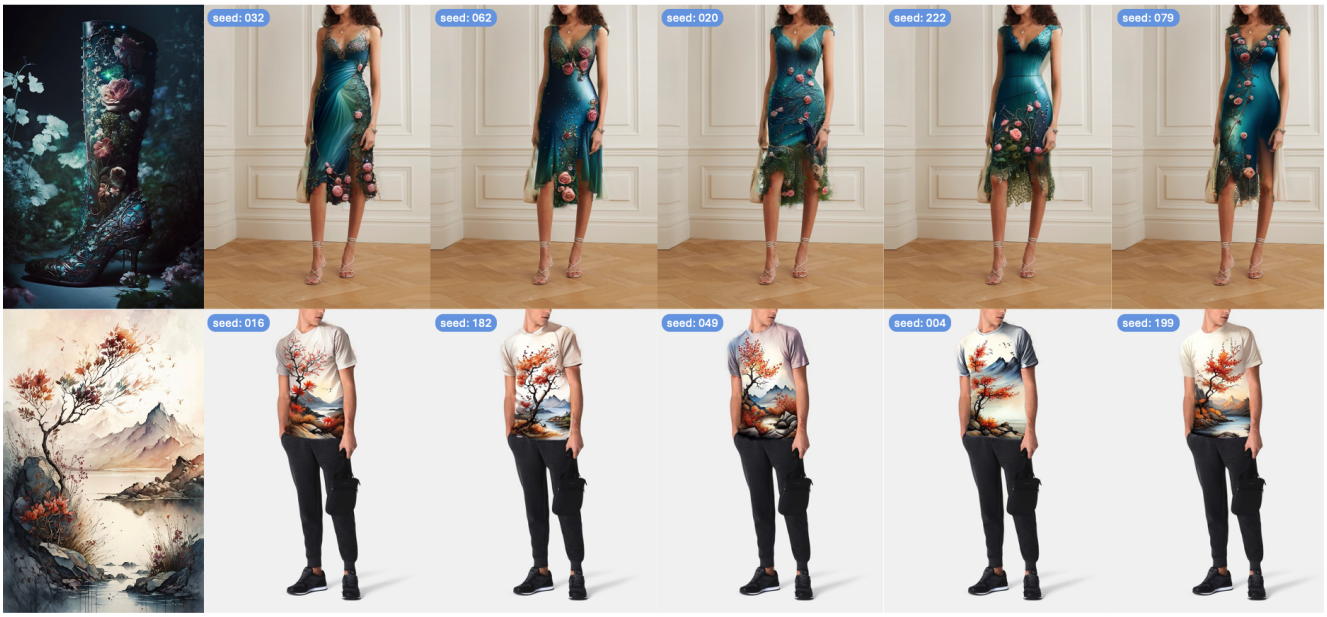

Figure 4. Diverse yet coherent results across random seeds, allowing exploration of multiple design options.

Figure 5. Visual results across diverse domains. From left to right: animation, 2D game characters, 3D game characters, and sculptures.

Figure 6. Visual comparison under the sketch+image setting.

Figure 7. Visual comparison under the sketch+text setting.

Figure 8. Visual comparison under the sketch+image+text setting.

You are an expert evaluator for virtual try-on systems.

You will receive five images:

1. input person image
2. image prompt (e.g., artwork, abstract imagery, or natural imagery)
3. sketch image (structure)
4. text prompt (design detail)
5. result A
6. result B

Evaluate result A versus result B using the three criteria below.
For each criterion pick exactly one number:

- 1 = result A is better
- 2 = result B is better
- 3 = cannot decide / tie

Evaluation criteria

1. Multimodal Prompt Consistency
   - Evaluate how well each virtual try-on result reflects the structure from the sketch, the visual from the image prompt, and the detail described in the text prompt.
   - Copy-pasting non-fashion elements from the image prompt as a failure.

2. Identity Preservation
   - Evaluate how well the non-fashion parts in each output image match the input person image.

3. Virtual Try-On Realism
   - Overall photographic realism: lighting consistency, texture sharpness, absence of artifacts.

Output format (IMPORTANT)

Return a single line:

Final answer: <criterion 1> <criterion 2> <criterion 3>

Example (do not explain):
Final answer: 1 2 3

Figure 9. An example of a prompt provided to the GPT-4V evaluator.

You are a helpful assistant that generates fashion-design text prompts for virtual try-on.

Please create "${nums}" text prompts for the fashion item "${category}" for virtual try-on applications.

Guidelines

1. Item Type
   - Every prompt must mention "${category}".

2. Design Details (mandatory)
   - Focus only on decorative or embellishment elements (not structural features such as shape or silhouette).
   - You may invent or vary decorative elements beyond the following examples:
     - lace, ribbon, embroidery, beads, crystal embellishments, buttons, character prints, logos, metal clasps, frills, pleats, sequins, tassels, charms, chains, cut-outs, appliqué, studs, zippers, layered trims, buckles, etc.

3. Descriptive Attributes
   - Include an emotional mood adjective (e.g., elegant, bold, romantic, vintage, sleek, sophisticated, charming, ethereal) in exactly 20 % of prompts only.

4. Complexity
   - Mix simple items (minimal details) with more intricate ones.
   - Even detailed prompts stay within the 1–2 embellishments limit.

5. Creativity
   - Balance familiar everyday designs with imaginative, unique creations.

6. Prompt Format
   - Write clear, concise sentences describing only the design details and overall impression of the item itself.
   - Do not mention shape, silhouette, colour, or texture.
   - Avoid styling verbs or suggestions such as "Channel", "Opt for", or "Elevate your look" that imply how to wear or style the item.

Number the prompts 1 to "${nums}".

Figure 10. An example prompt used to generate text prompts.

References

[1] Midjourney, 2025. Generative image platform. 1

[2] Namhyuk Ahn, Junsoo Lee, Chunggi Lee, Kunhee Kim, Daesik Kim, Seung-Hun Nam, and Kibeom Hong. Dreamstyler: Paint by style inversion with text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 674–681, 2024. 3

[3] Alberto Baldrati, Davide Morelli, Giuseppe Cartella, Marcella Cornia, Marco Bertini, and Rita Cucchiara. Multimodal garment designer: Human-centric latent diffusion models for fashion image editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3

[4] Seunghwan Choi, Sunghyun Park, Minsoo Lee, and Jaegul Choo. Viton-hd: High-resolution virtual try-on via misalignment-aware normalization. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2021. 1

[5] Zheng Chong, Xiao Dong, Haoxiang Li, Shiyue Zhang, Wenqing Zhang, Xujie Zhang, Hanqing Zhao, Dongmei Jiang, and Xiaodan Liang. Catvton: Concatenation is all you need for virtual try-on with diffusion models. *arXiv preprint arXiv:2407.15886*, 2024. 1

[6] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 1, 2

[7] Peike Li, Yunqiu Xu, Yunchao Wei, and Yi Yang. Self-correction for human parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3260–3271, 2020. 1, 3

[8] Davide Morelli, Matteo Fincato, Marcella Cornia, Federico Landi, Fabio Cesari, and Rita Cucchiara. Dress code: High-resolution multi-category virtual try-on, 2022. 1, 2

[9] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. 3

[10] Linzi Qu, Jiaxiang Shang, Hui Ye, Xiaoguang Han, and Hongbo Fu. Sketch2human: Deep human generation with disentangled geometry and appearance control. *arXiv preprint arXiv:2404.15889*, 2024. 3

[11] Wei Ren Tan, Chee Seng Chan, Hernan Aguirre, and Kiyoshi Tanaka. Improved artgan for conditional synthesis of natural image and artwork. *IEEE Transactions on Image Processing*, 28(1):394–409, 2019. 1

[12] Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. Gpt-4v (ision) is a human-aligned evaluator for text-to-3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22227–22238, 2024. 1, 2

[13] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. 2023. 3