RAGDiffusion: Faithful Cloth Generation via External Knowledge Assimilation

Supplementary Material

Figure 1. Generalization comparison on crazy difficult garments.

In Sec. 1, we demonstrate two additional benefits brought by the RAG system: enhanced generalizability, and human-interpretable control through landmark manipulation. In Sec. 2, we discuss the limitations of our work and outline directions for future research. Given that standard garment generation is a novel task, we elaborate on its practical significance in Sec. 3. To enhance reproducibility, we provide a detailed description of the network architecture, hyperparameters, and procedural steps in Sec. 4, thoroughly covering the process of RAG system construction, including dataset. In Sec. 5, we discuss preliminary knowledge relevant to this work. Finally, in Sec. 6, we detail the experimental evaluation process and provide additional results from the ablation study, along with extensive cases of RAGDiffusion within STGarment, Viton-HD and DressCode.

1. Additional rationale for introducing RAG

In addition to enhancing structural determinacy and eliminating structural distortion through the assimilation of structural landmarks and external knowledge, we have also discovered two additional benefits brought by the RAG system: enhanced generalizability, and human-interpretable control through landmark manipulation.

1.1. Enhanced generalizability due to RAG

Generally speaking, Retrieval-Augmented Generation (RAG) can **significantly enhance generalization and robustness in new scenarios simply by updating the retrieval database**, without the need for retraining. This provides a cost-effective and convenient maintenance solution for large generative models, avoiding the expensive process of retraining. We have also observed similar phenomena in RAGDiffusion.

RAGDiffusion is trained on upper-body clothing data and has not encountered lower-body garments during training. In this testing phase, we collect embeddings and corresponding landmarks for 856 lower-body/dress items and incorporate them into the external memory database. Subsequently, we gather 50 lower-body/dress in-the-wild clothing samples as a test set, using a ReferenceNet version as a baseline for comparison. The results in Fig. 2 demonstrate that retrieval significantly improves generalization capabilities.

By injecting the embeddings of lower-body garments along with their corresponding contour landmarks as conditional constraints, **RAGDiffusion produces accurate results in the lower-body domain, showcasing its strong out-of-distribution (OOD) compatibility**. In contrast, the ReferenceNet version is noticeably confused by the concept of lower-body garments and fails to yield meaningful garment structures. This is because current generative models necessitate corresponding training data to perform well. Other baselines face similar issues. *ReferenceNet requires retraining* with lower-body try-on training data to work effectively on lower garments. In contrast, *RAGDiffusion does not need retraining* or expensive try-on training data, as it can achieve results with just standard clothing database update, illustrating its greater flexibility. Of course, RAGDiffusion could perform better on lower garments if it underwent training similar to that on upper garments. This boost in generalization increases the operational maturity of RAGDiffusion, enabling it to effectively handle various OOD images submitted by users.

Generalization on Extreme Clothing Types. As RAGDiffusion works with real-world data, we provide more results to showcase how it handles super varied garments—like crazy patterns or funky designs in Fig. 1. I. ***Crazy patterns.*** Due to the pattern-level and detail-level faithful generation pipeline (Section 3), RAGDiffusion delivers accurate texture and logos even if the garment has crazy patterns, as shown in Main. Fig. 5, 6, and Fig. 1. II. ***Funky shapes.*** Highly varied shapes pose a significant challenge for all methods. Existing methods are nearly impossible to succeed in these scenarios. While our model still has a probability of generating distorted shapes, it also produces correct results sometimes, and overall, RAGDiffusion significantly outperforms the baselines.

Figure 2. RAGDiffusion is able to produce accurate results in the lower-body domain, showcasing its strong out-of-distribution compatibility and generalization ability.

Figure 3. Users can modify the final visual presentation by replacing recommended silhouette masks/landmarks with one of K candidates before UNet denoising, highlighting RAGDiffusion's advantages in human-interpretable control and retrieval-based recommendation manipulation.

1.2. Human-interpretable control through landmark manipulation due to RAG

In practice, the retrieval-acquired silhouette mask provides users with a visual shape preview opportunity before UNet denoising. Users can modify the final visual presentation by replacing recommended silhouette masks/landmarks (*i.e.*, selecting from the recommended K nearest neighbor landmark candidates during retrieval) before UNet denoising. For instance, in formal *shirt* scenarios, complete flattening may be required to convey a serious aesthetic, whereas in *hoodie* cases, users might prefer slightly bent sleeves with mild surface wrinkles to achieve a casual and relaxed appearance. As shown in Fig. 3, we demonstrate several style modification cases of flat-lay garments through human-interpretable manipulation. This functionality is absent in end-to-end generative models like ReferenceNet [4], TryOffDiff [10], and TryOffAnyone [11], highlighting

Figure 4. Failure cases about color bias. RAGDiffusion is possible to encounter color bias due to MSE loss constraints and illumination variance. Extended training durations, along with the incorporation of contrast-enhancing data augmentation techniques, can partially alleviate this issue.

RAGDiffusion's advantages in human-interpretable control and retrieval-based recommendation manipulation.

2. Limitations and future work

We present RAGDiffusion, an efficient RAG framework that supports the generation of standardized clothing assets by effectively addressing the prevalent challenges of structural hallucinations and fidelity in generated images. However, there is also limitations to discuss, which will help us improve the proposed framework further.

Actually, we have observed *a possibility of color bias in nearly pure-colored garments*, particularly under very bright or very dark situations, as shown in Fig. 4. This phenomenon is commonly encountered in image-based image editing, as the reconstruction loss constraints of Stable Diffusion *are not particularly sensitive to color discrepancies*. Additionally, *the illumination variance* can further impact the perceived colors of clothing. We note that extended training durations, along with the incorporation of contrast-enhancing data augmentation techniques, can partially alleviate this issue.

In our future work, we aim to enhance RAGDiffusion in both its application depth and breadth. Firstly, by strengthening the injection of color information and incorporating lighting simulation, we hope to address the potential color bias observed in garments. Secondly, we intend to expand RAGDiffusion to encompass additional categories such as bottoms, dresses, and shoes, thereby achieving more comprehensive coverage.

Retrieval Time	Denoising Time	LLM Caption Time
0.3 seconds	8.0 seconds	3.5 seconds

Table 1. Sampling time cost on an RTX 3090 GPU at resolution of 768×768. Retrieval cost is a negligible part of generation.

3. Downstream applications of the standard garments

The standard garment acts as a vital intermediary variable connecting a range of downstream applications, including garment design, product display, and virtual fitting. It actually serves as an essential standard element within e-commerce databases. In this context, we illustrate several examples that showcase the relationships between standard garments and their associated downstream applications in Fig. 5. The image editing results are sourced from SDXL-inpainting [7], while the remaining outcomes are derived from existing toolkits.

4. More implementation details

4.1. Computational costs report

The computational overhead is relatively negligible when facing a larger retrieval memory database. In practical generation, we store feature vectors of a standard garment set locally as the retrieval memory database. During sampling, we retrieve the most similar garment vectors efficiently with FAISS [2] from the memory database in parallel based on the feature vector of the input image. On the other hand, the speed bottleneck of the generation process lies in the iterative denoising process, where retrieval time is a negligible part, as shown in Tab. 1.

4.2. StructureNet

The extraction of pure visual features from in-the-wild clothing images presents challenges due to several local factors, including but not limited to: 1) occlusions or creases resulting from complex poses that obscure clothing content; 2) the shapes of garments being less pronounced when worn on a person, making it difficult to assess fit accuracy; and 3) in-the-wild images often containing foreground obstructions or inner garments, which can cause visual generative models to mistakenly incorporate these external objects into the final generation results. To mitigate the limitations of pure visual feature extraction, we introduce an LLM that leverages extensive pre-training on vast amounts of data to enhance our understanding of clothing.

Discrete attributes prediction by LLM. As shown in Tab. 2, we employ the Qwen2-VL-7B [1] language model to extract a total of 14 clothing attributes, of which 10 con-

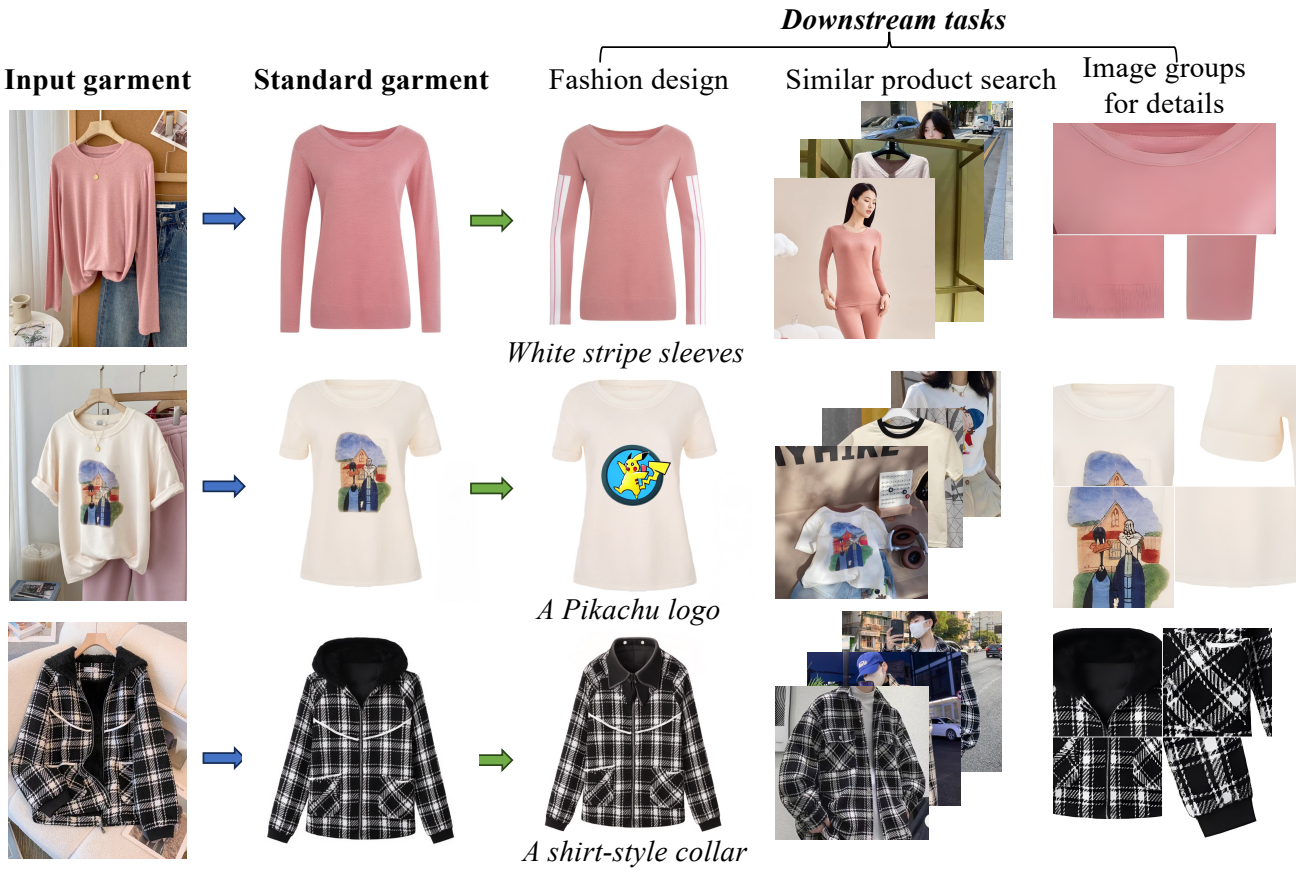

Figure 5. The standard garment serves as an essential standard element within e-commerce databases. In this context, we illustrate several examples that showcase the associated downstream applications.

Figure 6. Visualization of data items in our collected dataset STGarment.

tribute to the construction of the garment structure embeddings, indicated by an asterisk $*$. The language model also provides captions describing the clothing content, thereby enhancing our external knowledge. We utilize 4 NVIDIA H20 GPUs to deploy the Qwen2-VL-7B service, achieving the attribute and caption generation time of under 2 seconds.

Embedding encoding. In the construction of StructureNet, we first extract features f_{img} from in-the-wild clothing images or flat-lay images using the CLIP-ViT-L/14 [8] image encoder backbone (specifically from the *second-to-last* layer), resulting in a feature dimension of 768. Subsequently, we assign a 32-dimensional learnable embedding to each attribute extracted from the language model and concatenate the attribute features f_{attr} with the image features f_{img} to form a vector in $\mathbb{R}^{768+10\times32}$. Through the Resampler module [12], we conduct a non-linear mapping, ultimately forming an embedding e. The Resampler module consists of 4 layers of MLP and 4 layers of attention mechanisms, derived from the IP-Adapter open-source code. For features originating from different image domains (x_{itw}, x_{std}), we designate the extracted embeddings as (e_{itw}, e_{std}). The dual-tower ViT image encoder combined with the Resampler module is referred to

as StructureNet. As described in the main body of paper, StructureNet is trained on STGarment using contrastive learning for 4 days on 4 NVIDIA H20 GPUs with a batch size of 128.

4.3. EP-Adapter

Inspired by IP-Adapter-plus-XL [12], we employed a similar structure for feature information injection. The key element is that the Resampler module consists of 4 layers of MLP and 4 layers of attention mechanisms, serving as an adaptation head to modulate the structure embedding into the text embedding space. Subsequently, after encoding the values and keys, new additional cross-attention layers are integrated as follows:

$$\mathsf{attention}(Q, K_{\text{text}}, V_{\text{text}}) + \mathsf{attention}(Q, K_{\text{image}}, V_{\text{image}}). \tag{1}$$

Landmark fusion. In the Structure LLE algorithm, the final silhouette landmark $\hat{L}_{sil}$ is fused with optimal weights w^*. A naive approach for integrating silhouette landmarks is to directly apply linear interpolation on the binary masks using the optimal weights w^*; however, this often results in blurred and oversized boundaries, which may confuse the generative model. Therefore, we adopt a compromise approach: first, we perform linear interpolation of the landmarks using the optimal weights w^*, and then we calculate the Intersection over Union (IOU) between the fused mask and several original landmark masks. The original mask with the highest IOU serves as the final mask to be used. This operation effectively combines information from multiple masks, yielding a result with maximum consensus while preserving clear and accurate edges that correspond to a real garment template.

4.4. Guideline for collecting retrieval database

To construct a memory database for retrieval, we base our selection on the training set of 65,131 standard flat-lay garments. Initially, we filter approximately 15,000 samples from the original dataset according to the principle of category balance. Subsequently, we encode the samples into embeddings and employ Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [3] to cluster the samples. This step serves to eliminate potential noisy and non-standard data from the original dataset, thereby mitigating the adverse effects that outliers may have on retrieval results. We then perform random downsampling on the densely distributed samples to filter out overly similar instances. Through these operations, we aim to establish a high-quality, broadly representative standard flat-lay (embedding, garment) retrieval database with relatively low redundancy. By adjusting different clustering and downsampling parameters, we can generate retrieval libraries of four different scales: 1,000, 2,000, 4,000, and 8,000 samples, which are utilized in Section 4.2 of the main body of the paper.

4.5. Dataset

We collect a dataset named STGarment, consisting of 65,131 pairs of *in-the-wild upper clothing* and *standard flat lay clothing* with *corresponding attributes* for training, along with 1,969 pairs for testing. The in-the-wild clothing is categorized into three main display types: clothing worn on a person, clothing laid out indoors, and clothing hung on hangers. Fig. 6 illustrates several examples from the dataset. Before inputting the images into the network, all images are cropped or padded to ensure they are square and then resized to (768, 768).

Data augmentation. Following [6], we have implemented data augmentation techniques that could potentially enhance the model's generalization ability as well as its color accuracy performance. Specifically, the data augmentation operations include (a) horizontal flipping of images, (b) resizing standard garments and in-the-wild garments through padding (up to 10% of the image size), (c) randomly adjusting the image's hue within a range of -5 to +5, and (d) randomly adjusting the image's contrast within a specified range (between 0.8 and 1.2 times the original contrast). Each of these operations occurs independently with a 50% probability. Moreover, these operations are simultaneously applied to both the standard garment and in-the-wild images.

5. Preliminary

Stable diffusion. Our RAGDiffusion is an extension of Stable Diffusion [9], which is one of the most commonly used latent diffusion models. Stable Diffusion employs a variational autoencoder [5] (VAE) that consists of an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$ to enable image representations in the latent space. And a UNet ϵ_θ is trained to denoise a Gaussian noise ϵ with a conditioning input encoded by a CLIP text encoder [8] τ_θ. Given an image $\mathbf{x}$ and a text prompt $\mathbf{y}$, the training of the denoising UNet ϵ_θ is performed by minimizing the following loss function:

$$\mathcal{L}_{LDM} = \mathbb{E}_{\mathcal{E}(\mathbf{x}),\mathbf{y},\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t, \tau_\theta(\mathbf{y}))\|_2^2\right], \tag{2}$$

where $t \in \{1, ..., T\}$ denotes the time step of the forward diffusion process, and $\mathbf{z}_t$ is the encoded image $\mathcal{E}(\mathbf{x})$ with the added Gaussian noise $\epsilon \sim \mathcal{N}(0, 1)$ (*i.e.*, the noise latent). Note that the conditioning input $\tau_\theta(\mathbf{y})$ is correlated with the denoising UNet by the cross-attention mechanism.

IP-Adapter. The Image Prompt Adapter (IP-Adapter) is utilized to condition the Text-to-Image (T2I) diffusion

Table 2. All attributes and their values are displayed. The total of 14 clothing attributes is extracted using LLM, of which 10 contribute to the construction of the garment structure embeddings, indicated by $*$.

Attribute	**Choices**
*Category**	T-shirt, Hoodie, Shirt, Polo, Tank, Vest, Swimsuit, Sweater, Innerwear, Windbreaker, Down Jacket, Jacket, Suit, Waistcoat, Shawl, Dress, Skirt, Knitted Coat, Leather Short Coat, Leather Long Coat, Denim Jacket, Robe, Loungewear Top, Loungewear Dress, Sports Jacket, Knitted Cardigan, Leather Jacket
*Fit**	Loose, Regular, Slim
*Collar**	Suit, Shirt, Notched, Rounded, Ruffled, Naval, Hooded, Polo, V-neck, Square, Round, Strapless, One-shoulder, Off-shoulder, Neckline, Stand-up, Baseball
*Sleeve Length**	Sleeveless, Short, Mid, Long, Extra Long
*Fabric**	Gauze, Tweed, Fur, Chiffon, Denim, PVC, Micro-Suede, Fleece, Corduroy, Knit, Lace, Synthetic, Stretch, Linen, Wool, Silk, Knitting, Leather, Velvet, Fur Blend, Coated, Mixed, Special Fabric
Print	Floral, Animal, Skull, Character, Paisley, Baroque, Traditional, Cartoon, Artistic, Tech, Hand-painted, Striped, Plaid, Heart, Polka Dot, Star, Tie-dye, Camouflage, Linear, Text, Logo, Geometric, Color Block, Mixed, 3D Floral, Floral, Solid Color, Nature Scene, Objects
Surface Texture	Layered, Tied, Slit, Cutout, Ruched, Pleated, Spliced, Ruffle, Contrast Stitching, Quilted, Gathered, Applique, Overlay, Hand Decorated, Beaded, Washed, Dyed, Distressed, Frayed, Printed, Splatter, Foil, Rhinestone, Flocked, Embroidered, Edge Decoration, Embossed, Punched, Knit Rib, No Craft
Age	Adult, Child
Gender	Female, Male
*Length**	Extra Short, Short, Medium, Long, Extra Long, Uncertain
*With Inner Wear**	Yes, No
*Sleeves Rolled Up**	Yes, No
*Top Open**	Yes, No
*Top Tuck In**	Yes, No

model with a reference image for style control or content indication. This is typically achieved through global control at a high-level semantic layer. Specifically, it extracts features using an image encoder (*e.g.*, the CLIP [8] image encoder) and incorporates an additional cross-attention layer onto the invariant text conditioning. Here, we denote $Q \in \mathbb{R}^{N\times d}$ as the query matrices extracted from the intermediate features of the UNet, while $K_{\text{text}} \in \mathbb{R}^{N\times d}$ and $V_{\text{text}} \in \mathbb{R}^{N\times d}$ represent the key and value matrices derived from the prompt embeddings, where N signifies the batch size. The cross-attention layer of the text branch is computed as follows:

$$\mathsf{attention}(Q, K_{\text{text}}, V_{\text{text}}) = \mathsf{softmax}\left(\frac{QK_{\text{text}}^{\top}}{\sqrt{d}}\right)\cdot V_{\text{text}}. \quad (3)$$

Subsequently, the IP-Adapter computes the key and value matrices $K_{\text{image}} \in \mathbb{R}^{N\times d}$ and $V_{\text{image}} \in \mathbb{R}^{N\times d}$ from the embedding of the reference image, and integrates the cross-attention layers as follows:

$$\mathsf{attention}(Q, K_{\text{text}}, V_{\text{text}}) + \mathsf{attention}(Q, K_{\text{image}}, V_{\text{image}}). \quad (4)$$

During training, the weights of the original UNet are frozen, and only the projection layers of the key and value matrices in the image encoding branch, as well as the linear projection layer that maps the CLIP image embeddings, can be updated.

K Num.	SSIM ↑	LPIPS ↓	FID ↓	KID ↓
1	0.6872	0.3761	10.27	1.198
2	0.6915	0.3703	10.32	1.190
4	**0.6963**	<u>0.3684</u>	**9.990**	<u>1.092</u>
8	<u>0.6960</u>	**0.3681**	<u>10.01</u>	**1.090**

Table 3. The number K of retrieved nearest neighbors in SLLE impacts generation performance. It's the source data for Fig. 6 in the main body of the paper.

6. Experiment

6.1. Alignment during evaluation protocols

Considering that our task can essentially be viewed as reconstructing a standard flat lay from a conditioning image, we investigate the potential reasons affecting the performance of SSIM and LPIPS. As shown in Fig. 9, we find that (1) the positioning of the generated flat-lay garment may influence the results of SSIM and LPIPS. Due to the lack of explicit positioning, the generated garment may deviate from the ground truth cloth in both horizontal and vertical

Figure 7. More visual results of ablation study.

Figure 8. Retrieved results. We illustrate 4 nearest neighbor results along with their respective landmarks, given in-the-wild input. This visualization aids in intuitively understanding the effectiveness of our retrieval method.

directions, or exhibit minor differences in scale, which impacts SSIM and LPIPS. (2) Even when dealing with a standard flat-lay garment, there may still be some discrepancies in the state of the garment. Factors such as the extent to

Figure 9. We visualize the underlying reasons affecting the performance of SSIM and LPIPS including misalignment and the state of the garment. Factors such as the extent to which the sleeves are spread, the shooting angle, and lighting conditions can significantly affect the measurements of SSIM and LPIPS.

which the sleeves are spread, the shooting angle, and lighting conditions can affect the measurements of SSIM and LPIPS.

To address the measurement errors caused by misalignment (Reason 1), we systematically crop and align the generated images with the ground truth (GT) images according to the bounding boxes, as illustrated in the third column of the Fig. 9. The data presented in the tables of the main body of the paper have all undergone this alignment process.

6.2. About ControlNet baseline

We set the in-the-wild images as the input to the ControlNet conditioning branch, due to the structural aligned mask/canny images are absent without our proposed RAG. Actually, a straightforward ControlNet cannot handle this task well, as the input in-the-wild images do not have structural alignment with the desired output images.

6.3. Retrieval process

We illustrate 4 nearest neighbor results along with their respective landmarks, given in-the-wild input to demonstrate the effectiveness of our retrieval method in Fig. 8. Additionally, we present precise numerical results regarding how the number K of retrieved nearest neighbors impacts generation performance in Tab. 3, which serves as the source data for Fig. 6 in the main body of the paper.

6.4. More visual results

Fig. 7 provides more results about ablation study.
Fig. 10 provides more results on STGarment for inspection to demonstrate that RAGDiffusion synthesizes structurally and detail-faithful clothing assets.
Fig. 11 provides more cross dataset visual results on the unseen dataset Viton-HD, DressCode and the untrained categories lower-body/dresses from RAGDiffusion to validate the enhanced generalizability due to RAG.

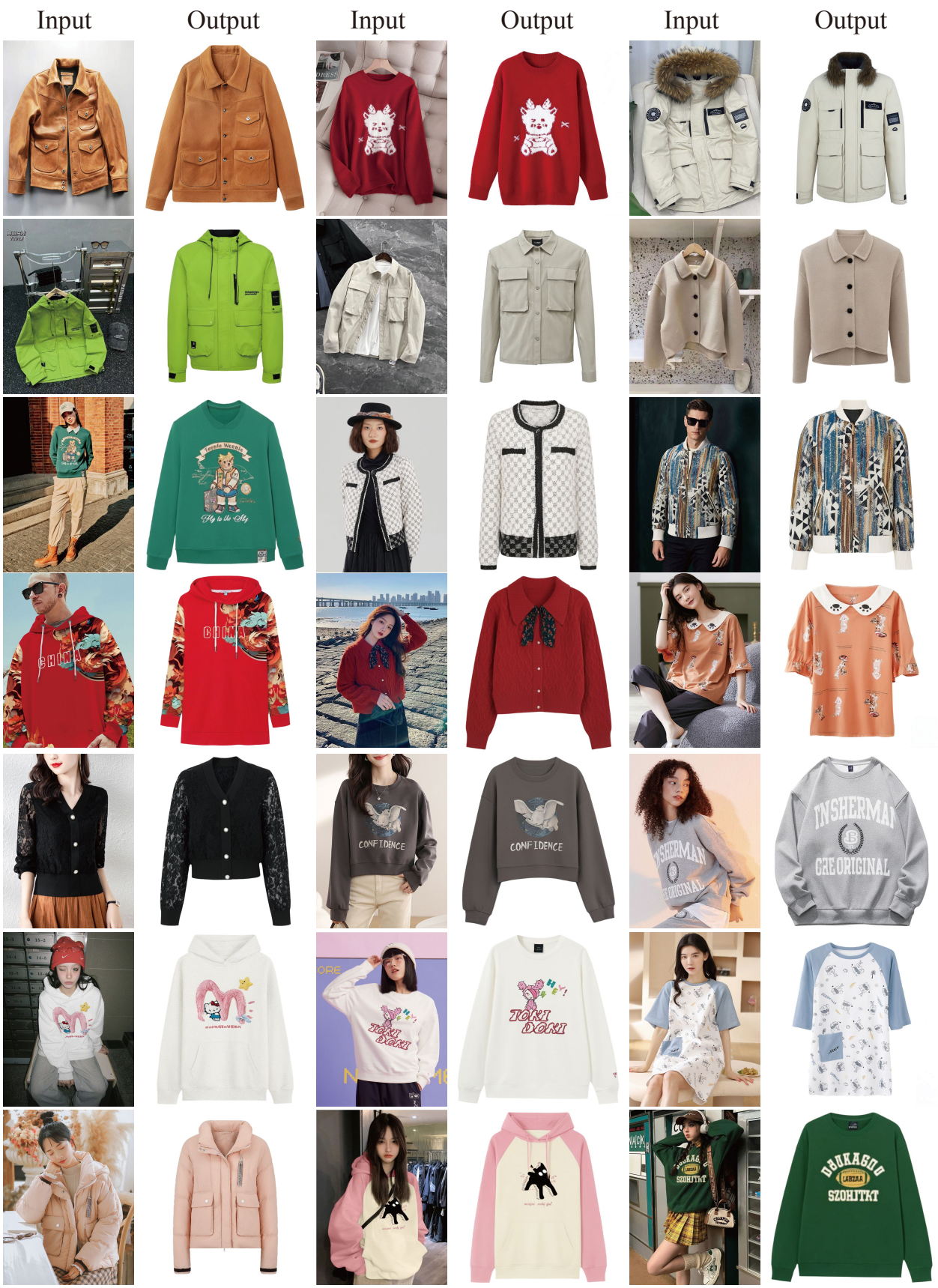

Figure 10. More visual results on the STGarment by RAGDiffusion. Best viewed when zoomed in.

Figure 11. More cross dataset visual results on the unseen dataset Viton-HD, DressCode and the untrained categories lower-body/dresses from RAGDiffusion to validate the enhanced generalizability due to RAG. Best viewed when zoomed in.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 4

[2] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024. 4

[3] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, pages 226–231, 1996. 6

[4] Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. *arXiv preprint arXiv:2311.17117*, 2023. 3

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 6

[6] Yuhan Li, Hao Zhou, Wenxiang Shang, Ran Lin, Xuanhong Chen, and Bingbing Ni. Anyfit: Controllable virtual try-on for any combination of attire across any scenario. *arXiv preprint arXiv:2405.18172*, 2024. 6

[7] 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. *arXiv preprint arXiv:2307.01952*, 2023. 4

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 2021. 5, 6, 7

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022. 6

[10] Riza Velioglu, Petra Bevandic, Robin Chan, and Barbara Hammer. Tryoffdiff: Virtual-try-off via high-fidelity garment reconstruction using diffusion models. *arXiv preprint arXiv:2411.18350*, 2024. 3

[11] Ioannis Xarchakos and Theodoros Koukopoulos. Tryoffanyone: Tiled cloth generation from a dressed person. *arXiv preprint arXiv:2412.08573*, 2024. 3

[12] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 5, 6