CoDE-GAN: Content Decoupled and Enhanced GAN for Sketch-guided Flexible Fashion Editing

ZHENGWENTAI SUN, The Hong Kong Polytechnic University, Hong Kong
YANGHONG ZHOU, The Hong Kong Polytechnic University, Hong Kong and Research Centre of Textiles for Future Fashion, Hong Kong
P. Y. MOK*, The Hong Kong Polytechnic University, Hong Kong and Research Institute of Intelligent Wearable Systems, Hong Kong

Rapid advancements in generative models, including generative adversarial networks (GANs) and diffusion models, have made possible of automated *image editing* through the use of text descriptions, semantic segmentation, and/or reference style images. Nevertheless, in terms of fashion image editing, it often requires more flexible, and typically iterative, modifications to the image content that existing methods struggle to achieve. This paper proposes a new model called Content Decoupled and Enhanced GAN (CoDE-GAN), which is formulated and trained for the task of image editing, drawing on methods from image reconstruction , more specifically, image inpainting with sketch-guidance. Through this proxy task, the trained model can be used for flexible image editing, generating new images with consistent colours and required textures based on sketch inputs. In this new model, a content decoupling block is introduced including specially designed dual encoders, which pre-process inputs and transform into separated structure and texture representations. Moreover, a content enhancing module is designed and applied to the decoder, improving the colour consistency and refining the texture of the generated images. The proposed CoDE-GAN can achieve coarse-to-fine results in one single stage. Extensive experiments on three datasets, covering human, garment-only and scene images, show that CoDE-GAN outperforms other state-of-the-art methods in terms of both generated image quality and editing flexibility. The code will be released once the paper is published.

CCS Concepts: • **Computing methodologies → Computer vision tasks**.

Additional Key Words and Phrases: Fashion image editing, content decoupling, content enhancement, GAN-based.

ACM Reference Format:
Zhengwentai Sun, Yanghong Zhou, and P. Y. Mok. 2018. CoDE-GAN: Content Decoupled and Enhanced GAN for Sketch-guided Flexible Fashion Editing. *J. ACM* 37, 4, Article 111 (August 2018), 23 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Image editing has received considerable attention in recent years and is used in many tasks, such as removal of unwanted objects and adjusting style. Traditionally, image editing is completed by the

*P. Y. Mok is the corresponding author (tracy.mok@polyu.edu.hk).

Authors' Contact Information: Zhengwentai Sun, The Hong Kong Polytechnic University, Hong Kong, Hong Kong, zhengwt.sun@connect.polyu.hk; Yanghong Zhou, The Hong Kong Polytechnic University, Hong Kong, Hong Kong and Research Centre of Textiles for Future Fashion, Hong Kong, Hong Kong, yanghong.zhou@connect.polyu.hk; P. Y. Mok, The Hong Kong Polytechnic University, Hong Kong, Hong Kong and Research Institute of Intelligent Wearable Systems, Hong Kong, Hong Kong, tracy.mok@polyu.edu.hk.

ACM 1557-735X/2018/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

Fig. 1. The user may have various and changeable demands on a fashion garment. For example, they may be satisfied with the skirt color but want to have different styles. Our proposed CoDE-GAN enables the user to draw simplified sketches to flexibly edit the clothing shape with consistent textures.

use of professional software or tools (e.g., Adobe Photoshop, Fotor Photos). The use of such software or tools, however, requires professional knowledge, and the process is also tedious, time-consuming and skill dependent. Benefited from the success in generative models such as generative adversarial networks (GAN) and diffusion models, image editing can now be automated using inputs such as text descriptions [4, 23, 24], semantic segmentation [10, 25], and reference style images [17]. Due to the growing demand for more interactive experiences on fashion online shopping, there has been a large increase in research studies focusing on fashion image editing applications. For example, leveraging GANs or diffusion models, fashion images are edited into different poses by pose-guided image synthesis [3, 30, 52] or different clothing are 'tried-on' on users' bodies on top of their reference photos [32, 49], with astonishing performance in terms of geometric shape or texture transformation.

Nevertheless, most image editing tasks in the fashion domain require more flexible and controllable modifications to the target fashion images. Fig. 1 illustrates a scenario that is common in fashion design or fashion presentation, in which a user wants to make some minor modifications to the partial region of the skirt, like changing its type to be a circle skirt (Fig. 1 (a)) or changing its length to be a tiered skirt (Fig. 1 (b)). Those above-mentioned methods are found not suitable. In order to facilitate users to edit their ideas on the image freely, sketch-guided methods were proposed [1, 18, 27, 43, 46, 48]. Nevertheless, sketch often provides structural information guidance, as it captures only the boundary and lacks detail within the inner region. The existing methods may fail to generate images with consistent structure and textures.

To address this issue, E2I [43] adopts a coarse-to-fine architecture, applying a contextual attention mechanism to improve the synthesized textures in the fine stage. Gated Conv [48] learns a soft mask to weight different regions, improve the model's ability in inferring the missing content by referring to the unmasked region. However, their methods never consider to effectively utilize sketch to guide the synthesis. Instead, they explicitly concatenate the sketch to learn a synthesis. Fig. 2 (a) illustrates this. Instead of simply concatenation, DeFLOCNet [27] argues that the sparse sketch may vanish through the network layers. Therefore, they propose to insert sketch into every skip connections between encoder and decoder. Fig. 2 (b) shows their ideas. Their approach could emphasize the importance of sketch but less effective in inferring sketch-contoured textures. Recently, based on the diffusion model, ControlNet [51] and MGD [3] leverage sketch to guide the image generation. However, they both requires, even heavily relies on, the input of textual descriptions for synthesizing image textures. If absence of a text prompt, ControlNet struggles to produce consistent textures, as illustrated in Fig. 2 (b). To provide more fine-grained textual

descriptions for structure and consistent texture generation, MGD [3] annotates two fashion datasets with textual sentences and sketches for fashion editing.

In this paper, we propose a novel GAN-based model, called **C**ontent **D**ecoupled & **E**nhanced **GAN** (CoDE-GAN), to address these issues. Different from previous GAN-based methods, our proposed model decouples the image content into structure and texture representations through a Content Decoupling Module (CDM). Specifically, a condition decoupling block (CDB) is first used to obtain the structure and texture conditions from the input set x. Then, different encoders are utilized to learn the specific representations for the structure and texture conditions. Fig. 2 (c) shows a simplified structure of our decoupling idea. There are two advantages to doing this: (1) The encoder ϵ_t only handles texture conditions and enables the model to focus more on texture synthesis and learn better texture representations. (2) The structure encoder ϵ_s benefits the latter synthesis process by obtaining structure representation that is distinct from texture representation. These enhanced representations can then help to generate better realistic images having a consistent structure with the input sketch and reasonable texture with the unmasked region. In addition, to further improve the consistency of textures within the content region, we add a Content Enhancement Module (CEM) to the generation decoder. Fig. 3 shows the detailed architecture. The CEM extracts intermediate features from the decoder and transforms them into a single feature map, and then adds a constraint loss to constrain the similarity between the feature map and the grey image. If the synthesized textures are consistent with unedited region, the content response should be similar in feature level.

In conclusion, our contributions are as follows:

(1) We propose a novel network for flexible editing of a fashion image generating content with consistent textures and sketches.
(2) The content decoupling module we designed could faithfully obtain sketch and texture representations which lead to robust performance on flexible editing.
(3) The content enhancement module we designed could improve the consistency of synthesized content.
(4) Extensive experiments are conducted to demonstrate the performance of our proposed CoDE-GAN on four datasets, including fashion human ATR dataset [26], in-shop Garment dataset [5], SG-Fashion Dataset [40] and LSUN outside church dataset [47]. On all these datasets, our method achieves robust and significant performance in FID, SSIM, and PSNR metrics.

2 RELATED WORK

2.1 Fashion Image Editing

Fashion image editing tasks could be classified, according to the target objects being edited, as *human-centric editing* and *garment-centric editing*.

Human-centric Editing. This task primarily addresses two major challenges: virtual try-on and pose transfer. Virtual try-on requires transferring a source cloth to the target human body. While the latter, pose transfer, is about synthesizing consistent human images across a range of poses.

VITON [12] is the first work that considered image-based virtual try-on. They tackled the misalignment issue between source clothing and the target human pose by using TPS [2] to adapt the cloth according to key point correspondences. Moreover, they introduced a refinement network that predicts a composition mask, locating the exact region for try-on. To preserve the characteristics of the source cloth, Wang et al. [41] introduced CP-VTON, which employs a Geometric Matching Module (GMM) to determine the TPS transformation based on human parsing and pose data. LA-VITON [21] leveraged the StyleGAN [20] structure, directing the generator to emphasize local clothing deformation, thereby enhancing the synthesized textures. Despite the significant results

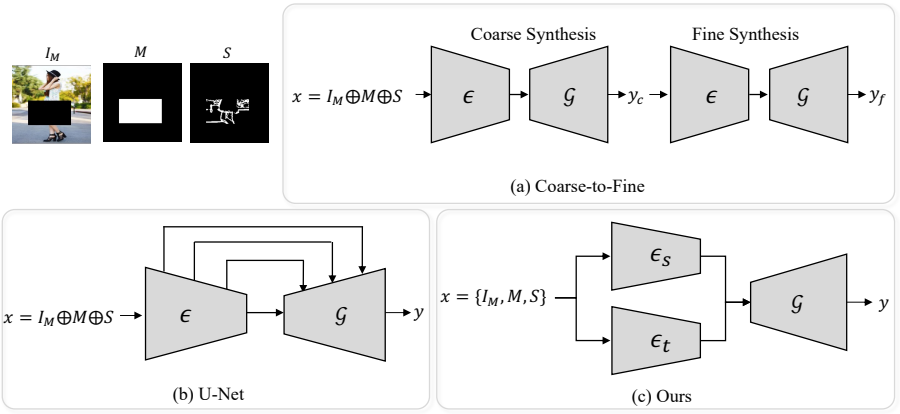

Fig. 2. Simplified network structure of the explicitly concatenation (a), sketch injection (b), and ours decoupling (c).

achieved by the previous work, they are less effective in addressing reasonable try-on results in self-occlusion regions (e.g., crossed arms). To address this issue, Yang et al. addressed this in ACGPN [45] by predicting post-try-on human parsing results, instead of the conventional composition mask. To allow seamless try-on results, they further introduce an inpainting module that fills in the area where the warped cloth will go.

Another approach to address complex pose and self-occlusion is considering pose transfer. Ren et al. [36] introduced a global flow framework that utilizes pose data to predict flow between source and target images. Subsequently, feature patches from the source image can be located in the target image through their proposed attention module. Han et al. [11], on the other hand, predicts this flow based on a human parsing map. Cui et al. [7] combined elements of both virtual try-on and pose transfer, resulting in a comprehensive 'dressing in order' framework. Their approach begins by generating a pose-transferred image and subsequently overlays clothes in a predetermined sequence. This method can simultaneously achieve virtual try-on and pose transfer. To improve the facial fidelity across various poses, SPG-VTON [14] introduces a face identity loss to preserve facial details.

To further enhance the resolution of synthesized images, Choi et al. [6] proposed the VITON-HD dataset and method that could first synthesize high-resolution virtual try-on images. Lee et al. [22] introduced HR-VITON that unifies the previous separated warp module and human parsing map after try-on synthesis into a single condition generator. Their method could effectively consider the relationship between parsing map and warped cloth and therefore preserve the cloth detail. However, both the VITON-HD and HR-VITON primarily focus on upper-body clothing and are unable to handle diverse clothing categories such as bottoms or full-body items like dresses. To overcome this limitation, AVTON [29] introduces the Zalando-Dataset, enabling virtual try-on for various cloth categories. Recently, with the achievement of diffusion models [9, 13, 34, 38, 39], Zhu et al.'s TryOnDiffusion [55] could achieve detail-preserved cloth warping and high-resolution try-on results by their diffusion model. Pan et al. [35] proposed to utilize the diffusion model to improve the warping margin. LaDI-VTON [32] takes the warped cloth as an auxiliary input and inverts the unwarped cloth into semantic space. Therefore, the try-on results could be controlled

not only by the cloth image but also by a text description. Other than clothing try-on, SDGAN [15] incorporates a style disentanglement encoder that supports virtual try-on for glasses.

While the aforementioned approaches have shown impressive outcomes, they come with inherent limitations. They demand multiple inputs like human parsing, pose estimation, text descriptions, etc. Such extensive input requirements not only increase the computational overhead but also limit the flexibility of image editing. For shape editing, like in Fig. 1, users first find a desired in-shop cloth and then utilizes the above-mentioned models to try it on, making the editing process less efficient.

Garment-centric Editing. Dai et al. [8] argued that it is important to edit design drafts. Their fashion editing workflow formulates the fashion editing task as a bidirectional image translation task. By translating an in-shop fashion garment to design draft, it benefits the designer in making modifications. Their pipeline is able to translate the edited drafts back into new in-shop garments. TailorGAN [5] achieves fashion attribute editing by specifying a reference image. To address the lack of paired data between input garments and edited images, TailorGAN proposes a self-supervision training pipeline. By reconstructing a masked attribute region with the guidance of a reference image, TailorGAN could process fashion editing tasks. Nevertheless, this method can only be applied to limited local areas like editing collars and sleeves, which leads to poor generalization to other attributes. Yan et al. [44] proposed CTS-GAN to disentangle fashion image attributes into colors, textures, and shapes as latent representations. Their method can effectively achieve a smooth interpolation from source to target attributes, thus leading to the editing of a fashion image. However, their method fails to edit a specified area but affects the whole image. Apart from editing a fashion item according to user's required attribute, FCBoost-Net [54] models fashion compatibility to edit a fashion item to be compatible with other specified items. Liu et al. [28] proposed MMC-GAN that could generate a compatible cloth according to image and textual conditions. Even though these models are capable of editing fashion garments to some extent, it remains challenging to achieve flexible and precise modifications that target specific garment areas while maintaining overall visual consistency and compatibility with other fashion elements.

2.2 Sketch-Guided Image Inpainting

Image inpainting recovers a masked region with consistent context to a valid region. It assumes that the masked location is given. Due to the loss of information, it is challenging to recover consistent structure and texture. Therefore, Nazeri et al. [33] proposed an edge-connect way for first reconstructing the sketch map in the damaged region. With prior information, the recovered edge map contributed to the completion of the task. Their edge-connecting pipeline enables user-guided editing and could achieve better semantic consistency with the help of inpainted edges. However, their methods explicitly take a masked edge map as input. This is against to real inpainting scenario that the edge map could only been obtained from a damaged image. Directly applying edge detection would turn the masked boundary as edges. The boundary edges would be inconsistent with the clean masked edge map and may lead to the degradation to arbitrary image editing. Therefore, Xu et al. [43] proposed a three-stage network E2I that utilizes sketches to assist in the inpainting process. In the initial stage of the E2I, it inpaints the sketches within the missing areas. These approximated sketches are then fed into the second and third networks, following a coarse-to-fine approach. Unlike Edgeconnect [33], which assumes that the damaged edge has been obtained, E2I directly employs its coarse-to-fine inpainting network with an empty edge as the input. Subsequent to this, edge detection is performed on the resulting images, and the corresponding areas are masked to produce the damaged edge map. Edgeconnect then integrates this damaged edge map back into their entire pipeline again. Although E2I could utilize the edge

map to guide the inpainting process, their pipeline requires a three-stage network and costs much memory and time to inference.

Other edge-guided inpainting work treat the edge as existing additional auxiliary information. In Gated Conv [48], they utilize a coarse-to-fine approach similar to the above-mentioned E2I [43], but with a significant modification: they replace the conventional convolution process with their proposed gated convolution to obtain soft gating weights. This soft gating is realized by introducing an extra convolution process to predict. Their gated convolution is robust in processing free-form masks (random strokes). In addition to the gated convolution, they also integrate an edge map directly into their model so as to guide the inpainting process. Following Gated Conv, Jo [18] introduced SC-FEGAN for addressing face editing tasks. They consider the editing process as image-to-image translation [16], choosing the U-Net structure instead. Their model can modify not just the image shape but also the color by incorporating a user-guided color map as an input. Yang et al. [46] consider the discrepancy between detected edges and human-drawn edges by utilizing an edge refinement network before the edge-guided inpainting network. To avoid the edge information diminishing in the feature space, DeFLOCNet [27] utilizes structure generation blocks to inject the edge into each skip connection in a U-Net. Furthermore, Zeng et al. [50] proposed a sketch-guided inpainting method that does not require a mask input.

Although the above-mentioned approaches could edit an image through learning a sketch-guided inpainting proxy task, they often overlook the gap between editing and inpainting. In editing, user-provided masks roughly determine the editing area. But when editing larger regions, the mask may cover a continuous space, similar to a box-shaped mask. Few work have ever studied the impacts of different masks, e.g. between free-form and box masks. Although Edgeconnect [33] and E2I [43] have considered the effect of mask ratios, they have not explored the influence of different mask shapes. Regarding the input edge, only the work by Yang et al. [46] addressed the shape differences between detected edges and those drawn by humans. However, the effectiveness of binarization remains unexplored in any of these studies.

Benefited from the outstanding capability of Stable Diffusion [37] in synthesizing images from textual descriptions, ControlNet [51] replicates and fine-tunes the encoder from Stable Diffusion to accept sketches as input. Later, Zhao et al. introduced Uni-ControlNet [53], a model capable of handling multiple input conditions simultaneously, such as depth map, segmentation, and others. As a result, both of these ControlNet-based models [51, 53] can generate images that reflect the structure of the input conditions and the textures of the input texts. Nevertheless, it struggles to synthesize textures that are consistent with the unedited regions unless detailed textual descriptions of the textures are provided.

3 METHOD

Our proposed CoDE-GAN includes a **C**ontent **D**ecoupling **M**odule (CDM) and a **C**ontent **E**nhancement **M**odule (CEM). In this section, we will first provide the problem formulation for the sketch-guided image editing. Then we will give the details of specific modules. We will also introduce the optimizing objectives.

3.1 Problem Formulation

The aim of sketch-guided image editing is to synthesize an image with user-intended sketches. Let $I \in \mathbb{R}^{3\times w\times h}$ be the ground truth RGB image where w is the image width and h is the image height, $M \in \mathbb{R}^{1\times w\times h}$ be the binary mask where 1 indicates editing or masked area and 0 indicates the unmasked region, and S be the input sketch, the sketch-guided image editing model will generate a new image which is filled in the consistent texture in the masked region M and has the consistent sketch with S. During the training stage, sketch S is extracted by edge detection network HED [42]

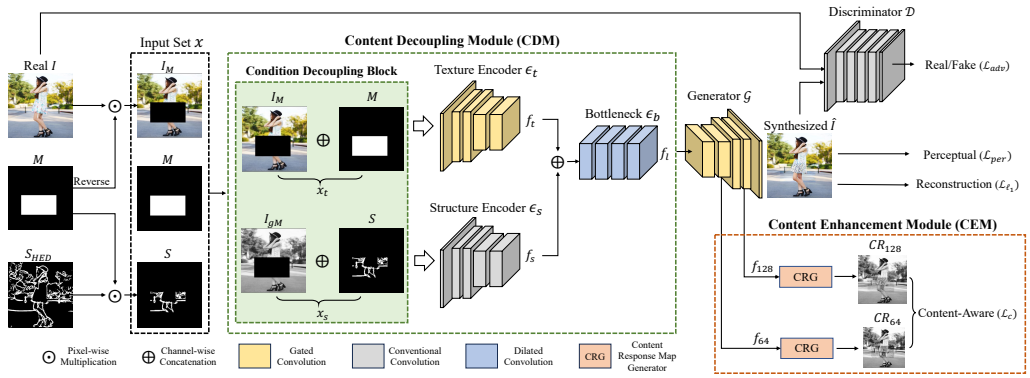

Fig. 3. An overview of our proposed CoDE-GAN. It incorporates Content Decoupling Module to obtain latent representation f_l of the input set x. In the latter generation process, a Content Enhancement Module is applied to further improve the consistency between the synthesized textures and the unedited textures.

$H(\cdot)$ and multiplied with the mask M, which can be defined as:

$$S = H(I) \odot M, \tag{1}$$

where $\odot$ is the element-wise multiplication. Since HED can only output a greyscale sketch map, S is binarized by setting the threshold to 0.6 to simulate users' drawn sketches. During the inference stage, S is drawn by the users in the editing area. In general, the inputs of the sketch-guided image synthesis are the set $x = \{I_M, M, S\}$, where I_M is the masked RGB image obtained by:

$$I_M = I \odot (1 - M). \tag{2}$$

To make the model learn specific texture and structure representation for better image synthesis, the CDM is designed to learn the decoupled texture representation and structure representation and fuse them to obtain better latent representation for image generation. Let the latent representation be f_l, it can be represented by $f_l = \text{CDM}(x)$.

The latent representation is then fed into a generator $\mathcal{G}$ to generate a synthesized image, which is defined by $\hat{I} = \mathcal{G}(f_l)$. Lastly, four loss functions are used to train the network to make the synthesized image $\hat{I}$ similar to the original image I as much as possible. The detail of the loss functions is illustrated in Section 3.5.

3.2 Content Decoupling Module

The content decoupling module consists of a Condition Decoupling Block (CDB), a structure encoder, a texture encoder, and a bottleneck.

Condition Decoupling Block. This block decouples the input x into two types of conditions: the texture condition x_t and the structure condition x_s. Given the image I, the mask M and the sketch S, the x_t and x_s can be computed by:

$$x_t = I_M \oplus M, \quad x_s = I_{gM} \oplus S, \tag{3}$$

where $\oplus$ is channel-wise concatenation, and $I_{gM} \in \mathbb{R}^{1\times w\times h}$ is the grey image of I_M. It can be seen from the formulas that the input $x_t \in \mathbb{R}^{4\times w\times h}$ aligns the setting of image inpainting and the input $x_s \in \mathbb{R}^{2\times w\times h}$ is conditioned to the sketch. Here, x_s incorporates sketch with the grey image instead of RGB image, because grey image is more effective to represent structural information than RGB

image and reduces the representation space from $\mathbb{R}^3$ to $\mathbb{R}^1$. Moreover, traditional image processing algorithms, such as Canny edge detection, typically work with grey images to obtain edge details.

Texture Encoder. The texture encoder ϵ_t feeds in the condition x_t and learn the texture representation by $f_t = \epsilon_t(x_t)$, where ϵ_t is the texture encoder. As the texture encoder mainly aims to reconstruct the texture of the masked region, which is the same as the image inpainting task, we adopt the encoder structure of Gated Conv [48]. Gated Conv designs a gated convolution that adapts a dynamic feature selection mechanism to make the convolution dependent on the soft mask that is automatically learned from data, and improves the texture consistency and inpainting quality of the masked region. Specifically, for the input feature f_{in}, a gated convolution $Conv_g$ applies an additional convolution to obtain a soft weight map and then multiples it with a learned feature of f_{in}. It is formulated as:

$$\text{Conv}_g(f_{\text{in}}) = \text{Conv}(f_{\text{in}}) \odot \sigma(\text{Conv}_d(f_{\text{in}})), \tag{4}$$

where Conv is the conventional convolution, Conv_d is the convolution that outputs single-channel feature map, and σ is the sigmoid function that scales learned gating to range $(0, 1)$.

Sturcture Encoder. The structure encoder ϵ_s takes the input x_s and learns the structure representation f_s by $f_s = \epsilon_s(x_s)$. The structure of ϵ_s is same with ϵ_t, but the gated convolution is replaced with conventional convolution. There are two reasons why we use conventional convolution here: 1) we wish this encoder mainly focused on capturing the basic structure of the whole image, and thus the texture information learning is not that important and will be achieved by the texture encoder. 2) Gated convolution adapts an extra convolution to learn the soft weighting map, leading to an increase in computation cost.

Bottleneck. Lastly, we fuse the texture representation and structure representation by a bottleneck structure to reduce the representation space. The bottleneck structure consists of four dilated gated convolution blocks. Dilated convolution can increase the receptive field without significantly increasing computation. This feature is important at the bottleneck layer, as a larger receptive field enables the network to capture more contextual information. Consequently, the bottleneck can better preserve feature representations, both local and global ones. We firstly concatenate f_t and f_s, and feed it to a bottleneck ϵ_b to object the fused latent representation f_l. It is formulated as:

$$f_l = \epsilon_b(f_t \oplus f_s). \tag{5}$$

3.3 Adversarial Generation

To allow the synthesized results more realistic and reasonable, the adversarial generation process is incorporated.

Generator. Given the fused latent representation f_l, the generator $\mathcal{G}$ could synthesizes a fake image $\hat{I}$:

$$\hat{I} = \mathcal{G}(f_l) \odot M + I_M. \tag{6}$$

The $\mathcal{G}$ consists of five gated convolution blocks with twice upsampling which is symmetric to the structure of encoder ϵ_t.

Discriminator. Following with Pix2Pix [16], we implemented a patch discriminator $\mathcal{D}$ which output real/fake discrimination on image patches instead of the whole image. Its discrimination could focus on local details and enhance the fidelity of the generated image. The structure of $\mathcal{D}$ is similar to an encoder, comprising six convolutional blocks. Besides, to stabilize the adversarial training process, we adopted spectral normalization on the discriminator as well [31].

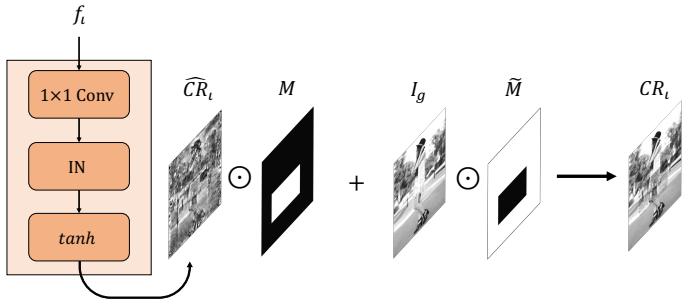

Fig. 4. Content response map generator (CRG) transforms features into content response map. The response map is masked and fused with a grey image.

3.4 Content Enhancement Module

To further improve the consistency of synthesized content, a Content Enhancement Module (CEM) is applied to the generator $\mathcal{G}$. As shown in Fig. 3, CEM extracts the features from the second and fourth blocks. The features have different resolutions and are denoted as f_{64} and f_{128} of which the subscript indicates the resolution of the feature map. It is important to note that interpolation is used to align different resolutions of features blocks, and this may also introduce artifacts and information loss. CEM is therefore designed to reduce such adversarial effects and better preserve the content quality. Since the CEM involves additional computation, we only applied it to the specific blocks, balancing the computation and content quality.

Then, the two features f_{64} and f_{128} are respectively fed into a **C**ontent **R**esponse Map **G**enerator (CRG) to generate the content response maps CR_{64} and CR_{128}. As Fig. 4 illustrates, we obtain the content response map CR_i by:

$$\begin{aligned} CR_i &= CEM(f_i) \\ &= \tanh\left[\mathrm{IN}(\mathrm{Conv_d}(f_i))\right] \odot M + I_g \odot (1 - M), \end{aligned} \tag{7}$$

where $i = 64$ or $i = 128$, $\mathrm{Conv_d}$ reduces the feature dimensionality of f_i to single, IN denotes an instance normalization layer, and tanh is a Tanh activation function. Then, we calculate the cosine similarity between CR_i and the grey image I_g and regard it as an objective function, which is computed by:

$$\mathcal{L}_c = \left(1 - \frac{CR_{64} \cdot I_g}{\|CR_{64}\| \left\|I_g\right\|}\right) + \left(1 - \frac{CR_{128} \cdot I_g}{\|CR_{128}\| \left\|I_g\right\|}\right). \tag{8}$$

We hope that by minimizing the similarity distance between the CR_i and the grey image I_g, the features of generator can also be optimized through gradient backpropagation.

We visualize content response maps at resolutions 64×64 (CR_{64}) and 128×128 (CR_{128}) in Fig. 5. It could be observed that the CEM could learn the structure and texture of the image and the content response map with a higher resolution clearly exhibits more uniform content and sharper boundaries. Since the input sketch is sparse and gradually diminishes in the CNN feature space, it is important to inject the sparse sketch information in the CNN space, especially in the generator. In DeFlocNet [27], the control inputs are injected in all blocks of encoders and generators to preserve the guidance information. However, this method will add additional computation costs and can not provide other content information except the input controls, like the structure and texture information around the sketch. In our case, we optimize the features of the generator to be like the original grey image which has rich structure and texture information, the generator will

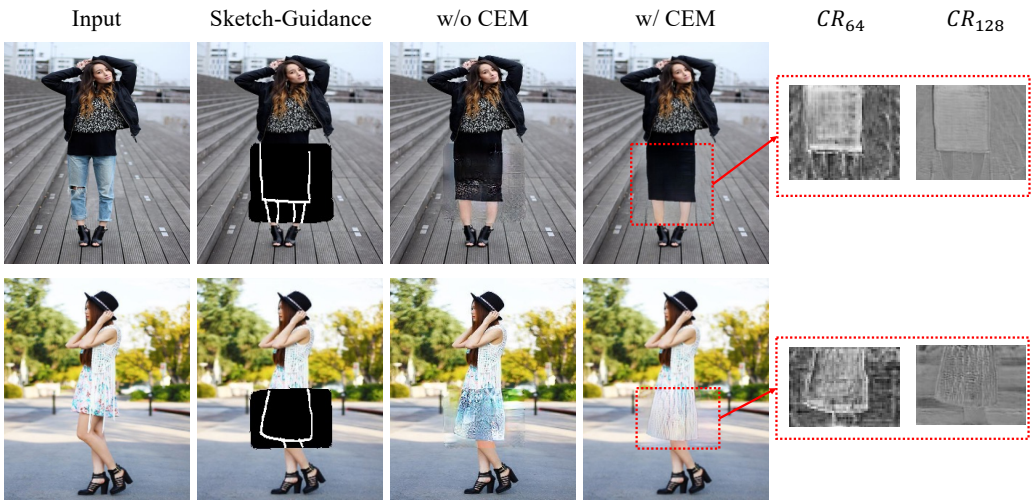

Fig. 5. Visualization of the synthesized content response map CR at resolution of 64×64 and 128×128.

learn to recover the structure and texture of the masked region as shown in Fig. 5. Therefore, our proposed CEM is able to enhance and refine the content information, leading to more detailed and high-quality generation results.

3.5 Optimization Objectives

For training the CoDE-GAN, except for the above-mentioned content-aware loss, we use reconstruction loss, perceptual loss, and generative adversarial loss. In the following, we will introduce these loss functions.

Reconstruction Loss. To ensure the generated image $\hat{I}$ is close to the RGB image I within the unmasked region, L1 loss is used between them on the unmasked region. It is defined by:

$$L_{\ell 1} = |I - \hat{I}|_1 \odot M. \tag{9}$$

Perceptual Loss. Following style transfer, we introduce perceptual loss [19] to keep the perceptual information as well. It is obtained by:

$$\mathcal{L}_{per} = \sum_i w_i \cdot L1\left(F_i(I) - F_i(\hat{I})\right), \tag{10}$$

where F_i stands for ith activation layer of VGG-19 network, and w_i is the corresponding weight. Specifically, the selected layers are *relu1_1, relu2_1, relu3_1, relu4_1* and *relu5_1*. In our experiments, we set the all corresponding weight w_i as 1.0.

Generative Adversarial Loss. The synthesis process is conditioned to inputs $x = \{I_M, M, S\}$. To allow the discriminator $\mathcal{D}$ to consider the conditions, despite the real/fake image I and $\hat{I}$, $\mathcal{D}$ will take x as well. We adopted hinge loss for optimizing spectral normalized discriminator $\mathcal{D}$:

$$\begin{aligned} \mathcal{L}_{adv}^{D} =& \mathbb{E}_{I,x}[\min(0, -1 + \mathcal{D}(I, x))] + \\ & \mathbb{E}_{\hat{I}_x,x}[\min(0, -1 - \mathcal{D}(\hat{I}, x))]. \end{aligned} \tag{11}$$

And the adversarial loss for the total network CoDE-GAN:

$$\mathcal{L}_{adv}^{G} = -\mathbb{E}_{\hat{I},x}[\mathcal{D}(\hat{I}, x)]. \tag{12}$$

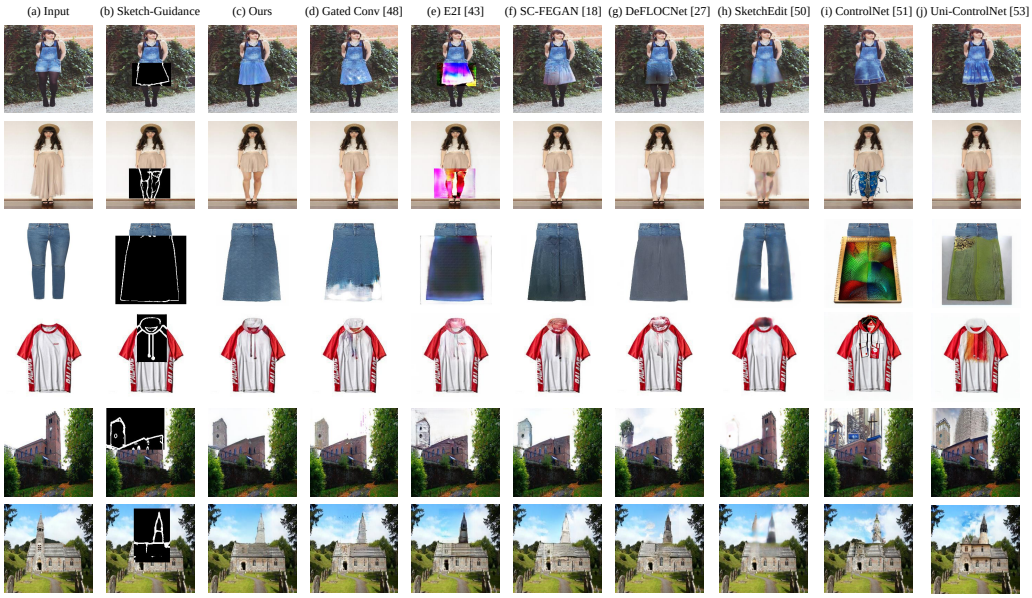

Fig. 6. Comparisons with state-of-the-art methods: Gated Conv [48], E2I [43], SC-FEGAN [18], DeFLOCNet [27], SketchEdit [50], ControlNet [51], and Uni-ControlNet [53]. The first two rows show the qualitative results in ATR Dataset. The third and fourth rows plot the results of SG-Fashion Dataset and Garment Dataset. The last two rows present object shifting of the tower part in LSUN outdoor church dataset.

The overall objectives are:

$$\mathcal{L} = \lambda_{\text{per}} \mathcal{L}_{\text{per}} + \lambda_{\ell 1} L_{\ell 1} + \lambda_c \mathcal{L}_c + \mathcal{L}_{adv}^G , \tag{13}$$

where $\lambda_{per}, \lambda_{\ell 1}, \lambda_c$, and λ_{adv} denote the coefficients for perceptual loss, reconstruction, content-aware loss and adversarial loss, respectively.

4 EXPERIMENTS

We conducted extensive quantitative and qualitative experiments on several datasets, such as fashion human, garment, and outdoor church dataset, to demonstrate the effectiveness of our proposed CoDE-GAN. In this section, we first introduce the dataset and experiments settings. Then, we compared with the state-of-the-art methods and conducted ablation studies of the proposed CDM and CEM modules. Lastly, we further discuss the influence of edge pre-processing and mask types. It shows that our methods are robust.

4.1 Dataset and Experiments Settings

Datasets. We investigate the effectiveness of our model on two garment datasets (Garment dataset [5] and SG-Fashion dataset [40]), a fashion human dataset (ATR dataset [26]), and an outdoor building dataset (LSUN outdoor church [47]): (1) Garment dataset consists of 9.6k images about upper clothing. (2) SG-Fashion collected 17k images with 72 clothing categories across upper, bottom, and full body clothing. (3) ATR dataset comprises of 17.7k humane images with various poses and complex background. (4) LSUN outdoor church is a subset of LSUN dataset which consists of 126k images. During our experiments, these above-mentioned datasets are divided into train

Fig. 7. Flexible editing results on challenging image content and areas. The first two rows show the editing of pose and clothing styles on fashion humans (from ATR dataset). The last two rows are samples from the Garment dataset that edit both sleeves simultaneously.

and test set with the ratio of 9:1. Except the LSUN outdoor church dataset that we adopt its official validation set for testing which contains 300 images.

Evaluation Metrics. Measuring the quality of edited image content in a quantitative way can be challenging. Collecting pairs of edited images for comparison can be time-consuming and expensive. Therefore, to evaluate the editing results in a more practical and cost-effective way, we evaluate sketch-guided image inpainting results on test dataset. This involves inputting a masked image along with the sketch of the masked region from the original image and outputting a reconstruction of the original image. In this way, we can utilize the original image as ground truth to calculate pair-wise metrics.

We employ the Fréchet Inception Distance (FID), Structural Similarity Index (SSIM), and Peak Signal-to-Noise Ratio (PSNR) as our quantitative metrics. FID evaluates the distance between the distribution of the generated images and the ground truth images. By calculating the Fréchet distance in Inception Net's feature space, FID effectively captures the perceptual similarity between the two distributions, with a lower FID indicating higher realism and perceptual similarity. PSNR employs pixel-wise differences to compute a ratio between the maximum possible power of a signal and the power of corrupting noise, serving as a measure of the visibility of errors. SSIM measures the structural similarity between the generated image and the ground truth image. SSIM is similar to PSNR but is believed better aligned with human perception. Higher values of SSIM and PSNR indicate better image quality. Given our primary aim of editing images rather than reconstructing them, we have selected FID as our principal metric.

Implementation Details. We utilize HED [42] edge detector to obtain sketches. For simulating manually drawn sketches, we binarize the detected edge maps with a threshold of 0.6. In terms of mask generation, we randomly generate a single rectangular box mask with a ratio of 30% in all experiments. For loss weights, we set $\lambda_{\ell 1} = 100, \lambda_{\text{per}} = 20, \lambda_c = 5, \lambda_{adv} = 1$ in all experiments. We run all experiments with batch size 12 in a single RTX 3090 GPU. We train 500k iterations in each dataset. We adopt the Adam optimizer with a learning rate of 1e-4 for the whole synthesis model and 4e-4 for the discriminator.

4.2 Comparison with State-of-the-Art Methods

We compare our CoDE-GAN with two approaches that utilize coarse-to-fine structures (E2I, and Gated Conv), two pixel translation structures (DeFLOCNet, SC-FEGAN) and mask-free pipeline (SketchEdit), as well as a diffusion-based model (ControlNet, Uni-ControlNet).

- Gated Conv [48]: For the implementation of gated conv, we adopted the implementation of PyTorch version and keep the hyper-parameters consistent with the original implementation.
- E2I [43]: It is a three-stage method which consists of an edge inpainting network and a coarse-to-fine network. In the sketch-guided inpainting, with sketch as a given input, we removed the edge inpainting network and trained the rest with consistent hyper-parameters and losses.
- DeFLOCNet [27]: DeFLOCNet utilized an encoder-decoder structure but chose to inject sketch into each skip connection. We trained it with its original hyper-parameters.
- SC-FEGAN [18]: The original SC-FEGAN utilizes color sketches to edit images. For a fair comparison, we only used the edge sketches as input.
- SketchEdit [50]: It is a mask-free pipeline that estimates a mask from an input sketch to identify affected regions. Without an explicit mask to exclude existing content, it relies solely on the input sketch, often resulting in minimal changes to the original image. Therefore, we only discussed and compared to this work with qualitative results, while a quantitative comparison cannot be given because no mask is used.
- ControlNet [51]: ControlNet leverages the generative capability of diffusion models to provide spatial controls in the text-to-image generation process. However, it requires inputting a textual description to control the synthesized textures. For a fair comparison, we examined its performance using Stable Diffusion V1.5's inpainting model [37] and input generic text descriptions such as *'super detail, high resolution, HD, 4k, best quality'*.
- Uni-ControlNet [53]: Uni-ControlNet enhances the controllability of ControlNet by allowing multiple input types in a single model. The supported inputs include sketches, segmentation maps, depth maps, and others. For a fair comparison with our model, we used Uni-ControlNet in its inpainting mode with sketch maps to generate images, keeping its text input the same as ControlNet.

4.2.1 Qualitative Comparison. Fig. 6 illustrates the qualitative comparison results with the above methods on four datasets. As shown in Fig. 6 (e), we can see that E2I has a relatively poor ability to generate reasonable image on the garment datasets and totally collapses on the ATR dataset which consists of the person images. This is because that E2I directly employs traditional convolution operation to the encoder-decoder architecture. On the one hand, the encoder-decoder network processes features through every layer, lacking direct connections between different layer. This lack of inter-layer connectivity restricts its ability to leverage low-level information effectively, limiting its capacity to generate detailed local textures. On the other hand, the traditional convolution operations cannot distinguish between valid and invalid input pixels, resulting in an undesirable blending of conditional information and generating synthesized results with blurred boundaries. Especially for the clothing images which have various styles, material textures and colors, it is much more difficult to infer much from the valid area to fill in the accurate content for masked region. Therefore, E2I performs relatively poor on the image generation on the Garment datasets. Different from E2I, DeFLOCNet adopts the U-Net architecture and add skip connections between the mirrored layers in the encoder and decoder stacks, allowing low-level information to flow pass across the network and can generate more realistic images than the simple encoder-decoder network. So DeFLOCNet generates better results than E2I. However, DeFLOCNet still uses the traditional convolution operations in the network and fails to generate realistic images. SC-FEGAN

and Gated Conv both use gated convolution operations in the network. They can generate realistic textures for the things without much variations well, like the skin textures (see 2nd row of Fig. 6) and building textures (see 5th and 6th row of Fig. 6). However, the generation results for the garments still have flaws. For example, SC-FEGAN cannot generate consistent textures with the current clothing (see Fig. 6 (f)) while Gated Conv fails to synthesize accurate textures at some pixels (see Fig. 6 (d)). SketchEdit is limited to editing small regions and often replicates the input, as seen in the forth row of Fig. 6. Its mask-free character restricts its ability to remove objects; for instance, the original tower remains in the last two rows of Fig. 6. In columns (i) and (j), both ControlNet and Uni-ControlNet perform well when editing relatively small and simple shapes. Nevertheless, they struggle to edit large regions with consistent textures; for instance, the third row shows strange textures compared to the original cloth. In contrast, our model employs a Content Decoupling Module to learn better texture and content representations and uses a Content Enhancement Module to ensure the consistency of synthesized textures. As a result, our proposed CoDE-GAN generates realistic images with consistent structures and fine-grained texture details on the garment datasets and building datasets.

Fig. 7 exhibits the flexibility of fashion image editing, where all methods are trained using randomly generated single box masks that cover 30% of the image area. By flexibility, we refer to the capability to arbitrarily edit any region regardless of its shape, are, or continuity. More than one mask is used in Fig. 7's samples, demonstrating our method's capacity to handle multiple edits simultaneously. The first two rows display a challenging scenario in the ATR dataset involving irregular masks. In the first row, the left hand's pose is altered and the jeans are lengthened. The second row modifies the short pants into a short skirt while removing the watermark. In these complex editing tasks, our CoDE-GAN succeeds in generating the most plausible textures. On the other hand, E2I, SC-FEGAN, DeFLOCNet, SketchEdit, ControNet and Uni-ControlNet exhibit artifacts on the masked background. Although Gated Conv can accurately represent the edited content, its synthesized clothing textures are less uniform than ours, especially when editing the geometric pattern on the floor in the second row. The third and fourth rows of Fig. 7 highlight CoDE-GAN's ability to handle simultaneous edits on two distinct areas (the sleeves) with discontinuous box masks, which lie outside the training distribution. Despite these challenging conditions, CoDE-GAN consistently synthesizes visually appealing edited images. On the other hand, E2I and DeFLOCNet tend to produce artifacts around the mask boundary. Both Gated Conv and SC-FEGAN struggle with the task, notably failing to correctly synthesize the cuff region in the third row and generating artifacts in the background. SketchEdit and ControlNet show that they are unable to edit the sleeves, whereas Uni-ControlNet can generate the shape of the sleeves but fills them with unrealistic textures.

Table 1. Quantitative comparisons in fashion human ATR dataset and LSUN outdoor church dataset.

Metrics	ATR Dataset								LSUN Outdoor Church Dataset							
	DeFLOC-Net [27]	SC-FE GAN [18]	Gated Conv [48]	E2I [43]	Control Net [51]	Uni-Control Net[53]	Ours	Ours (refine)	DeFLOC-Net [27]	SC-FE GAN [18]	Gated Conv [48]	E2I [43]	Control Net [51]	Uni-Control Net[53]	Ours	Ours (refine)
FID ↓	75.72	69.7	79.02	77.44	30.12	**21.63**	54.47	43.65	34.61	39.34	40.15	39.85	37.29	29.65	30.70	**23.86**
SSIM ↑	0.79	0.80	0.82	0.80	0.66	0.68	**0.83**	**0.83**	0.79	0.79	0.80	0.78	0.66	0.67	**0.81**	**0.81**
PSNR ↑	19.34	20.55	21.98	19.84	16.54	17.53	22.84	**22.95**	21.22	18.93	21.26	18.21	17.93	18.63	19.99	**22.04**

4.2.2 Quantitative Comparison. Tabs. 1 and 2 list the quantitative comparison with the state-of-the-art methods on four datasets. Our CoDE-GAN achieves competitive performance on all the datasets. In terms of FID, CoDE-GAN significantly outperforms other methods, except the ATR dataset where ControlNet and Uni-ControlNet achieve better scores. A lower FID indicates that the generated images are generally closer to the ground truth. However, in pair-wise metrics

Table 2. Quantitative comparisons in Garment dataset and SG-Fashion dataset.

Metrics	Garment Dataset								SG-Fashion Dataset							
	DeFLOC-Net [27]	SC-FE GAN [18]	Gated Conv [48]	E2I [43]	Control Net [51]	Uni-Control Net[53]	Ours	Ours (refine)	DeFLOC-Net [27]	SC-FE GAN [18]	Gated Conv [48]	E2I [43]	Control Net [51]	Uni-Control Net[53]	Ours	Ours (refine)
FID ↓	21.58	21.46	21.35	23.57	24.04	14.55	20.66	**6.21**	22.19	23.07	22.43	26.72	38.59	19.98	13.67	**6.73**
SSIM ↑	0.85	0.85	0.76	**0.87**	0.85	0.80	0.85	**0.87**	0.89	0.90	0.91	0.90	0.76	0.84	**0.92**	**0.92**
PSNR ↑	23.49	23.14	24.91	22.40	15.35	18.58	23.30	**25.15**	26.35	27.91	28.25	27.39	13.52	18.30	**30.53**	29.95

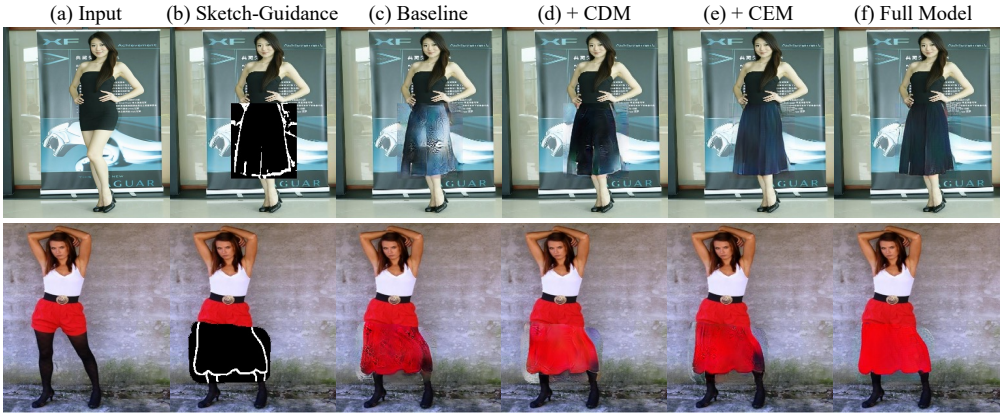

Fig. 8. Qualitative results for ablation studies.

such as SSIM and PSNR, both ControlNet and Uni-ControlNet perform poorly. These quantitative results align with the visual results in Figs. 6 and 7, where the generated textures from ControlNet and Uni-ControlNet differ from the input image. This highlights that our CoDE-GAN excels in maintaining structural and perceptual quality, as indicated by strong performance in the SSIM and PSNR metrics.

Moreover, it should be noted that our proposed CoDE-GAN does not include the refined stage. For fair comparison with Gated Conv and E2I, we followed Gated Conv and used our proposed CoDE-GAN to train a refined model, which further improves the performance by a large margin. This demonstrates that our proposed CoDE-GAN can be easily combined with other methods to improve the generation performance.

4.3 Ablation Studies

Table 3. Ablation results on the designed CDM and CEM modules.

CDM	CEM	FID ↓	SSIM ↑	PSNR ↑
-	-	69.63	0.8070	21.38
w/o Gated Conv	-	56.87	0.8291	22.55
-	✓	55.54	0.8276	22.68
w/ Gated Conv	✓	55.49	0.8212	22.64
w/o Gated Conv	✓	**54.47**	**0.8331**	**22.84**

In this section, we perform ablation studies to analyze the effectiveness of each module of our proposed CoDE-GAN. To this end, we train a series of variant models on the ATR dataset: i) The

Baseline model is the Gated Conv [48] without refine stage. ii) The +CDM is the Baseline model with encoder replaced by our proposed CDM. iii) The +CEM is the Baseline model which adds our proposed CEM. Tab. 3 and Fig. 8 respectively show the quantitative and qualitative results of the variant models and our full model. We can see from the results that our full model is superior to all the variant models. As Fig. 8 (c) shown, the generated image by Baseline reveals flaws when editing the dress length, producing incorrect textures. The CDM and CDM both perform better than Baseline in all metrics, but there are artifacts in the synthesized textures could be further improved. Particularly, compared with Baseline, Fig. 8 (d) shows that +CDM could help to improve the consistency between synthesized textures and unedited image content. On the other hand, +CEM could help to reduce less artifacts and allow the synthesized textures to be uniform. For example, the background artifacts in the second row of Fig. 8 (e) is reduced.in As we can see in Fig. 8 (f), our full model could generate reasonable edited textures.

Tab. 3 shows the quantitative results of ablation experiments carried out on the ATR dataset. The first row represents Gated Conv, our baseline model, on which our CoDE-GAN is built by applying CDM and CEM to its coarse network. The second and third rows show that our proposed CDM and CEM effectively reduce the FID score by about 25% compared to the baseline. Notably, CDM primarily enhances SSIM, while CEM improves FID and PSNR. Combining these two modules together further enhances all of the metrics.

In Tab. 3, w/ and w/o Gated Conv entries under the CDM column indicate whether or not to use gated convolution mechanism within our structure encoder ϵ_s. As shown in the fourth row, the gated convolution does not significantly improve the performance. While gated convolution is effective for emphasizing masked/unmasked regions, our ϵ_s focuses on capturing a fused representation of content and structure. The weighting effect introduced by gated convolution may lead to an imbalance representation.

4.4 Discussions

In this section, we will discuss the impact on more possible settings in our proposed content-aware loss (Eq. 8), different preprocessing on the edge, and training masks.

Table 4. Quantitative results on various combinations of loss function and ground truth types.

Loss Function	Ground Truth	FID ↓	SSIM ↑	PSNR ↑
L1	Grey	17.88	0.87	25.24
Cos	Grey	**5.02**	**0.89**	**26.17**
L1	Segmentation	6.82	0.87	25.08
Cos	Segmentation	6.98	0.86	24.91
L1	Colour	17.57	0.87	25.13
Cos	Colour	15.77	0.88	25.29

Analysis on Content-Aware Loss. Tab. 4 provides the comparison study of content-aware loss, evaluating various combinations of loss (Eq. 8) functions and ground truth types in the Garment dataset with 30% free-form masks. The optimal combination found involves using cosine similarity as a loss function to supervise the content response map with a grey image. This setup outperforms the rest by achieving the lowest FID score of 5.02 and the highest SSIM and PSNR scores of 0.89 and 26.17, respectively. When the L1 loss function is used with a grey image, we observe a decrease in performance. This is likely due to the lack of color information in the grey image, leading to variations in intensity across different images. On the other hand, cosine similarity helps normalize

these intensity ranges, making it a more effective choice for content constraint. We also examined the effect of using foreground segmentation. Although this led to improvements in the metrics, it was not as generalizable as using a grey image, especially with complex datasets like the ATR dataset or LSUN outdoor church dataset. The binary segmentation is insufficient to represent different content regions in these cases, whereas the grey image can distinguish different contents through intensity variations. Finally, we considered the use of a color image for supervision but found the model challenging to optimize. The content response map, synthesized by the CEM from feature maps, is expected to resemble the final RGB output. This places a heavy burden on the CEM, potentially requiring a larger model with more parameters. This goes against our design motivation of maintaining a lightweight, efficient module to apply content constrain. Therefore, this combination did not yield optimal results.

Table 5. Quantitative results on edge preprocessing methods.

Metrics	Binarized Edge					Greyscale Edge				
	DeFLOC-Net [27]	SC-FE GAN [18]	Gated Conv [48]	E2I [43]	Ours	DeFLOC-Net [27]	SC-FE GAN [18]	Gated Conv [48]	E2I [43]	Ours
FID ↓	21.58	21.46	21.35	23.57	**20.66**	9.03	7.77	6.89	17.07	**5.02**
SSIM ↑	0.85	0.85	**0.87**	0.85	0.85	0.86	0.86	0.87	0.87	**0.89**
PSNR ↑	23.49	23.14	**24.91**	22.40	23.30	24.32	24.11	24.69	23.97	**26.17**

Analysis on Edge Binarization. The sketch-controlled editing tasks require the user to input modified sketches. However, current sketch-controlled literatures typically utilize edges extracted via the Holistically-Nested Edge Detection (HED) technique as sketches, which often diverge from actual user inputs. To determine the influence of preprocessing methods on HED-extracted edges, we conducted a comparison study to assess the potential benefits of binarizing these edges to better mimic user-drawn sketches. The results of this investigation, presented in Tab. 5, are based on the Garment Dataset. Each result was evaluated on the corresponding edge preprocessing. By binarizing the extracted grayscale edges with a threshold of 0.6, we observe a substantial deterioration in the quantitative metrics. In particular, the FID metric demonstrates a more than twofold increase for most methods, except for E2I. This suggests that E2I is robust to edge form variations, likely due to its two-stage network implementation, which incorporates edges at each stage. This structure emphasizes the importance of edges, aiding in the reconstruction of spatial structure. Our method, however, still achieves the best FID score of 5.02 on grayscale edges and 20.66 on binarized edges. The decrease in performance is attributed to information loss during edge binarization. The HED-extracted edges represent the likelihood that a given pixel could be an edge. Consequently, aside from actual edge pixels, there is a higher probability assigned to pixels near the edge, providing crucial prior information about an image's spatial structure. Upon binarization, the data becomes too sparse to effectively guide the model in reconstructing spatial structure. Although grayscale edges allow models to perform well quantitatively, their characteristics do not resemble the naturalistic qualities of human-drawn sketches. Therefore, we adopt binarized edge training on all of the qualitative results.

Analysis on Training Mask Settings. The flexible editing may involve various mask types. To evaluate the impact on the robustness of different masks when trained on a specific mask type, we conduct comparison experiments on Garment Dataset. Tab. 6 and Tab. 7 presents results obtained from training with both free-form and box masks at varying mask ratios. Each result corresponds to a specific mask setting and is evaluated under six different mask configurations - two mask types

Table 6. Evaluation results when trained on free-form mask with different ratios.

Mask Ratio	30%			50%			70%		
Metrics	FID ↓	SSIM ↑	PSNR ↑	FID ↓	SSIM ↑	PSNR ↑	FID ↓	SSIM ↑	PSNR ↑
DeFLOCNet [27]	23.50 (5.84)	0.76 (0.90)	20.00 (27.27)	**14.10** (7.58)	0.81 (0.85)	22.14 (25.56)	13.39 (11.47)	0.81 (0.80)	22.52 (23.60)
SC-FEGAN [18]	24.92 (5.51)	0.78 (0.92)	20.91 (28.24)	21.72 (9.48)	0.79 (0.86)	21.44 (25.39)	19.29 (13.83)	0.80 (0.79)	22.07 (23.14)
Gated Conv [48]	25.92 (3.76)	**0.80** (0.93)	**21.88** (29.28)	29.25 (14.34)	**0.82** (0.88)	21.90 (26.69)	27.38 (20.81)	0.82 (0.82)	**23.11** (24.60)
E2I [43]	96.64 (15.84)	0.67 (0.92)	17.45 (28.07)	63.92 (32.26)	0.72 (0.79)	17.08 (21.38)	84.53 (93.62)	0.70 (0.60)	16.14 (16.05)
Ours	**22.55** (**2.79**)	0.78 (**0.94**)	21.19 (**30.62**)	14.13 (**4.76**)	0.81 (**0.89**)	**22.49** (**27.99**)	**12.45** (**7.10**)	**0.83** (**0.84**)	22.98 (**25.73**)

Table 7. Evaluation results when trained on box mask with different ratios.

Mask Ratio	30%			50%			70%		
Metrics	FID ↓	SSIM ↑	PSNR ↑	FID ↓	SSIM ↑	PSNR ↑	FID ↓	SSIM ↑	PSNR ↑
DeFLOCNet [27]	34.35 (9.03)	0.74 (0.86)	20.30 (24.32)	25.52 (18.45)	0.77 (0.77)	21.39 (21.32)	26.59 (27.48)	0.76 (0.71)	20.28 (19.51)
SC-FEGAN [18]	**22.63** (7.77)	**0.77** (0.86)	**21.97** (24.11)	**22.6** (14.14)	0.77 (0.77)	21.99 (21.13)	28.12 (21.11)	0.76 (0.69)	21.01 (18.95)
Gated Conv [48]	23.51 (6.89)	**0.77** (0.87)	21.04 (24.69)	23.77 (12.09)	0.78 (0.79)	21.74 (21.67)	**23.01** (15.90)	0.78 (0.72)	**22.02** (21.11)
E2I [43]	124.49 (17.07)	0.62 (0.87)	16.14 (23.97)	91.16 (33.91)	0.67 (0.75)	16.47 (18.48)	110.89 (78.77)	0.66 (0.59)	16.02 (13.70)
Ours	29.26 (**5.02**)	0.76 (**0.89**)	20.84 (**26.17**)	25.47 (**8.36**)	**0.79** (**0.82**)	**22.00** (**23.63**)	27.47 (**11.55**)	**0.79** (**0.75**)	21.52 (**21.91**)

Table 8. Relationships to mask ratios and region of interests.

Ratio Type	Free-Form			Box		
mask-to-image	30%	50%	70%	30%	50%	70%
mask-to-foreground	34%	55%	75%	46%	74%	91%

(box and free-form), and three mask ratios (30%, 50%, 70%). Additionally, results evaluated with mask configurations that align with the training settings are also included in the brackets.

Overall, when the evaluation mask aligns the training settings, our methods achieve the best performance (see from the brackets results). When evaluated in all kinds of masks, we find it is beneficial to increase training mask ratio as it brings improvement to most of the methods. This is because that most of the methods only calculate loss on reconstructed (masked) region. A greater mask area would lead to much gradient information. However, we find the gain is limited when trained on box mask with 70% area (performance decreased in the last three columns of Tab. 7). Since the box masks are continuous, it is much likely to present on the foreground of the image. Therefore, less information is provided when reconstructing the masked region. Tab. 8 shows the corresponding masked foreground region. 70% box mask would cover more than 90% foreground region. In conclusion, it is optimal to trained mask ratio with about 70% in the foreground which is free-form mask with 70% ratio or box mask with 50% ratio. Comparing with other SOTA methods, ours CoDE-GAN is robust when trained on free-form mask. Its FID achieves the best in 30% and 70% mask ratio. The difference when trained on 50% is less 1% to DeFLOCNet.

5 FAILURE CASE ANALYSIS

Fig. 9 provides two examples where CoDE-GAN encounters difficulties. CoDE-GAN's primary objective is to edit the shape of clothing content with textures consistent with the unmasked (reference) region. The model learns to automatically find the correct reference content through adversarial training. However, when the background content is distracting or similar to the foreground, our model struggles to perform satisfactorily. The first three columns in the example demonstrate the completion of the obscured coat and the removal of the bag against a brick background. While our

Fig. 9. Failure case in handling distracting background.

CoDE-GAN method performs better on reproducing the background brick texture, it unfortunately falls short in synthesizing convincing clothing textures. The synthesized textures in the regions of interest are overly influenced by the background, causing an undesired shift towards background textures instead of the intended clothing. The final three columns attempts to modify short pants into an A-line dress. The similarity in color between the clothing and the background presents a challenge for the model, resulting in a failure to identify the correct reference color, despite the provision of sketches to outline the shape of the dress. Consequently, the synthesized dress textures align more closely with those of the ground floor. Therefore, while CoDE-GAN generally exhibits effective performance across numerous samples, it continues to face challenges when handling images with visually intricate or ambiguous backgrounds.

6 COMPUTATION EFFICIENCY ANALYSIS

Fashion editing often requires interactions that users iteratively make modifications to a fashion item until they are satisfied with the final outcome. Therefore, the proposed method should be computationally efficient, enabling quick responses to user inputs. Tab. 9 compares the efficiency of our CoDE-GAN and the baseline model.

Table 9. Comparison of execution times for Training and Inference

	Training	Inference (GPU)	Inference (CPU)
Baseline	0.523 s	0.0127 s	0.1884 s
Ours CoDE-GAN	0.589 s	0.0145 s	0.2098 s

The training time was calculated for one forward and backward pass with a batch size of 12, averaging 100 iterations on a single NVIDIA RTX 3090 GPU. For inference time, only a forward

pass on a single sample was measured, with the average taken over 100 iterations on the same GPU. For CPU inference, we used an Intel(R) Xeon(R) Gold 5220R CPU @ 2.20 GHz.

Compared with the baseline model, our CoDE-GAN slightly increases computational complexity by 12.6% in GPU training and 14.2% in GPU inference. However, given the 31.1% improvement in FID, this increase is worthwhile. Furthermore, in a pure CPU environment, our model responds to user input in approximately 0.2098 seconds, which is adequate for interactive uses.

7 INTERACTIVE WEB USER INTERFACE

Fig. 10. Interactive web user interface for allowing sketch-controlled editing.

Fig. 10 presents an interactive web user interface (UI), specifically designed to enable users to easily edit images. Users can upload their own images onto the web UI, and using the mask layer, they can draw a transparent black mask to mark the area they wish to edit. Following this, users can sketch their desired edits on top of the image using the sketch layer. Once the desired edits have been input, users can click the inference button, triggering our backend CoDE-GAN model to generate the edited results. This user-friendly web UI leverages Python as backend service, with basic HTML and JavaScript forming the front-end interface. Notably, our Web UI has the potential to be a valuable resource for other sketch-controlled methods, providing the community with a tool that's both accessible and easy to use. The UI's web-based nature reduces the need for user-side installations, allowing access via a web browser. Unlike existing model (SC-FEGAN) that packages the UI and the model together into an executable file exclusive for Windows systems, our web UI can be effortlessly adapted to MacOS without any need for code modification or compilation. Furthermore, the model deployed as a backend service provides flexibility for other developers. They can use the API in similar applications, test different models with minor adjustments, and choose whether to deploy the backend services locally or globally, given the availability of a public IP address. The hardware requirement is a minimum of 4GB runtime memory, providing further flexibility for deployment.

8 CONCLUSION AND FUTURE WORK

In conclusion, we have proposed a new method, CoDE-GAN, to allow for flexible sketch-controlled editing of fashion image content with consistent textures. Our approach decouples content and texture representation to overcome the obstacle of reconstructing content regions contoured by sketches due to the lack of information within them. We have shown through experiments on the fashion human ATR dataset and garment-based Garment and SG-Fashion datasets that CoDE-GAN achieves superior results compared to state-of-the-art methods in terms of perceptual quality and editing flexibility. CoDE-GAN has the potential to significantly streamline the image editing workflow in the fashion editing task, as it only requires users to provide sketches in the specific editing area rather than a full sketch of the entire image.

As in future work, there are several directions that can be explored to improve the CoDE-GAN method proposed in this study. One possibility is to integrate the use of additional guidance, such as texture patches, to edit the clothing textures. Another direction is to incorporate more advanced generative models, such as Generative Flow models or Denoising Diffusion models, to improve the quality of the generated images. Furthermore, it would be interesting to explore the use of CoDE-GAN for other applications beyond the fashion industry, such as image inpainting or guided image reconstruction. Finally, it would be valuable to further evaluate the performance of CoDE-GAN on a wider range of datasets beyond the fashion dataset and the church dataset to demonstrate its generalizability.

ACKNOWLEDGMENTS

The work described in this paper was supported in part by the Innovation and Technology Commission of Hong Kong under grant ITP/028/21TP and by the Research Institute of Intelligent Wearable Systems under grant P0049355/CD95.

REFERENCES

[1] Alberto Baldrati, Davide Morelli, Giuseppe Cartella, Marcella Cornia, Marco Bertini, and Rita Cucchiara. 2023. Multimodal garment designer: Human-centric latent diffusion models for fashion image editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 23393–23402.

[2] Serge Belongie, Jitendra Malik, and Jan Puzicha. 2002. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence* 24, 4 (2002), 509–522.

[3] Ankan Kumar Bhunia, Salman Khan, Hisham Cholakkal, Rao Muhammad Anwer, Jorma Laaksonen, Mubarak Shah, and Fahad Shahbaz Khan. 2023. Person image synthesis via denoising diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5968–5976.

[4] Jianbo Chen, Yelong Shen, Jianfeng Gao, Jingjing Liu, and Xiaodong Liu. 2018. Language-based image editing with recurrent attentive models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8721–8729.

[5] Lele Chen, Justin Tian, Guo Li, Cheng-Haw Wu, Erh-Kan King, Kuan-Ting Chen, Shao-Hang Hsieh, and Chenliang Xu. 2020. TailorGAN: making user-defined fashion designs. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 3241–3250.

[6] Seunghwan Choi, Sunghyun Park, Minsoo Lee, and Jaegul Choo. 2021. Viton-hd: High-resolution virtual try-on via misalignment-aware normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14131–14140.

[7] Aiyu Cui, Daniel McKee, and Svetlana Lazebnik. 2021. Dressing in order: Recurrent person image generation for pose transfer, virtual try-on and outfit editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14638–14647.

[8] Qiyu Dai, Shuai Yang, Wenjing Wang, Wei Xiang, and Jiaying Liu. 2021. Edit Like A Designer: Modeling Design Workflows for Unaligned Fashion Editing. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3492–3500.

[9] Prafulla Dhariwal and Alexander Quinn Nichol. 2021. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.). https://openreview.net/forum?id=AAWuCvzaVt

[10] Anna Frühstück, Krishna Kumar Singh, Eli Shechtman, Niloy J Mitra, Peter Wonka, and Jingwan Lu. 2022. InsetGAN for Full-Body Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7723–7732.

[11] Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R Scott. 2019. Clothflow: A flow-based model for clothed person generation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 10471–10480.

[12] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. 2018. Viton: An image-based virtual try-on network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7543–7552.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.

[14] Bingwen Hu, Ping Liu, Zhedong Zheng, and Mingwu Ren. 2022. SPG-VTON: Semantic prediction guidance for multi-pose virtual try-on. *IEEE Transactions on Multimedia* 24 (2022), 1233–1246.

[15] Wenmin Huang, Weiqi Luo, Jiwu Huang, and Xiaochun Cao. 2024. SDGAN: Disentangling Semantic Manipulation for Facial Attribute Editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 2374–2381.

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.

[17] Shuhui Jiang, Jun Li, and Yun Fu. 2021. Deep learning for fashion style generation. *IEEE Transactions on Neural Networks and Learning Systems* 33, 9 (2021), 4538–4550.

[18] Youngjoo Jo and Jongyoul Park. 2019. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1745–1753.

[19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 694–711.

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

[21] Hyug Jae Lee, Rokkyu Lee, Minseok Kang, Myounghoon Cho, and Gunhan Park. 2019. LA-VITON: A network for looking-attractive virtual try-on. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*. 0–0.

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

[23] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. 2020. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7880–7889.

[24] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. 2019. Storygan: A sequential conditional gan for story visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6329–6338.

[25] Yuheng Li, Yijun Li, Jingwan Lu, Eli Shechtman, Yong Jae Lee, and Krishna Kumar Singh. 2021. Collaging class-specific gans for semantic image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14418–14427.

[26] Xiaodan Liang, Si Liu, Xiaohui Shen, Jianchao Yang, Luoqi Liu, Jian Dong, Liang Lin, and Shuicheng Yan. 2015. Deep human parsing with active template regression. *IEEE transactions on pattern analysis and machine intelligence* 37, 12 (2015), 2402–2414.

[27] Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, Jing Liao, Bin Jiang, and Wei Liu. 2021. Deflocnet: Deep image editing via flexible low-level controls. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10765–10774.

[28] Linlin Liu, Haijun Zhang, Qun Li, Jianghong Ma, and Zhao Zhang. 2023. Collocated Clothing Synthesis with GANs Aided by Textual Information: A Multi-Modal Framework. *ACM Transactions on Multimedia Computing, Communications and Applications* 20, 1 (2023), 1–25.

[29] Yu Liu, Mingbo Zhao, Zhao Zhang, Yuping Liu, and Shuicheng Yan. 2024. Arbitrary Virtual Try-on Network: Characteristics Preservation and Tradeoff between Body and Clothing. *ACM Transactions on Multimedia Computing, Communications and Applications* 20, 5 (2024), 1–23.

[30] Furong Ma, Guiyu Xia, and Qingshan Liu. 2021. Spatial consistency constrained GAN for human motion transfer. *IEEE Transactions on Circuits and Systems for Video Technology* 32, 2 (2021), 730–742.

[31] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=B1QRgziT-

[32] Davide Morelli, Alberto Baldrati, Giuseppe Cartella, Marcella Cornia, Marco Bertini, and Rita Cucchiara. 2023. LaDI-VTON: Latent Diffusion Textual-Inversion Enhanced Virtual Try-On. *arXiv preprint arXiv:2305.13501* (2023).

[33] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. 2019. EdgeConnect: Structure Guided Image Inpainting using Edge Prediction. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*.
[34] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*. PMLR, 8162–8171.
[35] Shougan Pan, Zhengwentai Sun, Chenxing Wang, and Junkai Zhang. 2024. A 3D Virtual Try-On Method with Global-Local Alignment and Diffusion Model. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4415–4419.
[36] Yurui Ren, Ge Li, Shan Liu, and Thomas H Li. 2020. Deep spatial transformation for pose-guided person image generation and animation. *IEEE Transactions on Image Processing* 29 (2020), 8622–8635.
[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10684–10695.
[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=St1giarCHLP
[39] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. 2021. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems* 34 (2021), 1415–1428.
[40] Zhengwentai Sun, Yanghong Zhou, Honghong He, and PY Mok. 2023. Sgdiff: A style guided diffusion model for fashion synthesis. In *Proceedings of the 31st ACM International Conference on Multimedia*. 8433–8442.
[41] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. 2018. Toward characteristic-preserving image-based virtual try-on network. In *Computer Vision–ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part XIII*. 607–623.
[42] Saining Xie and Zhuowen Tu. 2015. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*. 1395–1403.
[43] Shunxin Xu, Dong Liu, and Zhiwei Xiong. 2020. E2I: Generative inpainting from edge to image. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 4 (2020), 1308–1322.
[44] Han Yan, Haijun Zhang, and Zhao Zhang. 2023. Learning to Disentangle the Colors, Textures, and Shapes of Fashion Items: A Unified Framework. *IEEE Transactions on Multimedia* (2023).
[45] Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, and Ping Luo. 2020. Towards photo-realistic virtual try-on by adaptively generating-preserving image content. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7850–7859.
[46] Shuai Yang, Zhangyang Wang, Jiaying Liu, and Zongming Guo. 2020. Deep plastic surgery: Robust and controllable image editing with human-drawn sketches. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*. Springer, 601–617.
[47] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. 2015. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. *arXiv preprint arXiv:1506.03365* (2015).
[48] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2019. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4471–4480.
[49] Xiaoyun Yuan, Difei Tang, Yebin Liu, Qing Ling, and Lu Fang. 2016. Magic glasses: from 2D to 3D. *IEEE Transactions on Circuits and Systems for Video Technology* 27, 4 (2016), 843–854.
[50] Yu Zeng, Zhe Lin, and Vishal M Patel. 2022. Sketchedit: Mask-free local image manipulation with partial sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5951–5961.
[51] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3836–3847.
[52] Pengze Zhang, Lingxiao Yang, Xiaohua Xie, and Jianhuang Lai. 2021. Lightweight Texture Correlation Network for Pose Guided Person Image Generation. *IEEE Transactions on Circuits and Systems for Video Technology* (2021).
[53] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. 2024. Uni-controlnet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems* 36 (2024).
[54] Dongliang Zhou, Haijun Zhang, Jianghong Ma, Jicong Fan, and Zhao Zhang. 2023. Fcboost-net: A generative network for synthesizing multiple collocated outfits via fashion compatibility boosting. In *Proceedings of the 31st ACM International Conference on Multimedia*. 7881–7889.
[55] Luyang Zhu, Dawei Yang, Tyler Zhu, Fitsum Reda, William Chan, Chitwan Saharia, Mohammad Norouzi, and Ira Kemelmacher-Shlizerman. 2023. TryOnDiffusion: A Tale of Two UNets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4606–4615.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009