TailorEdit: An Adaptive Framework for Instruction-Guided Fashion Image Editing

Xiaoling Gu, Lingda Zhu, Yongkang Wong, Zhou Yu, Huan Li, Zizhao Wu, and Mohan S. Kankanhalli, *Fellow, IEEE*

Fig. 1: *TailorEdit* supports four common fashion image editing tasks. Given multi-modal input conditions, TailorEdit adaptively identifies the appropriate editing task and generates high-fidelity, visually consistent results.

***Abstract*—Fashion image editing has garnered significant attention due to its growing demand in e-commerce, social media, and virtual try-on applications. However, existing methods are typically designed for specific editing tasks in isolation, lacking a unified framework capable of handling diverse editing requirements. This work addresses this limitation from two critical perspectives. First, we construct *InstructFashion*, a large-scale, high-quality dataset specifically curated for instruction-guided fashion image editing. It is generated through carefully designed pipelines that cover four distinct editing tasks. Second, we propose *TailorEdit*, an adaptive framework for instruction-guided fashion image editing. It integrates human segmentation map-based denoising guidance, modular LoRA-based editing experts, and a dynamic expert routing mechanism to enable precise and semantically coherent modifications. Extensive quantitative and qualitative evaluations demonstrate that TailorEdit consistently outperforms state-of-the-art methods in terms of realism, coherence, and instruction adherence. Our code is available at https://github.com/EndaJude/TailorEdit.**

***Index Terms*—Fashion Image Editing, Instruction-Driven, LoRA**

Xiaoling Gu, Lingda Zhu, Zhou Yu, and Zizhao Wu are with the Zhejiang Key Laboratory of Space Information Sensing and Transmission, School of Computer Science and Technology, Hangzhou Dianzi University, Zhejiang, China (e-mail: guxl@hdu.edu.cn; endajude@hdu.edu.cn; yuz@hdu.edu.cn; wuzizhao@hdu.edu.cn). Huan Li is with the School of Computer Science and Technology, Zhejiang University, Zhejiang, China (e-mail: lihuan.cs@zju.edu.cn). Yongkang Wong and Mohan S. Kankanhalli are with the School of Computing, National University of Singapore, Singapore (e-mail: yongkang.wong@nus.edu.sg; mohan@comp.nus.edu.sg). Zhou Yu is the corresponding author.

Manuscript received April 19, 2021; revised August 16, 2021.

I. Introduction

Due to the rising demand in e-commerce, social media, and virtual try-on applications, fashion image editing has received widespread attention from both industry and academia [1], [2]. Existing methodologies primarily fall into three distinct categories: image-based editing [3]–[7], text-driven editing [8], [9], and multi-modal conditioned editing [10], [11]. However, these algorithms are typically designed independently for specific tasks. This task-specific isolation leads to three key limitations: (1) redundant development efforts across similar editing paradigms, as each new task often requires redesigning dedicated models; (2) inability to leverage shared knowledge across tasks, limiting generalization of learned representations; and (3) increased complexity for end-users who must navigate and switch between disparate tools.

In this work, we aim to develop a unified framework for supporting diverse fashion image editing tasks. To ensure broad applicability, we conducted a comprehensive analysis of existing fashion editing scenarios and identified four core editing tasks: *Addition*, *Removal*, *Replacement* and *Alteration*. These tasks encompass the majority of practical use cases in fashion-related applications, such as adding and removing accessories, replacing clothing items, or modifying specific attributes (*e.g.*, color, material, or shape).

However, integrating multiple editing tasks within a unified generative framework presents several challenges. First, existing datasets (*e.g.*, Fashion-Gen [12], DeepFashion-MultiModal [13], and Viton-HD Multimodal [11]) fail to meet the demands of multi-task fashion editing. Constructing a high-

quality dataset that comprehensively covers various editing tasks poses a substantial challenge. Second, the heterogeneous input conditions require an adaptive architecture capable of harmonizing cross-modal signals. Third, learning objectives across tasks necessitate a flexible framework with a generalized training strategy. This approach must balance task-specific constraints and ensures cross-task compatibility.

We address these challenges from two distinct perspectives. Firstly, we construct a large-scale, well-annotated, high-quality fashion image dataset, named *InstructFashion*. The images are sourced from online commercial fashion platforms and processed through carefully designed data generation pipelines. High data quality was achieved through rigorous manual filtering, culminating in a final dataset of 41,324 high-fidelity image samples. Each sample comprises (1) an original human image along with its human segmentation map, (2) an editing instruction, (3) a target human image reflecting the edited result along with its human segmentation map, and (4) a reference image of the garment or accessory provided specifically for the *Addition* and *Replacement* tasks.

Secondly, we propose *TailorEdit*, an adaptive framework designed for diverse fashion image editing tasks. As illustrated in Fig. 1, given specific input conditions, TailorEdit can dynamically identify the appropriate editing task and generate high-fidelity, visually consistent results. These tasks are guided by multi-modal conditions, which consist of a source image and an instruction. For *Addition* and *Replacement* tasks, an additional reference image is required. The key idea of TailorEdit is to leverage a set of Low-Rank Adaptation (LoRA) [14] experts, each trained for a specific editing task, where each expert independently captures task-specific editing patterns. TailorEdit consists of three core components: (1) **Human Semantic Guidance**. To maintain human semantic consistency during editing, we adopt the *human segmentation map as denoising guidance*. This is implemented through a ControlNet, built upon a pre-trained InstructPix2Pix (IP2P) model [15] and trained on a multi-task dataset covering four distinct editing operations. (2) **Adaptive Attention Module**. The base network is initialized from the pre-trained IP2P model and incorporates the previously trained ControlNet, along with an adaptive attention module for precise editing. To enable task-aware adaptation, we train four *low-rank editing experts* by integrating LoRA adapters into the self-attention and cross-attention layers of the U-Net architecture. Each expert is independently fine-tuned on a single-task dataset, ensuring modularity and adaptability. (3) **Dynamic Expert Routing**. To efficiently leverage the specialized experts, we introduce a *dynamic expert routing mechanism*. First, a global gating module analyzes the input instruction to identify the target editing task and activates the appropriate expert. Then, a local gating module adaptively modulates the selected expert's contribution across different U-Net layers, enabling fine-grained control over its influence.

We summarize our contributions as follows: (1) We construct *InstructFashion*, a large-scale, well-annotated, high-quality dataset specifically designed for instruction-guided fashion image editing. It addresses the critical need for diverse and comprehensive data to advance the development and evaluation of editing models. (2) We propose *TailorEdit*, a novel adaptive framework for instruction-guided fashion image editing. This unified approach seamlessly supports multiple image editing tasks, making it a versatile solution for various fashion editing scenarios. (3) Extensive quantitative and qualitative experiments validate the effectiveness of TailorEdit, consistently showing that it outperforms state-of-the-art methods in terms of realism, coherence, and instruction adherence.

II. Related Work

A. Fashion Image Editing

Contemporary research in fashion image editing can be categorized into three primary paradigms: image-based editing [3]–[7], text-driven editing [8], [9], [16], and multi-modal conditioned editing [10], [11], [17]. For example, previous studies [3], [4] have developed methods for image-based editing, transferring garments from a reference image to a target individual. Günel *et al*. [8] leveraged feature-wise linear modulation to manipulate fashion images based on textual descriptions, while Dong *et al*. [10] proposed FE-GAN, which enables fashion image editing using free-form sketches and sparse color strokes. Early fashion image editing approaches predominantly relied on Generative Adversarial Networks (GANs) [18]. However, with diffusion models [19] surpassing GANs in training stability, generation quality, and diversity, recent fashion image editing methods have shifted towards diffusion-based architectures [5]–[7], [9], [11]. For example, advancements in image-based editing [5]–[7] have leveraged diffusion models to enhance fine-grained feature consistency. On the text-driven side, Wang and Ye [9] introduced TexFit for local fashion image editing via textual prompts. In the multi-modal paradigm, Baldrati *et al*. [11] proposed Ti-MGD, a multi-modal conditioned fashion image editing approach that integrates multiple conditioning factors into the generation process. In contrast to previous approaches, our work introduces an adaptive framework for fashion image editing and presents a high-quality dataset specifically designed for multi-task fashion image editing.

B. Instruction-guided Image Editing

Instruction-guided image editing interprets editing intent through concise instructions [20]–[22]. Unlike textual prompts, instructions more closely align with user intent and provide a more intuitive way to specify editing goals. InstructPix2Pix (IP2P) [15] pioneered the use of instructions to guide image editing in diffusion models, yet it often inadvertently alters unintended areas. To address these issues, recent studies emphasize identifying target regions and constraining the generation process. For instance, Li *et al*. [23] introduced ZONE, a zero-shot instruction-guided local image editing method that leverages diffusion models' localization capabilities to infer editing regions, eliminating the need for explicit guidance. Guo and Lin [24] proposed FoI, which enables precise, region-specific multi-instruction editing without requiring additional training or test-time optimization. Geng *et al*. [25] introduced InstructDiffusion, a unified framework that transforms diverse visual tasks as intuitive image modifications guided by human

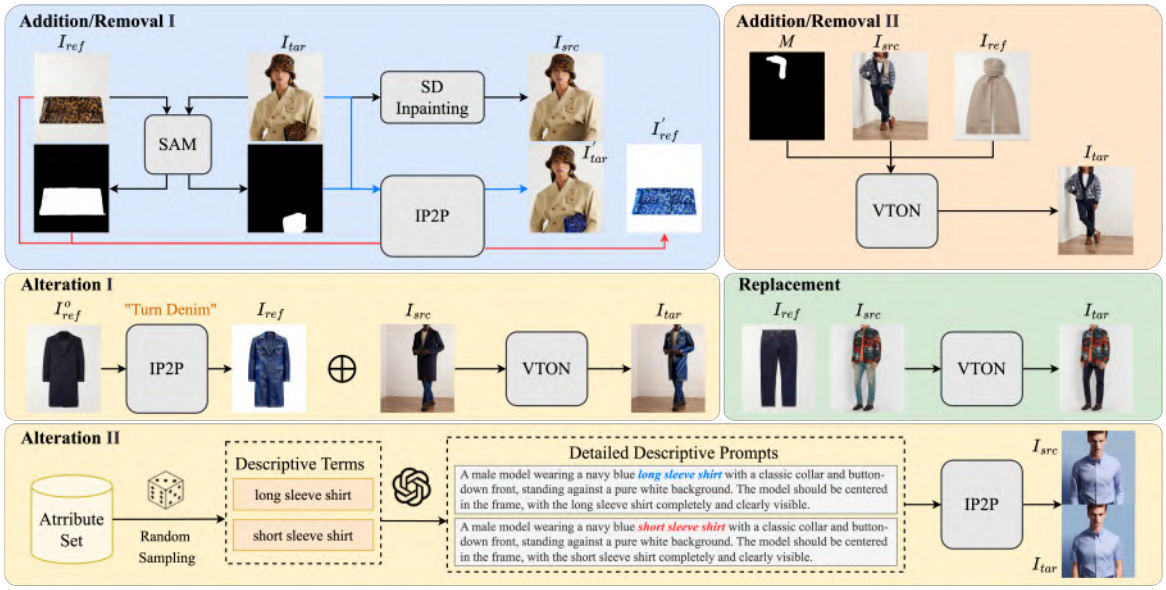


Fig. 2: Data generation pipelines for four distinct editing tasks. We first collect samples for the *Addition* task. Then, samples for the *Removal* task are derived by simply swapping the source and target images from the *Addition* task. Note that both the *Addition* and *Alteration* tasks utilize two types of generation pipelines.

instructions. In contrast to these approaches, our work introduces an instruction-guided fashion image editing framework that seamlessly supports multiple editing tasks, offering a versatile solution for various fashion editing scenarios.

III. The InstructFashion Dataset

Currently, there exists no dataset that fully meets the requirements of multi-task fashion editing. To address this challenge, we introduce *InstructFashion*, a large-scale, well-annotated, high-quality fashion image dataset designed to support diverse editing tasks. The original images are gathered from commercial fashion platforms (*i.e.*,, NetAPorter, Shopbop), each annotated with a category label. We develop specialized data generation pipelines for four distinct editing tasks, as illustrated in Fig. 2. The details of each editing task are as follows:

- **Addition Task.** For accessories such as bags and belts, we first select a reference image I_{ref} and a target image I_{tar}. Next, we employ an image segmentation model [26] to extract masks from both I_{ref} and I_{tar}. Using the mask extracted from I_{tar}, we apply the inpainting function of Stable Diffusion (SD) [27] to I_{tar} for removing the accessory and generating a source image I_{src} with the accessory removed. To further enhance dataset diversity, we modify the accessory's color in both I_{ref} and I_{tar}. Specifically, given an image pair (I_{ref}, I_{tar}), we use IP2P model [15] to alter the accessory's color, producing a new image pair. For accessories such as scarves and sunglasses, we adopt a different approach. First, we manually annotate a mask for the accessory from I_{src}. Then, using the IDM-VTON [28], we transfer the accessory from I_{ref} onto the model in I_{src}, generating a new target image I_{tar}.
- **Removal Task.** The data samples for the *Removal* task are constructed by simply swapping I_{src} and I_{tar} from the *Addition* task.
- **Replacement Task.** We employ IDM-VTON [28] to generate the target image I_{tar} based on I_{src} and I_{ref}.
- **Alteration Task.** For attribute modifications such as color and material, we first use the IP2P model [15] to adjust the color or material of the original reference image $I^{\text{o}}_{\text{ref}}$, producing an updated reference image I_{ref}. Then, leveraging I_{ref} and the source image I_{src}, we apply the IDM-VTON [28] to generate the target image I_{tar}. For shape modifications, we generate edited image pairs (I_{src}, I_{tar}) using the IP2P model. First, we define a set of shape attributes. Then, we sample two descriptive terms, such as "long sleeve shirt" and "short sleeve shirt". These terms are expanded into detailed descriptive prompts using Large Language Models (*e.g.*, GPT-3.5). Finally, based on these detailed descriptions, we follow the dataset generation workflow in IP2P to produce the edited image pairs (I_{src}, I_{tar}).

We initially obtained 150K image pairs or triplets from the aforementioned data generation pipelines, but many were of low quality. We manually filtered out these samples based on two main criteria: (1) whether the edited image semantically aligns with the instruction, and (2) whether the generated image preserves fine-grained details. We recruited five graduate students who are highly familiar with the image editing domain, and they spent about two weeks completing the data filtering process. To maintain dataset reliability, they carefully examined key aspects such as the face, limbs, and clothing.

Finally, we curated a dataset of 41,324 multi-modal samples, comprising 10,504 samples each for the *Addition* and *Removal* tasks, 10,244 samples for the *Replacement* task, and 10,072 samples for the *Alteration* task. Each sample consists of an original image and its associated human parsing, an editing

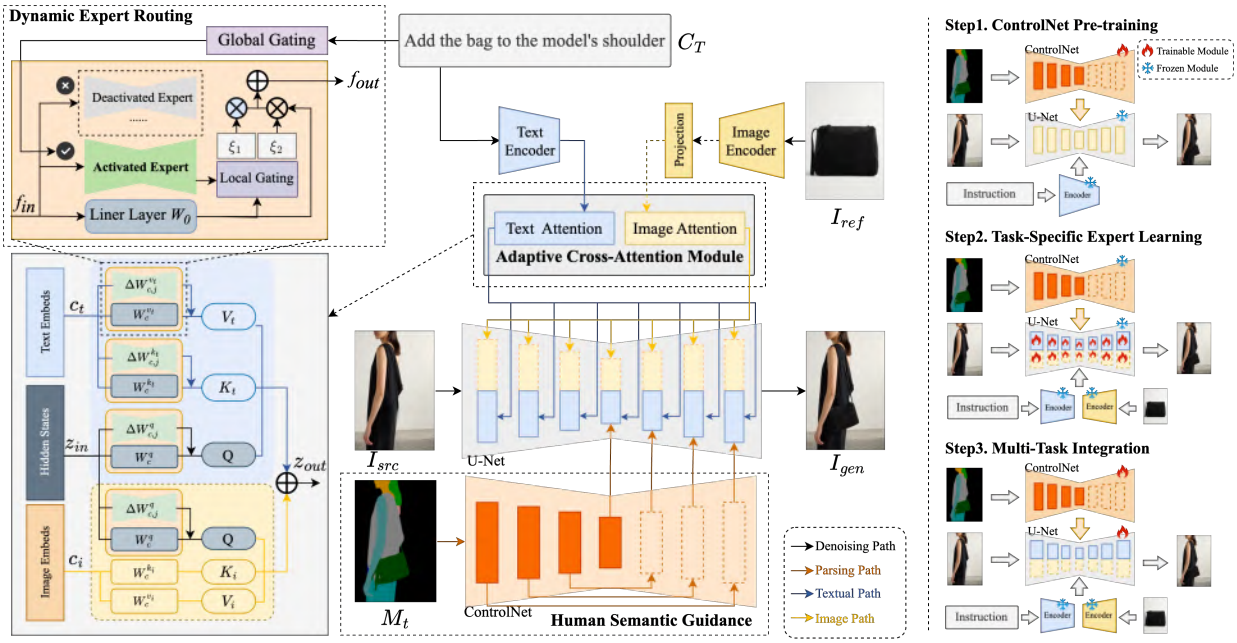


Fig. 3: **Left**: The overall architecture of *TailorEdit*. For clarity, only the structure of the Adaptive Cross-Attention Module is illustrated; the Adaptive Self-Attention Module shares a similar design and is omitted. **Right**: An overview of the three-stage training strategy used for *TailorEdit*.

instruction, and a target image with its corresponding human parsing. Additionally, a reference image is provided for the *Addition* and *Removal* tasks. The editing instructions are generated using GPT-3.5. To obtain segmentation maps for both I_{src} and I_{tar}, we employ a human parser [29] to generates 18 semantic labels, such as skin, left hand, sunglasses, pants, *etc*.

IV. The Proposed TailorEdit

A. Preliminaries

Built on the Stable Diffusion (SD) [27], IP2P [15] leverages the power of latent diffusion models to enable instruction-based image editing. For an input image x, the diffusion process first encodes it into a latent representation $z = \varepsilon(x)$. Noise $\epsilon \sim \mathcal{N}(0,1)$ is then progressively added to z, resulting in a noisy latent z_t at timestep $t \in T$, where the noise level increases over time. IP2P trains a denoising network ϵ_θ, initialized from SD weights, to predict the original noise from z_t, conditioned on both the image (C_I) and the textual instruction (C_T). The network ϵ_θ is fine-tuned by minimizing the following latent diffusion objective:

$$\mathcal{L}_{\text{IP2P}} = \mathbb{E}_{z,t,\epsilon\sim\mathcal{N}(0,1)}\left[\left\|\epsilon - \epsilon_\theta(z_t, t, C_I, C_T)\right\|_2^2\right] \quad (1)$$

IP2P adopts classifier-free guidance with two conditioning signals:

$$\begin{aligned}\hat{\epsilon}_\theta(z_t, t, C_I, C_T) =& \epsilon_\theta(z_t, t, \emptyset_I, \emptyset_T) \\ &+ \alpha_I \cdot (\epsilon_\theta(z_t, t, C_I, \emptyset_I) - \epsilon_\theta(z_t, t, \emptyset_I, \emptyset_T)) \\ &+ \alpha_T \cdot (\epsilon_\theta(z_t, t, C_I, C_T) - \epsilon_\theta(z_t, t, C_I, \emptyset_T))\end{aligned} \quad (2)$$

During training, conditions are randomly omitted by setting $C_I = \emptyset_I$ or $C_T = \emptyset_T$ with a certain probability. Here, α_I and α_T denote the guidance scales that control the influence of image and text conditions, respectively, during the denoising process.

B. Overall Structure

Given a source image I_{src}, an editing instruction C_T, an optional reference image I_{ref}, and a target human segmentation map M_t, TailorEdit generates an edited image I_{gen} that accurately reflects the given instruction while preserving visual coherence and realism. Fig. 3 illustrates the overall structure of TailorEdit, which mainly consists of four components: feature extraction, human semantic guidance, an adaptive attention module and dynamic expert routing.

1) Feature Extraction: To encode textual instructions, we employ a pre-trained CLIP [30] text encoder. For *Addition* and *Replacement* tasks, where reference images are required, we also utilize a pre-trained CLIP image encoder to extract visual features from the reference inputs. To align the extracted image features with the model's latent feature space, we use a pre-trained projection network [31] consisting of a linear transformation followed by Layer Normalization.

2) Human Semantic Guidance: Maintaining human semantic consistency is crucial for achieving realistic fashion image editing results. A primary challenge is ensuring precise alignment between fashion items and the human body while preserving structural integrity across various editing tasks. To address this, we incorporate the human segmentation map as denoising guidance, leveraging its ability to capture detailed body structure and constrain the spatial arrangement of fashion items. Specifically, we develop a ControlNet based on a pre-trained IP2P model, augmenting it with a human segmentation

map as an additional conditioning signal during the denoising process.

3) Adaptive Attention Module: To achieve specialized and efficient editing, we introduce an adaptive attention module. It facilitates the training of four low-rank editing experts, each tailored to a specific editing task using customized datasets. These experts are designed to independently capture task-specific editing patterns, enhancing the model's ability to perform fine-grained modifications. For each editing expert, we incorporate LoRA adapters into the cross-attention layers of the U-Net. This allows for lightweight and efficient adaptation to different editing instructions. The outputs from both text-based and image-based cross-attention are subsequently fused to update the latent representations:

$$\begin{aligned} z_{\text{out}} &= \text{Attention}(Q, K_t, V_t) + \text{Attention}(Q, K_i, V_i), \\ \text{where} &\begin{cases} Q = (W_c^q + \Delta W_{c,j}^q) z_{\text{in}} \\ K_t = (W_c^{k_t} + \Delta W_{c,j}^{k_t}) c_t; V_t = (W_c^{v_t} + \Delta W_{c,j}^{v_t}) c_t \\ K_i = W_c^{k_i} c_i; V_i = W_c^{v_i} c_i \end{cases} \end{aligned} \tag{3}$$

Here, W_c^q, $W_c^{k_t}$, $W_c^{v_t}$, $W_c^{k_i}$ and $W_c^{v_i}$ represent the weight matrices of the linear projection layers for queries, textual keys/values, and visual keys/values, respectively. The terms $\Delta W_{c,j}^q$, $\Delta W_{c,j}^{k_t}$, $\Delta W_{c,j}^{v_t}$ denote the task-specific LoRA adapter weights for the j-th editing task. c_t and c_i denote the textual and visual embeddings used for conditioning during the attention computation.

In addition, we integrate LoRA adapters into the self-attention layers of the U-Net to further enhance task-specific adaptability while maintaining computational efficiency. The self-attention mechanism is formulated as:

$$\begin{aligned} z'_{\text{out}} &= \text{Attention}(Q', K', V'), \\ where &\begin{cases} Q^{'} = (W_s^q + \Delta W_{s,j}^q) z'_{\text{in}} \\ K^{'} = (W_s^k + \Delta W_{s,j}^k) z'_{\text{in}} \\ V^{'} = (W_s^v + \Delta W_{s,j}^v) z'_{\text{in}} \end{cases} \end{aligned} \tag{4}$$

Here, W_s^q, W_s^k, W_s^v denote the weight matrices of the linear projection layers used to compute the queries, keys, and values in the self-attention mechanism. Correspondingly, $\Delta W_{s,j}^q$, $\Delta W_{s,j}^k$ and $\Delta W_{s,j}^v$ represent the task-specific LoRA adapter weights for the j-th editing task.

4) Dynamic Expert Routing: Based on the learned editing experts, we introduce a *dynamic expert routing mechanism*. First, a global gating module analyzes the input instruction to identify the appropriate editing task and activates the relevant expert. Then, a local gating module dynamically adjusts the expert's contribution across different layers, enabling fine-grained control over its influence.

Given an input editing request, TailorEdit first analyzes the textual instruction C_T to determine the most relevant editing task. This decision process is performed using a lightweight global gating module G_{global}, which is formulated as:

$$g = \text{Softmax}\left(G_{\text{global}}(\text{CLIP}(C_T))\right) \tag{5}$$

where $\text{Softmax}()$ is the softmax function, $\text{CLIP}(C_T)$ is the text embedding extracted with CLIP [30], and $g \in \mathbb{R}^4$ represents the four impact scales for the four experts. Finally, the editing expert with the highest scale value is activated. The global gating module is implemented as a multi-layer perceptron (MLP) and is pre-trained using a task classification loss on the multi-task dataset.

Once the appropriate editing expert is activated, a local gating module G_{local} dynamically adjusts the expert's contribution across different layers, as an expert may exert varying strengths of guidance at each layer. G_{local} calculates the dynamic weights $\xi = [\xi_1, \xi_2]$ for the outputs of the linear projection layer and the activated LoRA expert, which is formulated as:

$$\begin{aligned} \xi_1, \xi_2 &= \text{Softmax}\left(G_{\text{local}}(W_0 \cdot f_{\text{in}}, \Delta W_j \cdot f_{\text{in}})\right) \\ f_{\text{out}} &= \xi_1 W_0 \cdot f_{\text{in}} + \xi_2 \Delta W_j \cdot f_{\text{in}} \end{aligned} \tag{6}$$

where W_0 is the weight matrix of the linear projection layer and ΔW_j is the trained LoRA adapter weights. The local gating module is implemented with a linear projection layer. The integration of the global and local gating modules enhances the model's ability to adapt to diverse editing scenarios.

C. Training Strategy

The training process of TailorEdit consists of three steps: ControlNet Pre-training, Task-Specific Expert Learning, and Multi-Task Integration. In the first step, we pre-train the ControlNet using a mixed dataset that covers four distinct editing tasks. This enhances the model's generalization ability, ensuring robustness and adaptability across various fashion editing scenarios. During training, we use the human segmentation map of the target image as the ground truth. During inference, the segmentation map is generated by a pre-trained *Parsing Prediction Network*, allowing the model to operate effectively without the need for manual annotations. The Parsing Prediction Network is built upon a U-Net architecture, utilizing cross-entropy loss and IoU loss to segment the input fashion image into 18 semantic regions, such as skin, left hand, sunglasses, pants, *etc*. In the second step, we construct the entire network based on a pre-trained IP2P model, incorporating the previously trained ControlNet and the *adaptive attention module*. Each editing expert is fine-tuned independently on a single-task dataset, allowing for specialized learning while maintaining modularity. This process ensures that the model performs effectively across different editing tasks while retaining flexibility. In the third step, the ControlNet, alongside the U-Net and the local gating module, is fine-tuned on the multi-task dataset, further refining the model's ability to preserve human-body alignment and editing precision. The global gating module, which dynamically routes inputs to the appropriate editing task based on textual instructions, is pre-trained using a task classification loss on the multi-task dataset.

V. Experiments

A. Experimental Settings

1) Implementation Details : All images are resized to 512×512. The model is optimized using the AdamW optimizer with a fixed learning rate of $1e$-5. To enable classifier-free guidance, we randomly drop the text or image input with a probability of 5% each, and both inputs simultaneously with a

TABLE I: Quantitative results of different models on four editing tasks. For each baseline, the absolute improvement ((BETTER) or (WORSE)) when finetuned on InstructFashion dataset are shown. The best and suboptimal values are highlighted in bold and underlined, respectively.

Editing Task	Method	PSNR ↑	FID ↓	LPIPS↓	CLIP-I↑	DINO↑	CLIP-T↑
Removal	ZONE [23]	24.532	36.105	0.073	0.9176	0.980	0.2699
	InstructPix2Pix [15]	27.367 (+8.568)	20.653 (−69.490)	0.060 (−0.155)	0.9317 (+0.1378)	0.992 (+0.101)	0.2739 (+0.0469)
	MagicBrush [33]	27.982 (+5.558)	25.966 (−20.157)	0.069 (−0.043)	0.9266 (+0.0636)	0.985 (+0.020)	0.2707 (+0.0174)
	InstructDiffusion [25]	28.333 (+4.057)	22.449 (−14.638)	0.060 (−0.022)	0.9326 (+0.0384)	0.991 (+0.010)	0.2709 (−)
	TailorEdit (Ours)	**29.043**	**19.935**	**0.057**	**0.9389**	**0.993**	**0.2740**
Alteration	ZONE [23]	19.026	21.522	0.126	0.9116	0.975	0.2683
	InstructPix2Pix [15]	20.429 (+3.177)	19.721 (−9.400)	0.104 (−0.073)	0.9170 (+0.0315)	0.982 (+0.023)	0.2749 (+0.0142)
	MagicBrush [33]	19.947 (+2.890)	24.642 (−7.181)	0.116 (−0.044)	0.9094 (+0.0484)	0.972 (+0.014)	0.2726 (+0.0193)
	InstructDiffusion [25]	**24.547** (+7.443)	**15.311** (−8.659)	**0.066** (−0.101)	**0.9320** (+0.0433)	**0.991** (+0.025)	0.2774 (+0.0174)
	TailorEdit (Ours)	19.908	19.101	0.124	0.9153	0.977	**0.2781**
Addition	PaintByExample [34]	23.765 (−0.606)	26.697 (+2.960)	0.070 (−)	0.9192 (−0.0034)	0.987 (−0.002)	0.2708 (−0.0079)
	AnyDoor [35]	24.740 (+1.385)	22.224 (−4.107)	0.082 (+0.014)	0.9439 (+0.0092)	0.982 (−0.001)	**0.2817** (+0.0019)
	ObjectStitch [36]	24.543 (+1.253)	23.850 (−5.580)	0.072 (−0.011)	0.9315 (+0.0162)	0.983 (+0.001)	0.2760 (+0.0009)
	OOTDiffusion [37]	25.774	20.766	**0.047**	**0.9486**	0.990	0.2791
	TailorEdit (Ours)	**26.993**	**18.314**	0.055	0.9407	**0.994**	0.2784
Replacement	PaintByExample [34]	17.457 (−0.865)	35.236 (+1.661)	0.117 (−0.016)	0.8835 (+0.0205)	0.973 (+0.012)	0.2523 (−0.0071)
	AnyDoor [35]	17.456 (+1.196)	24.520 (−6.163)	0.137 (−0.005)	0.8897 (+0.0136)	0.971 (+0.010)	0.2770 (+0.0153)
	ObjectStitch [36]	17.787 (+1.556)	22.045 (−5.556)	0.111 (−0.026)	0.9008 (+0.0013)	0.977 (+0.009)	0.2706 (+0.0054)
	OOTDiffusion [37]	20.513	25.736	**0.074**	**0.9339**	0.984	**0.2792**
	TailorEdit (Ours)	**20.539**	**19.123**	0.085	0.9153	**0.986**	0.2733

probability of 5%. The training process of TailorEdit consists of three steps: *ControlNet Pre-training*, *Task-Specific Expert Learning* and *Multi-Task Integration*. In the first step, we pre-train ControlNet on a mixed dataset that spans four distinct editing tasks, using a batch size of 16 for 10k steps. In the second step, we build the complete model upon a pre-trained IP2P model [15], incorporating the previously trained ControlNet and the adaptive attention module. Each task-specific expert is fine-tuned independently on its corresponding single-task dataset with a batch size of 8 for 120K steps. In the final step, we fine-tune the ControlNet, U-Net, and local gating module jointly on the multi-task dataset with a batch size of 8 for 220K steps. During inference, we adopt the DDIM sampler [32] with 50 steps and set the guidance scale to 2. All experiments are conducted on our *InstructFashion* dataset.

All experiments are conducted on NVIDIA RTX 4090 GPUs. Under our experimental configuration, a single RTX 4090 with 24 GB of VRAM is sufficient to meet all computational requirements. Training time varies across stages: the first stage requires approximately 20 GPU hours for ControlNet convergence; the second stage, which trains four expert modules, consumes an average of 16 GPU hours per expert to achieve stable editing performance; and the third stage requires around 40 GPU hours to fine-tune the full architecture to its optimal state.

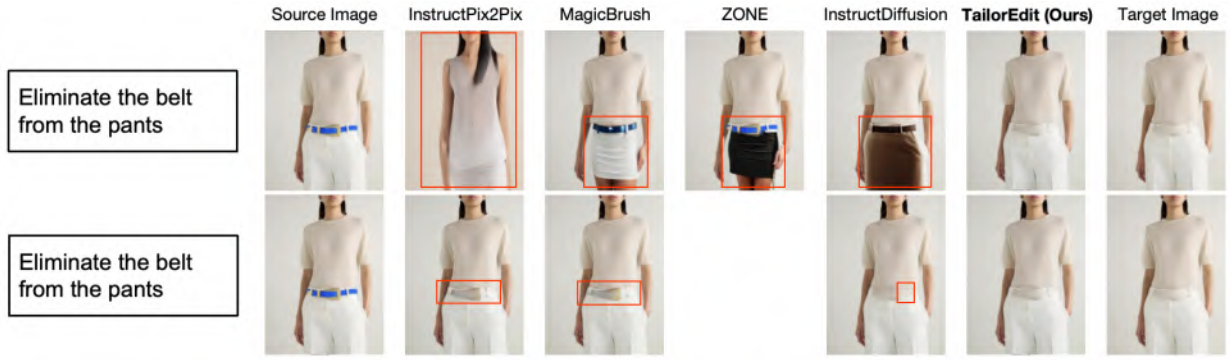
Due to the introduction of four expert modules and a decoupled attention module, the U-Net contains approximately 1B parameters, while the ControlNet component remains at 340M parameters. Inference time depends on several factors; on a single NVIDIA RTX 4090 GPU, with a batch size of 1 and 30 diffusion steps, our model requires about 5 seconds per image, corresponding to roughly 48 TFLOPs of computation.

2) Baselines and Metrics: We divide the four editing tasks into two groups (with or without reference images I_{ref}) and evaluate them against appropriate baselines. For the *Removal* and *Alteration* tasks, we compare our approach with four representative instruction-driven image editing methods: IP2P [15], MagicBrush [33], ZONE [23], and InstructDiffusion [25]. For the *Addition* and *Replacement* tasks, we select three representative reference-based image generation methods as baselines: AnyDoor [35], ObjectStitch [36], and PaintByExample [34]. Furthermore, owing to the resemblance to virtual try-on scenarios, we include OOTDiffusion [37] as an additional baseline. To ensure a fair comparison, all baseline methods that support fine-tuning were trained on our *InstructFashion* dataset, with the best-performing checkpoints selected based on evaluation metrics. For methods that do not support fine-tuning, we used their official public checkpoints.

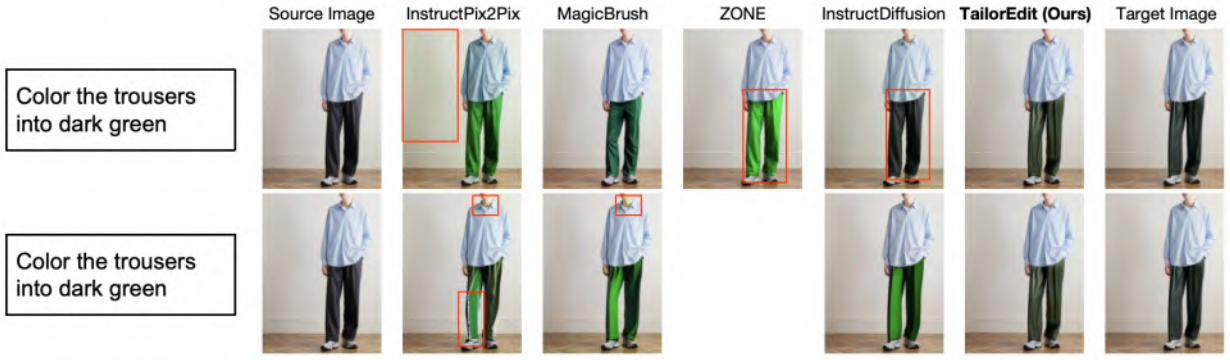
We evaluate our model using six metrics: PSNR [38], FID [39], LPIPS [40], CLIP-I [41], DINO [42], and CLIP-T [41]. Three of these metrics assess image quality: PSNR measures pixel-level differences between edited and target images; FID captures distributional differences in deep features; and LPIPS quantifies perceptual similarity. Subject fidelity is assessed using CLIP-I and DINO, which measure spatial layout consistency and semantic alignment between the edited and target images. Finally, text fidelity is evaluated using CLIP-T, which measures the alignment between the edited images and their corresponding textual prompts generated by GPT-4o.

B. Main Results

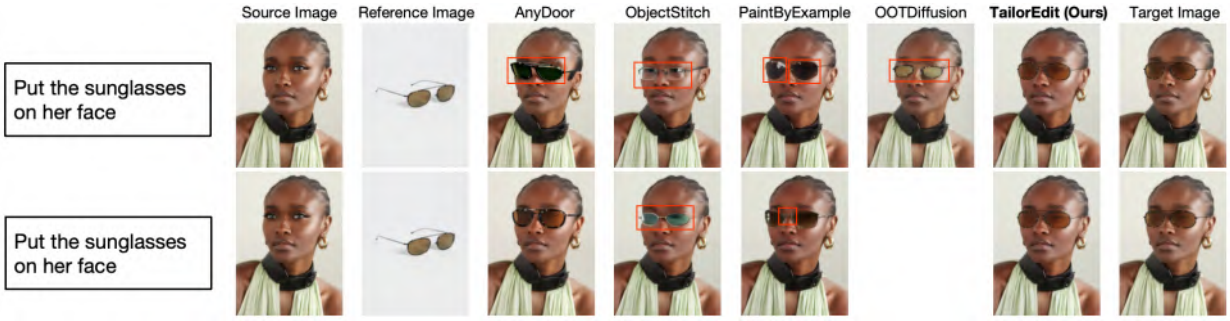
1) Quantitative Analysis: Table I presents a comprehensive quantitative comparison between our proposed method and several baseline approaches. For each baseline, we report performance both before and after fine-tuning on the *InstructFashion* dataset to assess their generalization capabilities. For metrics where *higher is better* (e.g., PSNR), a value shown as "X (+Y)" indicates a post–fine-tuning score of X, with an improvement of Y after fine-tuning (marked in blue), while "X



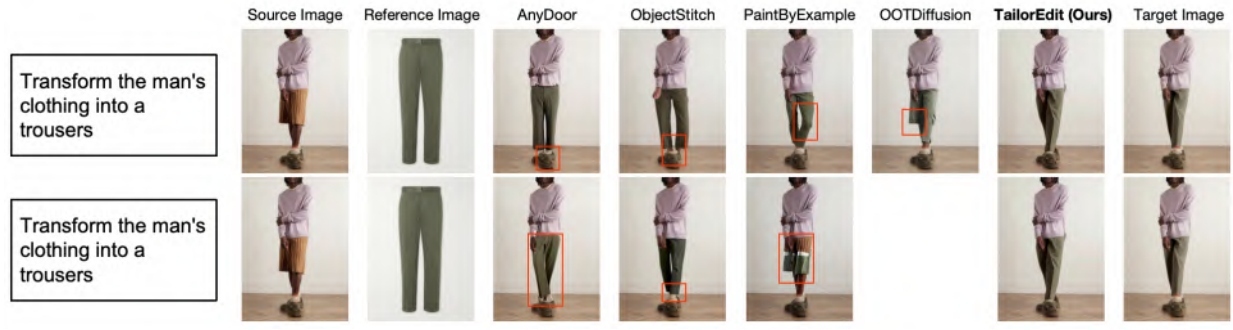
(a) Qualitative results of *Removal* task.



(b) Qualitative results of *Alteration* task.



(c) Qualitative results of *Addition* task.



(d) Qualitative results of *Replacement* task.

Fig. 4: Qualitative comparisons of the proposed method and baseline approaches across four editing tasks. For each task, a representative example is randomly selected, with baseline results shown before fine-tuning (top row) and after fine-tuning (bottom row) on the *InstructFashion* dataset.

TABLE II: Ablation studies of the effect of TailorEdit's components on four editing tasks. The best and suboptimal values are highlighted in bold and underlined, respectively.

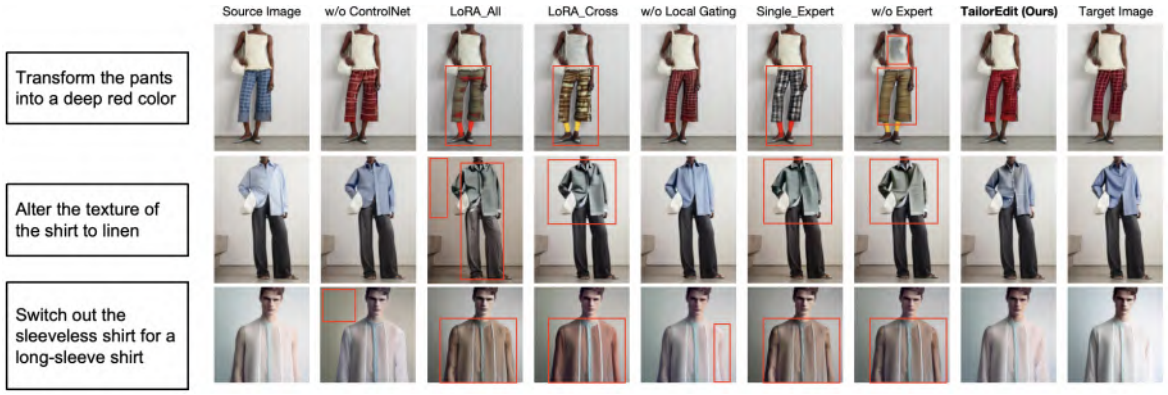
Editing Task	Method	PSNR ↑	FID ↓	LPIPS ↓	CLIP-I ↑	DINO ↑	CLIP-T ↑
Removal	w/o ControlNet	27.745	21.796	0.064	0.9350	0.991	0.2726
	LoRA_All	21.028	37.634	0.137	0.8687	0.964	0.2512
	LoRA_Cross	27.726	21.384	0.063	0.9335	0.991	0.2735
	w/o Local Gating	<u>28.900</u>	<u>20.046</u>	<u>0.058</u>	<u>0.9379</u>	**0.993**	**0.2744**
	Single_Expert	18.944	42.699	0.164	0.8453	0.960	0.2457
	w/o Expert	24.596	26.333	0.081	0.9182	0.988	0.2692
	TailorEdit (Ours)	**29.043**	**19.935**	**0.057**	**0.9389**	**0.993**	<u>0.2740</u>
Alteration	w/o ControlNet	17.204	23.311	0.175	0.9025	0.970	0.2728
	LoRA_All	13.507	42.440	0.262	0.8311	0.922	0.2400
	LoRA_Cross	16.303	26.521	0.181	0.9003	0.968	0.2690
	w/o Local Gating	<u>19.029</u>	<u>20.838</u>	<u>0.130</u>	<u>0.9141</u>	**0.978**	<u>0.2756</u>
	Single_Expert	15.737	29.867	0.181	0.8806	0.964	0.2580
	w/o Expert	16.059	26.510	0.168	0.8966	0.970	0.2656
	TailorEdit (Ours)	**19.908**	**19.101**	**0.124**	**0.9153**	<u>0.977</u>	**0.2781**
Addition	w/o ControlNet	<u>26.206</u>	<u>19.519</u>	<u>0.060</u>	<u>0.9373</u>	<u>0.993</u>	**0.2785**
	LoRA_All	20.655	32.829	0.127	0.8906	0.969	0.2648
	LoRA_Cross	24.754	22.889	0.073	0.9297	0.989	0.2764
	w/o Local Gating	26.201	20.721	<u>0.060</u>	0.9353	0.992	0.2778
	Single_Expert	19.210	38.043	0.148	0.8634	0.961	0.2577
	w/o Expert	21.257	28.857	0.104	0.9123	0.981	0.2715
	TailorEdit (Ours)	**26.993**	**18.314**	**0.055**	**0.9407**	**0.994**	<u>0.2784</u>
Replacement	w/o ControlNet	19.285	22.428	0.101	<u>0.9058</u>	0.981	<u>0.2712</u>
	LoRA_All	15.466	29.340	0.143	0.8697	0.965	0.2570
	LoRA_Cross	17.863	24.884	0.124	0.8955	0.972	0.2667
	w/o Local Gating	<u>20.398</u>	<u>21.292</u>	<u>0.091</u>	0.9049	<u>0.984</u>	0.2677
	Single_Expert	19.225	23.790	0.095	0.9016	0.981	0.2681
	w/o Expert	17.258	28.826	0.122	0.8905	0.973	0.2638
	TailorEdit (Ours)	**20.539**	**19.123**	**0.085**	**0.9153**	**0.986**	**0.2733**

(Y)" indicates a drop of Y (marked in red). For metrics where *lower is better* (e.g., FID), "X (Y)" denotes a decrease of Y (i.e., improved performance, marked in blue), and "X (+Y)" indicates an increase of Y (i.e., worse performance, marked in red). A value shown as "X (–)" indicates no change after fine-tuning.

The results of ZONE [23] are reported without fine-tuning, as it is inherently a training-free method and does not require further optimization on new datasets. Similarly, we evaluate OOTDiffusion [37] without fine-tuning, given that it is a virtual try-on model specifically designed for garment replacement and synthesis. Since the samples in our *Addition* and *Replacement* tasks are generated using a different virtual try-on algorithm, we argue that fine-tuning OOTDiffusion on *InstructFashion* is unnecessary and may not yield meaningful improvements under this setting. From the results, we observe these findings: (1) For the *Removal* task, our method consistently outperforms all baseline approaches across all evaluation metrics. Moreover, we observe that fine-tuning significantly boosts the performance of most baseline models, reinforcing the importance of adapting to task-specific data distributions. (2) For the *Alteration* task, our method achieves the highest CLIP-T score, indicating strong text-image alignment. It also delivers competitive results in FID and CLIP-I scores. Although InstructDiffusion [25] performs well on this task, it shows a substantial drop in performance on the *Removal* task, highlighting its limited generalization across diverse editing scenarios. (3) For the *Addition* and *Replacement* tasks, our method outperforms the baselines in terms of PSNR, FID, and DINO scores, demonstrating its strength in producing visually faithful and semantically consistent edits. While it may not achieve the highest score on every metric, our method remains highly competitive and consistently ranks among the top-performing models. OOTDiffusion [37] performs well on the *Addition* and *Replacement* tasks, which is consistent with our expectations. This is because OOTDiffusion is specifically designed for virtual try-on and garment transfer, making it well-suited for tasks that involve adding or replacing clothing items. Furthermore, we observe that PaintByExample [34] benefits only marginally from fine-tuning, suggesting its limited capacity for high-quality generation in this context. In summary, these results affirm the strong performance, robustness, and adaptability of our proposed method across multiple fashion editing tasks.

2) Qualitative Analysis: Fig. 4 presents a qualitative comparison between our method and baseline approaches across four editing tasks. For the *Removal* task, our method effectively removes accessories such as belts. ZONE [23] struggles to remove the belt, and other baseline methods also perform poorly before fine-tuning. Although their results improve after fine-tuning, the edits still lack the visual coherence and realism achieved by our approach. For the *Alteration* task, our method accurately modifies garment colors while preserving the original structure and contextual consistency. In contrast, while baseline methods can alter colors, their results often appear unnatural and unrealistic, and frequently introduce unintended

(a) Qualitative results of *Removal* task.



(b) Qualitative results of *Alteration* task.

(c) Qualitative results of *Addition* task.

(d) Qualitative results of *Replacement* task.

Fig. 5: Qualitative comparisons of the ablated variants and the proposed method across four editing tasks.

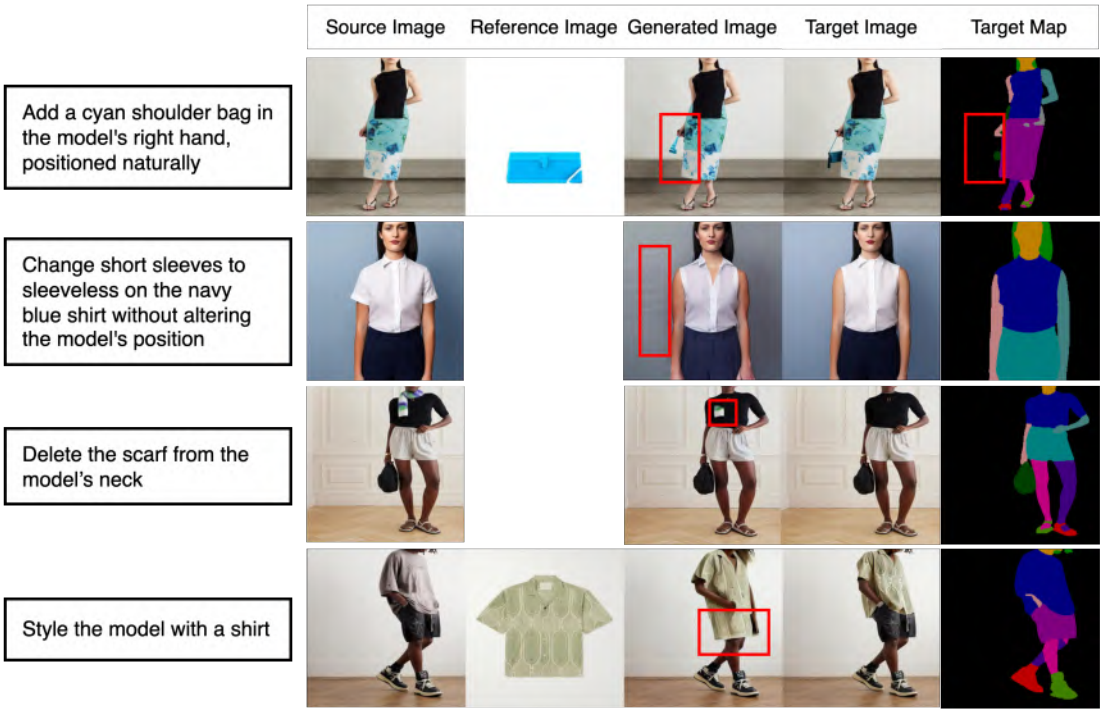


Fig. 6: Several representative failure cases of our method.

changes in irrelevant regions such as the background or facial area. For the *Addition* task, our method seamlessly integrates accessories like sunglasses in a natural way. In comparison, although baseline methods are capable of adding sunglasses, they often distort their appearance, leading to less realistic results. AnyDoor [35] is able to address this issue after fine-tuning, further demonstrating that fine-tuning on our dataset significantly enhances the performance of baseline models. For the *Replacement* task, our method successfully replaces garments while preserving the overall body structure. On the other hand, all baselines struggle to maintain body proportions and spatial coherence. These results highlight the effectiveness of our approach, which consistently surpasses baseline methods across different editing tasks.

3) Ablation Study: We conducted several ablation experiments to evaluate the effectiveness of key components in our proposed framework: (1) w/o ControlNet: This variant excludes the ControlNet component entirely from the architecture. During both the task-specific expert learning and multi-task integration stages, the model operates without the guidance of human-body structural information provided by ControlNet. (2) LoRA_All: a model variant that applies LoRA to all linear layers in the U-Net. (3) LoRA_Cross: a model variant that applies LoRA only to cross-attention layers in the U-Net. (4) w/o Local Gating: a model variant that removes the local gating module. (5) Single_Expert: This variant trains task-specific experts independently, without proceeding to the multi-task integration stage. Each expert is optimized solely on its corresponding editing task, with no further joint training or gating refinement applied. (6) w/o Expert: This variant removes the task-specific expert modules. Instead, it combines the pre-trained ControlNet with the IP2P model [15] and fine-tunes the entire architecture directly on the multi-task dataset.

Table II presents the quantitative evaluation results. From the results, we observe the following findings: (1) Our method consistently outperforms all ablated variants across all editing tasks. (2) Removing the ControlNet leads to a noticeable drop in performance, highlighting the critical role of structural guidance in generating accurate and coherent edits. In the *Addition* task, human semantic information provides less explicit guidance compared to the other tasks. This is because, in the *Addition* task, the predicted human segmentation map often does not align well with the ground truth. For example, when adding a bag to a person, the person might hold the bag from the front or the side, leading to variations in the spatial alignment. (3) While applying LoRA to all linear layers provides greater flexibility, it does not necessarily lead to better results. (4) Incorporating LoRA into both self-attention and cross-attention layers of the U-Net is essential for achieving optimal performance. (5) Introducing the local gating module enhances editing precision. (6) The multi-task integration stage, combined with gating refinement, proves essential for achieving optimal performance. Especially, *Removal* and *Addition* tasks perform worse when trained alone. In contrast, after joint optimization on our multi-task dataset, performance significantly improves, demonstrating the effectiveness of multi-task optimization. (7) The task-specific expert modules significantly boost the model's ability to handle diverse editing instructions. In summary, these findings highlight the effectiveness of each component within our framework.

The qualitative evaluation results are shown in Fig. 5, offering visual comparisons among the different model variants. Our full method consistently generates more realistic and

semantically coherent editing results compared to all ablated baselines. Notably, the w/o Local Gating variant produces outputs that are closest to our complete model in overall quality. However, it still underperforms in capturing fine-grained visual details, particularly in regions with intricate textures or subtle semantic changes. These results highlight the contribution and necessity of each individual component within our proposed framework.

4) Failure Cases: Fig. 6 presents several representative failure cases of our method, which can be categorized as follows: (1) In the *Addition* task, when adding accessories such as bags, the generated results may be incomplete due to inaccuracies in the predicted human segmentation maps. (2) In the *Alteration* task, when modifying fashion attributes such as shape, our method may unintentionally alter background regions, leading to a loss of contextual consistency. (3) In the *Removal* task, when removing accessories such as scarves, our method occasionally fails to completely eliminate the target item, resulting in residual artifacts. (4) In the *Replacement* task, our method may incorrectly modify unintended regions, such as surrounding garments or accessories.

VI. Conclusion and Future Work

We introduce *TailorEdit*, a novel adaptive framework for instruction-guided fashion image editing. TailorEdit incorporates human segmentation map-based denoising guidance, low-rank task-specific experts, and a dynamic expert routing mechanism to enable precise and semantically coherent edits. To support this research, we construct *InstructFashion*, a large-scale, high-quality dataset tailored for instruction-guided fashion image editing. This dataset was generated through carefully designed pipelines encompassing four distinct editing tasks. Extensive quantitative and qualitative evaluations demonstrate the effectiveness of TailorEdit, consistently outperforming state-of-the-art methods in terms of realism, semantic coherence, and adherence to editing instructions.

While TailorEdit achieves strong performance in instruction-guided fashion image editing, several promising directions remain for future work. First, we plan to extend our framework to handle more diverse and complex scenarios, such as multi-instruction editing, which introduces new challenges in maintaining visual consistency and faithfully following all instructions. Second, we intend to expand InstructFashion by incorporating a wider variety of garment types and body poses to further improve the model's robustness and generalization. Finally, we aim to enhance the scalability of our low-rank expert modules by developing more efficient parameter-sharing strategies, facilitating broader deployment in real-world applications.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62471168, 62422204 and 61802100, and in part by the Zhejiang Provincial Natural Science Foundation of China under Grants LDT23F02025F02, LRG26F020001, LD24F020015, and LMS26F020015, in part by the Key Research and Development Program of Zhejiang Province No. 2025C01026 and in part by the Scientific Research Innovation Capability Support Project for Young Faculty.

References

[1] D. Song, J.-H. Zeng, M. Liu, X.-Y. Li, and A.-A. Liu, "Fashion customization: Image generation based on editing clue," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 6, pp. 4434–4444, 2024.

[2] D. Zhou, H. Zhang, J. Ma, and J. Shi, "Bc-gan: A generative adversarial network for synthesizing a batch of collocated clothing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 5, pp. 3245–3259, 2024.

[3] H. Yang, R. Zhang, X. Guo, W. Liu, W. Zuo, and P. Luo, "Towards photo-realistic virtual try-on by adaptively generating-preserving image content," in *CVPR*, 2020, pp. 7850–7859.

[4] Y. Ge, Y. Song, R. Zhang, C. Ge, W. Liu, and P. Luo, "Parser-free virtual try-on via distilling appearance flows," in *CVPR*, 2021, pp. 8485–8493.

[5] L. Zhu, D. Yang, T. Zhu, F. Reda, W. Chan, C. Saharia, M. Norouzi, and I. Kemelmacher-Shlizerman, "Tryondiffusion: A tale of two unets," in *CVPR*, 2023, pp. 4606–4615.

[6] D. Morelli, A. Baldrati, G. Cartella, M. Cornia, M. Bertini, and R. Cucchiara, "Ladi-vton: Latent diffusion textual-inversion enhanced virtual try-on," in *ACM Multimedia*, 2023, pp. 8580–8589.

[7] J. Kim, G. Gu, M. Park, S. Park, and J. Choo, "Stableviton: Learning semantic correspondence with latent diffusion model for virtual try-on," in *CVPR*, 2024, pp. 8176–8185.

[8] M. Günel, E. Erdem, and A. Erdem, "Language guided fashion image manipulation with feature-wise transformations," *arXiv preprint arXiv:1808.04000*, 2018.

[9] T. Wang and M. Ye, "Texfit: Text-driven fashion image editing with diffusion models," in *AAAI*, vol. 38, no. 9, 2024, pp. 10 198–10 206.

[10] H. Dong, X. Liang, Y. Zhang, X. Zhang, X. Shen, Z. Xie, B. Wu, and J. Yin, "Fashion editing with adversarial parsing learning," in *CVPR*, 2020, pp. 8120–8128.

[11] A. Baldrati, D. Morelli, G. Cartella, M. Cornia, M. Bertini, and R. Cucchiara, "Multimodal garment designer: Human-centric latent diffusion models for fashion image editing," in *ICCV*, 2023, pp. 23 393–23 402.

[12] N. Rostamzadeh, S. Hosseini, T. Boquet, W. Stokowiec, Y. Zhang, C. Jauvin, and C. Pal, "Fashion-gen: The generative fashion dataset and challenge," *CoRR*, vol. abs/1806.08317, 2018.

[13] Y. Jiang, S. Yang, H. Qiu, W. Wu, C. C. Loy, and Z. Liu, "Text2Human: text-driven controllable human image generation," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 162:1–162:11, 2022.

[14] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *ICLR*, 2022.

[15] T. Brooks, A. Holynski, and A. A. Efros, "InstructPix2Pix: Learning to follow image editing instructions," in *CVPR*, 2023, pp. 18 392–18 402.

[16] Z. Yang, T. Chu, X. Lin, E. Gao, D. Liu, J. Yang, and C. Wang, "Eliminating contextual prior bias for semantic image editing via dual-cycle diffusion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 2, pp. 1316–1320, 2024.

[17] L. Yang, Y. Zhao, Z. Yu, B. Zeng, M. Xu, S. Hong, and B. Cui, "Spatio-temporal energy-guided diffusion model for zero-shot video synthesis and editing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 6, pp. 6034–6046, 2025.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[19] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NeurIPS*, vol. 33, pp. 6840–6851, 2020.

[20] Y. Huang, L. Xie, X. Wang, Z. Yuan, X. Cun, Y. Ge, J. Zhou, C. Dong, R. Huang, R. Zhang *et al.*, "Smartedit: Exploring complex instruction-based image editing with multimodal large language models," in *CVPR*, 2024, pp. 8362–8371.

[21] T. Fu, W. Hu, X. Du, W. Y. Wang, Y. Yang, and Z. Gan, "Guiding instruction-based image editing via multimodal large language models," in *ICLR*, 2024.

[22] Y. Li, Y. Bian, X. Ju, Z. Zhang, Y. Shan, and Q. Xu, "Brushedit: All-in-one image inpainting and editing," *arXiv preprint arXiv:2412.10316*, 2024.

[23] S. Li, B. Zeng, Y. Feng, S. Gao, X. Liu, J. Liu, L. Li, X. Tang, Y. Hu, J. Liu *et al.*, "Zone: Zero-shot instruction-guided local editing," in *CVPR*, 2024, pp. 6254–6263.

[24] Q. Guo and T. Lin, "Focus on your instruction: Fine-grained and multi-instruction image editing by attention modulation," in *CVPR*, 2024, pp. 6986–6996.

[25] Z. Geng, B. Yang, T. Hang, C. Li, S. Gu, T. Zhang, J. Bao, Z. Zhang, H. Li, H. Hu *et al.*, "Instructdiffusion: A generalist modeling interface for vision tasks," in *CVPR*, 2024, pp. 12 709–12 720.

[26] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W. Lo, P. Dollár, and R. B. Girshick, "Segment anything," in *ICCV*, 2023, pp. 3992–4003.

[27] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022, pp. 10 674–10 685.

[28] Y. Choi, S. Kwak, K. Lee, H. Choi, and J. Shin, "Improving diffusion models for authentic virtual try-on in the wild," in *ECCV*, ser. Lecture Notes in Computer Science, vol. 15144, 2024, pp. 206–235.

[29] P. Li, Y. Xu, Y. Wei, and Y. Yang, "Self-correction for human parsing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3260–3271, 2022.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763.

[31] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang, "Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models," *CoRR*, vol. abs/2308.06721, 2023.

[32] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *ICLR*, 2021.

[33] K. Zhang, L. Mo, W. Chen, H. Sun, and Y. Su, "MagicBrush: A manually annotated dataset for instruction-guided image editing," *Advances in Neural Information Processing Systems*, vol. 36, pp. 31 428–31 449, 2023.

[34] B. Yang, S. Gu, B. Zhang, T. Zhang, X. Chen, X. Sun, D. Chen, and F. Wen, "Paint by example: Exemplar-based image editing with diffusion models," in *CVPR*, 2023, pp. 18 381–18 391.

[35] X. Chen, L. Huang, Y. Liu, Y. Shen, D. Zhao, and H. Zhao, "Anydoor: Zero-shot object-level image customization," in *CVPR*, 2024, pp. 6593–6602.

[36] Y. Song, Z. Zhang, Z. Lin, S. Cohen, B. Price, J. Zhang, S. Y. Kim, and D. Aliaga, "Objectstitch: Object compositing with diffusion model," in *CVPR*, 2023, pp. 18 310–18 319.

[37] Y. Xu, T. Gu, W. Chen, and C. Chen, "OOTDiffusion: Outfitting fusion based latent diffusion for controllable virtual try-on," *arXiv preprint arXiv:2403.01779*, 2024.

[38] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *ICPR*, 2010, pp. 2366–2369.

[39] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *NeurIPS*, vol. 30, 2017.

[40] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018, pp. 586–595.

[41] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi, "Clipscore: A reference-free evaluation metric for image captioning," *arXiv preprint arXiv:2104.08718*, 2021.

[42] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *ICCV*, 2021, pp. 9650–9660.

Xiaoling Gu received the PhD degree in computer science from Zhejiang University in 2017. She is currently an Associate Professor at the School of Computer Science and Technology, Hangzhou Dianzi University, China. Her current research interests are in the areas of multimedia analysis, visual-language learning and image generative models.

Lingda Zhu is currently a master candidate at Hangzhou Dianzi University. Her research interests include computer vision and deep learning.

Yongkang Wong is a senior research fellow at the School of Computing, National University of Singapore. He is also the Assistant Director of the NUS Centre for Research in Privacy Technologies (N-CRiPT). He obtained his BEng from the University of Adelaide and PhD from the University of Queensland. He has worked as a graduate researcher at NICTA's Queensland laboratory, Brisbane, QLD, Australia, from 2008 to 2012. His current research interests are in Image/Video Processing, Machine Learning, and Social Scene Analysis.

Zhou Yu received the B.Eng. and Ph.D. degrees from Zhejiang University, Zhejiang, China, in 2010 and 2015, respectively. He is currently a Full Professor with the School of Computer Science and Technology, Hangzhou Dianzi University, China.His research interests include multimodal analysis, visual-language learning, and large multimodal models. He has authored and co-authored more than 40 scientific articles and has also served as an invited reviewers or a program committee members for top conferences and prestigious journals including AAAI, IJCAI, IEEE Trans. on PAMI, IEEE Trans. on MM, IEEE Trans. On CSVT, IEEE Trans. on NNLS.

Huan Li is a ZJU-100 Professor at Zhejiang University and a recipient of an EU Marie Curie Individual Fellowship. He was an Assistant Professor at Aalborg University in Denmark from 2020 to 2023 and a Senior Engineer at Alibaba from 2018 to 2019. He received his PhD from Zhejiang University in 2018. His research focuses on data-centric AI, efficient AI, and spatiotemporal data management. He is a member of IEEE and ACM.

Zizhao Wu is currently an Associate Professor with the Faculty of Digital Media Technology, Hangzhou Dianzi University. He received the Ph.D. degree from the Department of Computer Science and Technology, Zhejiang University, in 2013. His main research interests include computer vision and computer graphics.

Mohan S. Kankanhalli is the Provost's Chair Professor at the Department of Computer Science of the National University of Singapore. He is the director with the N-CRiPT and also the Dean, School of Computing at NUS. Mohan obtained his BTech from IIT Kharagpur and MS & PhD from the Rensselaer Polytechnic Institute. His current research interests are in Multimedia Computing, Multimedia Security, Image/Video Processing and Social Media Analysis. He is active in the Multimedia Research Community and is on the editorial boards of several journals.