Domain-Specialized Text-Guided Image Editing: Improving Edit Accuracy via Targeted Diffusion Fine-Tuning (Fashion Domain)

Sona Krishnan
sk3449@cornell.edu
sk3449

David Thai
dt573@cornell.edu
dt573

Vishali Vallioor
vv266@cornell.edu
vv266

Abstract

Text-guided image editing allows users to change specific aspects of an image using natural language instructions. Although general diffusion-based editors can perform well for general images, they often struggle with specialized categories such as fashion, where details and localization matter.

In this project, we explore whether fine-tuning a model on domain-specific data can improve edit accuracy and visual quality for fashion images. Using the DeepFashion2 dataset, we automatically construct training triplets of the structure (source image, instruction, target image) by pairing images of the same person wearing different outfits and generating edit instructions using the caption differences. We then fine tune using Selective parameter efficient tuning. Instead of updating 860M+ UNet parameters, we freeze the full UNet and unfreeze only a targeted subset of layers responsible for text image conditioning and garment details specifically. This exposes 23.84% of the UNet, allowing the model to learn domain specific transformations without overfitting. We additionally implemented mask aware training using DeepFashion2 segmentation masks, ensuring edits remained localized to garments while preserving the background.

We evaluate the model on DeepFashion2 and unseen FashionPedia [Zhu et al., 2019] images, where the experiments show that this approach improves garment realism, edit localization, and texture fidelity compared to the pretrained baseline.

1 Introduction

Text-guided image editing allows users to modify images using natural language instructions by combining language understanding with generative modeling. Diffusion-based models such as InstructPix2Pix [Brooks et al., 2023] and SDEdit [Meng et al., 2021] perform well for many general editing tasks, but often struggle with precision and realism in more specialized domains. In the case of fashion images, the edits usually distort textures, incorrectly modify clothing boundaries, or apply changes too broadly across the image.

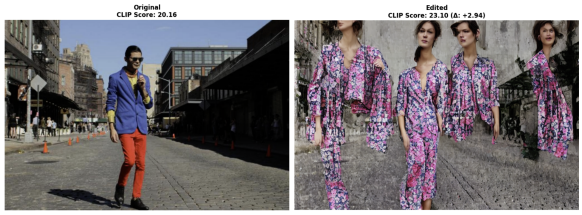

Figure 1: An example from our first attempt to edit clothing on a person. The prompt we had used for this specific example was "change the clothes this person is wearing to a floral pattern".

The motivation for our research stems from an emerging opportunity to reduce costs in fashion design iteration. In May of 2025, a European online retailer successfully integrated generative AI into their content production pipeline, reducing campaign production timelines from weeks to days, [Reuters, 2025]. This also cut their research cost by 90 percent, additionally allowing them to rapidly design for "micro-trends" in the industry. This experiment yielded impressive results, showing a potential to optimize large portions of the fashion design process.

The current fashion design cycle is very resource-intensive, and relies on physical samples to test textures and viewing angles. This workflow is not only significantly slowed due to a huge reliance on physical models but also contributes to the industry's environmental footprint. In addition to the consumer waste created when people throw out retail clothing, the companies that produce these items generate even more industrial waste. In a study by the Bren School of Environmental Science at UC Santa Barbara, it was discovered that "for every pound of clothing fabric that we throw away as a consumer, a business throws away 40 pounds" [Jones et al., 2024]. This report revealed the shocking scale with which businesses and designers spend resources on refining their product, creating the need to find alternatives to simplify that process. We believe that our work addresses this need to digitize parts of the fashion industry and reduce waste from both ends of the fashion distribution pipeline. We foresee our work being applicable to clothing designers for rapid iteration and consumers for digital fitting on self-uploaded media.

To directly provide extra information of issues in fashion image editing, in Figure 1, we can see that there are failure modes that commonly arise when applying diffusion editors meant for general purpose to fashion images. The baseline InstructPix2Pix model [Brooks et al., 2023] here frequently seems to struggle with preserving fine-grained garment structure, which causes textures to stretch or even warp when the edits are applied. Secondly, color and pattern edits seem to bleed outside of the intended real clothing region, which affects skin, hair, and background elements. Third, repeated textures tend to appear overall misaligned with garment folds and seams, since the model lacks priors about fashion information and details. Overall, addressing these issues in the broader region of fashion editing makes us primarily interested in this domain specialization, to fine-tune on fashion-centric case scenarios.

These limitations motivate our focus on domain specialization for text-guided editing. Our goal was to investigate whether fine-tuning a diffusion model on fashion-specific data can help it learn relevant visual structures and improve both edit accuracy and visual quality. The fashion domain is particularly challenging because it not only requires semantic correctness but also the preservation of shape and texture details. In this project, we are building on the

InstructPix2Pix framework and fine tuning the model using selective parameter efficient tuning (PEFT) on the DeepFashion2 dataset. PEFT provides a way to adapt large diffusion models without retraining the full network, making it suitable for domain specific specialization. It freezes the full UNet and unfreeze only a targeted subset of layers responsible for text image conditioning, training only 23.84 % of parameters. We also use mask aware denoising to localize edits, and leverages DeepFashion2 triplets to learn clothing specific transformations. Our results show that Selective PEFT enables strong specialization with minimal compute and preserves the general fashion editing capabilities of InstructPix2Pix. We also compare the domain specialized and baseline models using semantic and similarity evaluation metrics.

2 Related Work

We examined other works to guide us in approaching our goal to fine-tune InstructPix2Pix [Brooks et al., 2023] for the fashion domain. To begin, we analyzed the existing technologies like InstructPix2Pix, SDEdit, DiffEdit, and SwiftEdit2. InstructPix2Pix introduced supervised training for instruction following image to image diffusion. The other approaches attempt to localize edits but they still rely on general purpose priors, which limits their performance in specialzed domains.

Building on this structure, we introduced "LoRA: Low-Rank Adaptation of Large Language Models" [Hu et al., 2021] to fine-tune the general purpose models. We will primarily be following the methodology from DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. This paper introduces the "personalization" of text-to-image diffusion models. This paper's approach specifically aimed to improve "novel articulations and variation in lighting conditions", while still maintaining high visual quality. Although LoRA is a popular choice for parameter efficient tuning, we decided not to use it in our final model. In our early experiments, LoRA adapters were to restrictive for the kinds of high accuracy texture changes that are required in fashion specific editing. Because LoRA limits updates to low-rank transformations, it struggled to capture the structure of fabrics, repeated patterns, and subtle shading variations. In contrast, selectively unfreezing specific attention and convolutional layers gave the model significantly more expressive power while still avoiding the cost of full end-to-end fine-tuning. This selective strategy provided a better balance between efficiency and flexibility, enabling the model to learn fashion-specific patterns and localized garment edits that LoRA was unable to capture during initial tests.

In addition, we explored different diffusion models like DDPM, DDIM, and Stable Diffusion; introducing noise based image generation. InstructPix2Pix extends Stable Diffusion with paired before and after edit examples and editing instructions.

We will also use CLIP-based metrics [Wolf et al., 2020] for evaluating the quality of our model's improvement as they are common for evaluating text-image alignment and edit quality. Directional Similarity [Goel et al., 2022] is also well suited for evaluating semantic transformations.

3 Problem Definition

Let I represent the original fashion image and t be a natural language edit instruction. Our goal is to produce an edited image $I' = f_\theta(I, t)$ using a diffusion model with parameters θ, such that:

(1) The edited image I' accurately reflects the meaning of the text prompt t;
(2) The visual content unrelated to the instruction is preserved from the original image I.

InstructPix2Pix [Brooks et al., 2023] formulates this as a conditional denoising problem: the U-Net is trained to predict the noise residual for the edited image while attending to the original image and a CLIP embedding of the edit instruction. However, in specialized domains like fashion, the pretrained model often struggles with texture, garment localization, and color consistency.

To adapt the model to fashion-specific editing, we fine-tune a selected subset of U-Net parameters using triplets from DeepFashion2: (I,t,I*) where I* is the target image representing the desired edit. The fine-tuning objective encourages the model to learn fashion focused transformations such as recoloring clothing, adding patterns, or modifying textures.

Our evaluation focuses on three metrics commonly used in text-guided image editing: CLIP Similarity, CLIP Directional Similarity and LPIPS. These metrics allow us to evaluate both semantic correctness, directionality, and perceptual quality (realism). Our goal is therefore to learn a model that simultaneously improves edit accuracy and garment localization while maintaining high perceptual quality.

4 Data

To align the InstructPix2Pix [Brooks et al., 2023] architecture with the nuances of fashion design (poses, texture, lighting, skin-clothing boundaries, etc.), our fine-tuning process requires a data set that is structured as a supervised triplet: a conditioned source image (c_I), a natural language editing instruction (T), and a ground-truth target image (c_O). In our exploration of available datasets that fit this space, we failed to find a large, accessible dataset that fits our specific niche in the fashion domain. We initially investigated filtering the MagicBrush dataset [Zhang et al., 2023], a widely used dataset for text-to-image editing diffusion models, for clothing-related items, but found that it had minimal fashion images. Our initial approach was to define a dictionary of clothing related keywords (e.g. shirt, tank-top, pants, etc.) and filter for captions containing these words. However, this filtration yielded a critically low sample ($< 2000 samples$), which would pose a risk of overfitting and failing to capture the diverse structural and textural attributes of clothing needed to generate realistic images. To overcome this dataset issue, we decided to generate our own version using existing fashion-related datasets that had captions.

The first dataset we used was **DeepFashion2** [Ge et al., 2019], which was accessed directly via the Hugging Face API [SaffalPoosh, 2025]. DeepFashion2 contains around 291K fashion images with detailed labels for categories, poses, and attributes, which we formatted to fine-tune our model on. Using this dataset, we generate triplets by extracting same person pairs of multiple outfits, captions describing outfits, and segmentation masks.

In contrast, while the **FashionPedia** dataset [Zhu et al., 2019] is a good benchmark for evaluating fashion models, it is not ideal for training InstructPix2Pix because it does not provide paired edited images or edit instructions. Therefore, we use FashionPedia only for our baseline evaluation of the pretrained model, and we use DeepFashion2 for training and further fine-tuning. For fine-tuning, the DeepFashion2 data is split into 80% training, 10% validation, and 10% testing, allowing the model to learn mappings between the original image, the instruction, and the target edited output. We use FashionPedia to evaluate our fine tuned model on unseen data because it offers a diverse style and clothing categories.

5 Methodology

5.1 Baseline: InstructPix2Pix

As our baseline model, we chose to use InstructPix2Pix [Brooks et al., 2023], which is a supervised image-to-image diffusion model that conditions directly on both an input image and a natural language edit instruction. The way the model framework operates is that the input image is noised, and the U-Net denoiser predicts the corresponding noise residual while attending to a CLIP embedding of the instruction. This allows the model to modify the image in a direction aligned with the prompt. Because the model is trained on a broad set of general edits, it provides a reasonable starting point but struggles with the fine-grained understanding of clothing structure and fashion-specific textures needed for high-quality clothing edits.

5.2 Triplet Generation From DeepFashion2

A key innovation of this project is creating a synthetic dataset of paired editing examples. Since we couldn't find a suitable, accessible dataset given the project time-frame, our best option was to generate our own version of the necessary data (triplets of conditioned image, instruction, output image). Our approach to generating this triplet format of data was to use existing the fashion-related image datasets, FashionPedia and DeepFashion2. We first grouped images by person ID. The DeepFashion2 images include a pid field, so we grouped samples by pid and identify individuals with 2+ images. We then constructed paired examples, where for each pair of images of the same person, one becomes the source image and the other becomes the target image. Next, we formed a text edit instruction to automatically generate instructions. We generated 2000 triples with the structure of (conditioning image, prompt, target image).

Figure 2: Example of how Triplet Generation was done by using the same person from dataset wearing 2 different outfits where one becomes the source image and the other becomes the target.

5.3 Selective Parameter Efficient Fine-Tuning

To adapt the baseline InstructPix2Pix model to the fashion domain, we used selective parameter-efficient fine-tuning that targets the layers most responsible for text–image alignment and garment details. Instead of updating all 860M parameters of the U-Net, we freeze the majority of the network and unfreeze only a chosen subset of layers whose functions directly influence semantic editing, texture, and visual detail.

Cross-Attention Layers. The cross-attention layers determine how the model uses the text prompt while denoising the image. By unfreezing the `to_k`, `to_v`, and `to_out` projections in `attn2`, we allow the model to relearn how specific fashion-related instructions should influence the edited image. These layers play a major role in connecting the meaning of the prompt to the visual changes the model generates, so updating them helps improve how accurately the model follows fashion-specific text instructions.

Self-Attention Layers. Self-attention handles how different pixels or regions of an image relate to each other. By unfreezing the `to_q`, `to_k`, and `to_v` layers inside `attn1`, we allow the model to better understand the shapes and textures of clothing. This helps the model maintain garment boundaries, follow fabric folds, and avoid distortions when applying edits such as patterns or color changes. Updating these layers makes the edits look more natural and consistent with the original garment.

Mid Blocks. The mid-block of the U-Net is where the features are combined. Unfreezing this block helps the model learn broader fashion concepts, such as overall outfit style or fabric type. This is useful for prompts that require more than just color changing patterns or changing the clothing style because these edits rely on higher level visual understanding.

Up Sampling Block. We also unfreeze the convolutional layers in the up sampling blocks (`up_blocks.2` and `up_blocks.3`), which are responsible for generating the final detailed output. These layers control the small details in fabric textures and fine patterns. Allowing these layers to update is important for fashion editing, where the accuracy and realness often depends on very subtle visual details.

Altogether, this selective unfreezing gives us about 204M trainable parameters, which is 23.84% of the full U-Net. This is far more efficient than fine-tuning all 860M parameters but still gives the model enough flexibility to learn fashion-specific transformations. In practice, this approach gave us a good balance of lower training cost and memory use, while still achieving strong improvements in edit accuracy, texture quality, and localization of garment changes.

6 Experiments

After we completed training for an initial fine-tuned InstructPix2Pix model [Brooks et al., 2023], we wanted to understand how much further we could push its performance. We noticed that despite improvement from the evaluation metrics, we still lacked high quality visual output that would make an image look realistic. We frequently observed the model's edits leaking past the boundaries of the subject and into the background.

We explored two different experiments to further understand how to expand on our model's performance. The first was to to redesign our loss function to put further emphasis on restricting the model's editing space. In the second experiment, we expanded by performing an Ablation test to better understand the systems behind hyperparameter tuning and which settings balance bias and variance.

6.1 Improving Clothes Masking

A consistent issue we noticed visually was that the model would consistently edit areas outside the region specified in the text prompt (A clear example would be **Figure 1** in our introduction, showing the base InstructPix2Pix model will edit far beyond the subject's boundaries in an image).

One particular subset of examples motivated our interest to explore the "bounding boxes" within the FashionPedia dataset. The example that caught our attention was from our initial evaluations of our fine-tuned model on the DeepFashion2 dataset, which contains "black bars" that surround the images width-wise. In regards to our color-specific image editing instructions, we noticed that the bars on the side would sometimes be affected by the targeted color.

Figure 3: Example of color-related editing affecting the sides of the image that were not intended to be part of the background.

Luckily for us, FashionPedia additionally has "bounding boxes" associated with the image data. These are a set of coordinates corresponding to each image that details where an item of clothing exists. The coordinates form a rectangular "box" that confines where the model should recognize clothing. For example, in figure 4, the "23" wrapped around the subject's feet refer to "high-heel shoes". It was clear that the fine-tuned model we had at this point was failing to properly confine its editing within just the subject's clothing, so our hypothesis was that we could use these bounding boxes to produce a tighter bound on the editing region or mask.

We also recognized a limitation with this approach as this point, which was that in the case where an article of clothing's shape were to be edited (e.g. "make this dress longer"), the bounding box would prevent the model from actually altering anything outside of it. Additionally, changes to texture (e.g. "make this jacket puffy") would also alter past the defined tight bounds, so we decided to take a softer approach to bounding edits.

We chose to take a **hierarchal** approach to incorporating the bounding box loss. This approach is inspired by a research paper focused on taking a Bayesian Approach to Digital Matting [Chuang et al., 2001], where the researchers introduce the idea of "Trimap" -

Figure 4: Example of bounding boxes from the FashionPedia dataset. The annotated numbers around the box symbolize what type of clothing it encapsulates.

splitting an image into foreground, background, and a transitory region. This essentially gives the model some more "wiggle room" in where it's allowed to edit the image. Instead of a binary "yes/no" bound for each box, we created the following three "zones":

(1) **Tight Zone (Z_{tight}):** The precise segmentation mask where the primary edit occurs.
(2) **Context Zone ($Z_{context}$):** A transition region defined by the bounding box of the garment, expanded by a factor of 1.15, excluding the tight mask. This zone allows for minor structural adjustments (e.g., ruffles extending slightly beyond the original silhouette).
(3) **Background Zone (Z_{bg}):** The region outside the expanded bounding box, where strict preservation is enforced.

Figure 5: Example of calculating the "zones" the model is allowed to edit.

Implementation Details. The model was fine-tuned through a training loop not too different from the original. We used a max of

4 epochs with the AdamW optimizer. To ensure stable convergence, we used a dynamic learning rate scheduler. We also used the same early-stopping mechanism based on CLIP directional similarity on a held-out validation set, which would stop the loop if the metric failed to improve for 2 consecutive epochs (patience = 2).

6.1.1 Original vs. Modified Loss Function. Originally, the InstructPix2Pix model's loss function contains a term used to maintain the structure of the original image: $\mathcal{L}_{\text{preserve}}$. We found that tweaking this parameter to be higher didn't affect the model's ability to edit within the boundaries of our subject. This was another source of motivation to alter our loss function.

To address this, we wanted to tell the model to focus on denoising the part of the image we actually want to edit (clothes on a subject), while preserving the background. Instead of using extra perceptual or text-alignment losses, we opted to use constraints directly associated with the zones we set up. The idea is to encourage editing within the tight zone, allow for potential editing in the context zone, and heavily punish any editing within the background.

Our fine-tuning objective is defined as:

$$\mathcal{L}_{\text{total}} = \underbrace{||\epsilon - \epsilon_\theta||^2 \odot M}_{\text{Tight Edit}} + \lambda_{\text{ctx}} \underbrace{||\epsilon - \epsilon_\theta||^2 \odot (B \odot (1 - M))}_{\text{Context Transition}} + \lambda_{\text{bg}} \underbrace{||\epsilon - \epsilon_\theta||^2 \odot (1 - B)}_{\text{Background Freeze}} \tag{1}$$

In this new cost function, the symbol M denotes the tight segmentation mask that isolates only pixels belonging to the clothing region we want to edit. B represents the expanded region we're going to allow the model to edit into due to variations in clothing shape. Inside the context term, we multiply B and M's complement to represent the area only within the context zone and not an overlap of the two. Each term is also multiplied by the mean squared error between the true noise and the model's predicted noise. And finally, each term (besides the tight editing zone) has a lambda term (with subscript ctx for context and bg for background) to allow us to control how intensely to weigh the terms.

Experiment Results. The additional computation in our loss function scaled the training time by exactly double our original (2 hours -> 4 hours). This was probably due to the added complexity of needing to calculate the 3 separate bounding boxes along with its additional impact on the loss at each iteration. We chose to use the same metrics, CLIP score and Directional Similarity, as our original baseline to gauge how much improvement there was in our new edits.

DeepFashion2 Evaluation: We evaluated the fine-tuned model on 5 test images and 4 editing prompts (the same as how we evaluated the baseline model). The results are show in a table below this paragraph as well as the comparison to our original baseline model.

We found that **directional similarity was much higher** for this bounding-box aware model, but it also performed **slightly worse than the baseline on the CLIP score** (table 1). This result aligned with our original hypothesis that adding structural constraints to where the model can edit will preserve the original image's background better. Additionally, we found that the model was visually more consistent with editing in the desired regions.

Table 1: CLIP Similarity and Directional Similarity for the Bounding-Box Fine-Tuned Model.

Prompt	CLIP Score (↑)	Dir. Similarity (↑)
add stripes to the clothes	23.49	9.18
change shirt to blue	21.11	7.88
change to a floral pattern	21.25	13.67
make the shirt red	23.72	16.92
Mean	**22.14**	**11.91**

Table 2: Comparison of Mean CLIP Score and Directional Similarity between the Baseline and Bounding Box Fine-Tuned Model.

Method	CLIP Score (↑)	Directional Sim. (↑)
Baseline	**23.03**	7.82
Fine-tuned (Ours)	22.14	**11.91**

We suspect the lack in CLIP score improvement is due to the trade-off between higher preservation of the original image and a more expressive model.

We also saw that the model was able to fix the original example that first raised the issue and was no longer editing the black bars on the sides of our image data.

Figure 6: A corrected version of a previous example from our bounding-box model

Visually, the most noticeable change was in the improvement of the floral-related editing prompt. It seemed that the original baseline model would prioritize CLIP score too intensely, which often resulted in a blend of floral patterns that barely resembled the original human subject. Additionally, we often saw the subject's proportions being distorted with extra limbs / body parts. In figure 7, notice how the second image from each row (our new fine-tuned floral edits) resemble the original image much better. Although not perfect, we were excited to see that the edits would no longer contort the subject.

Figure 7: The results of the Bounding-Box Fine-Tuned model, showing clear improvement across all edits.

6.2 Ablation Test

In this section, we discuss the methods of the ablation study for our InstructPix2Pix model, building off of our fine-tuned model. Ablation studies have been used in neural networks to acquire a more thorough understanding of the model's overall performance by altering a new components [Sidratul et al., 2022]. We observe the change in performance and understand how the model ascertains an overall change in performance, and make relevant changes or modifications.

Within the context of this ablation test, we used the same DeepFashion2 triplet dataset plus the CLIP-based evlauation stack, but rather – training 2 epochs while toggling key hyperparameters to optimize and save time.

We use a total of 4 different ablation runs, and all four runs share the same DeepFashion2 triplet loader (with 1.5k pairs, 512^2 resolution, batch 1) and evaluation stack (CLIP-score, directional similarity, LPIPS, SSIM) on both DeepFashion2 test triplets plus the 100-image FashionPedia stream. Every configuration trains for exactly 2 epochs, with the same instruct-pix2pix backbone, but swaps the learning rate, $\lambda preserve$, inference guidance scales, and which UNet blocks are unfrozen.

The following are the 4 different configurations we used for the ablation test:

Ablation Test Configurations

- **Configuration A** ($\lambda = 0.5$, lr $= 1 \times 10^{-5}$, attn+mid+up): The baseline selective unfreezing of attention blocks, in addition to upper UNet resnets; observes the balanced preservation VS edit strength.
- **Configuration B** ($\lambda = 0.8$, lr $= 2 \times 10^{-5}$, attn+mid+up): Same layers as A but with stronger preservation weight and higher learning rate to test whether or not the aggressive regularization helps.
- **Configuration C** ($\lambda = 0.3$, lr $= 5 \times 10^{-6}$, attn+mid): Only attention and mid blocks are trainable, with a gentler learning rate and much lower preservation weight to encourage bolder edits.
- **Configuration D** ($\lambda = 0.5$, lr $= 1 \times 10^{-5}$, attn+mid+res): Adds down/up-block Resnet convolutions to A's configuration, giving the model more high-frequency capacity for cross-domain edits.

6.2.1 Quantitative Evaluation. In Table 3, we can see each of the 4 different configurations we used for the ablation test (A-D). In Figure 9, we can see the DeepFashion2 bar chart peaks with Configuration C in the Validation set, showing that a lighter λ and smaller LR allows edits to follow the textual directions without overshooting. Configuration A is close behind, but Configuration B lags due to an excessive amount of regularization.

Table 3: Ablation metrics across DeepFashion2 and FashionPedia (2-epoch runs).

Config	Dataset	CLIP (↑)	Dir. Sim. (↑)	LPIPS (↓)	SSIM (↑)
A: $\lambda = 0.5$, lr=1e−5, attn+mid+up	DeepFashion2	20.51	12.44	0.356	0.495
A: $\lambda = 0.5$, lr=1e−5, attn+mid+up	FashionPedia	19.43	9.52	0.409	0.488
B: $\lambda = 0.8$, lr=2e−5, attn+mid+up	DeepFashion2	20.87	11.25	0.270	0.555
B: $\lambda = 0.8$, lr=2e−5, attn+mid+up	FashionPedia	19.70	10.57	0.364	0.528
C: $\lambda = 0.3$, lr=5e−6, attn+mid	DeepFashion2	21.86	11.62	0.332	0.484
C: $\lambda = 0.3$, lr=5e−6, attn+mid	FashionPedia	20.08	10.89	0.400	0.486
D: $\lambda = 0.5$, lr=1e−5, attn+mid+res	DeepFashion2	21.72	12.34	0.297	0.492
D: $\lambda = 0.5$, lr=1e−5, attn+mid+res	FashionPedia	20.14	10.74	0.378	0.515

Table 4: Training/validation behavior for the four ablation configurations (2 epochs each).

Configuration	Train Loss (E1→E2)	Val Loss (E1→E2)
A: $\lambda = 0.5$, lr=1e−5, attn+mid+up	0.0405 → 0.0412	0.0433 → **0.0400**
B: $\lambda = 0.8$, lr=2e−5, attn+mid+up	0.0513 → 0.0524	0.0484 → **0.0545**
C: $\lambda = 0.3$, lr=5e−6, attn+mid	**0.0345 → 0.0354**	**0.0355 → 0.0333**
D: $\lambda = 0.5$, lr=1e−5, attn+mid+res	0.0415 → 0.0418	0.0396 → 0.0417

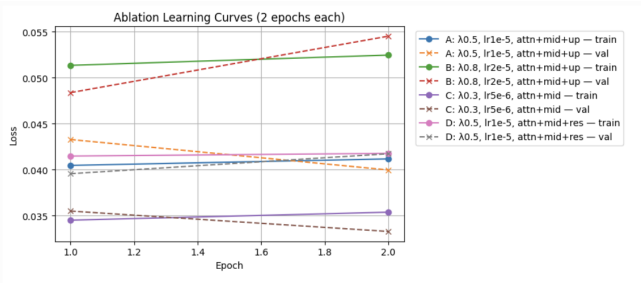

Figure 8: "Ablation Model Performance: Learning Curves, Loss VS Epoch"

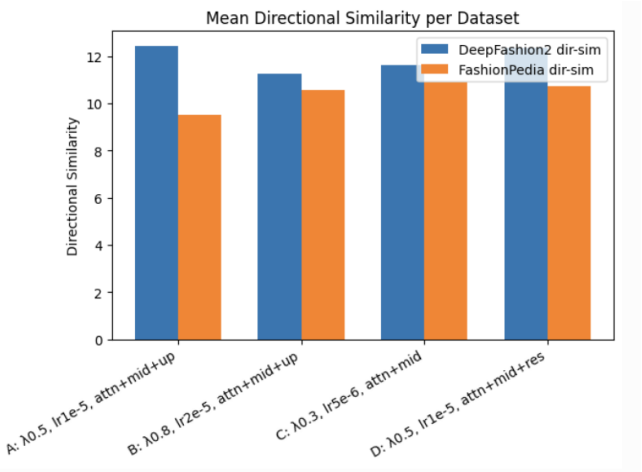

Figure 9: "Ablation Model Performance: Mean Directional Similarity per Dataset"

In the ablation study, we also use a LPIPS score [Zhang et al., 2018], which measures the perpectual similarity between the values. We wanted to introduce a new metric to simply see how it works in this realm and how it varies with different hyperparameters.

On the FashionPedia dataset, the extra down/up-block capacity in Configuration D nudges the directional similarity and CLIP scores way above the other, suggesting it does a much better job at preventing overfitting as it performs exceptionally with transferring to the validation stream.

In the end, Configration C also balances the lower LPIPS score with the respectable SSIM on the DeepFashion2 dataset, which implies that the edits remain perceptually close to the targets, without destroying the underlying texture. Configuration D shows the overall best SSIM on FashionPedia, which is well consistent with freezing fewer layers in lower resolutions, whereas Configuration B again trails due to high λ inhibiting meaningful changes but also injecting noise (higher LPIPS, lower SSIM).

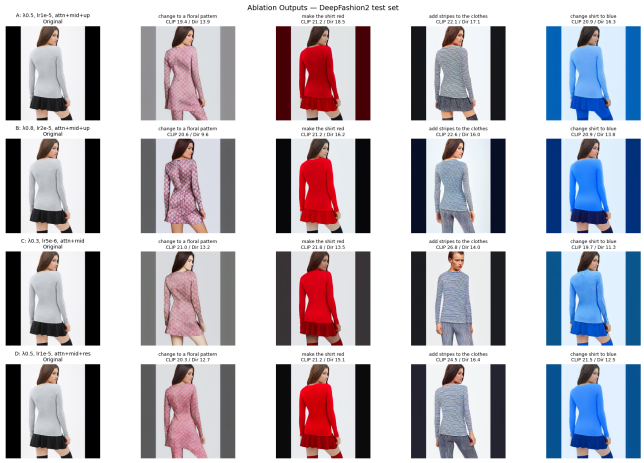

Figure 10: "Ablation Model Performance Configurations – on DeepFashion2 Test Set"

The DeepFashion2 grid performance configurations in Figure 10 shows that Configuration C produces the most overall faithful edits, with florals/striples appearing exactly on garments with minimal if no background spillover, plus color prompts are fully satisfied. Configuration A falls close behind, with edits being recognizable but textures sometimes bleeding into the pants or background. Configuration B looks unchanged, with high preservation weight and large LR keeping the model from committing to very strong edits, and occasional additions end up creeping in. Finally, Configuration D makes the requested edits but softens details since the additional Resnet layers allow broader changes.

The FashionPedia grid shows that Configuration D generalizes best, with the pose, identity, and lighting well satisfied (but a few mistakes especially with the floral prompt and the blue prompt). On the other hand, Configuration C produces solid edits but on some photos the change is way more subtle. Configuration A and B struggle on the domain shift a lot, with backgrounds completely changing and models changing. Overall, Configuration C and D

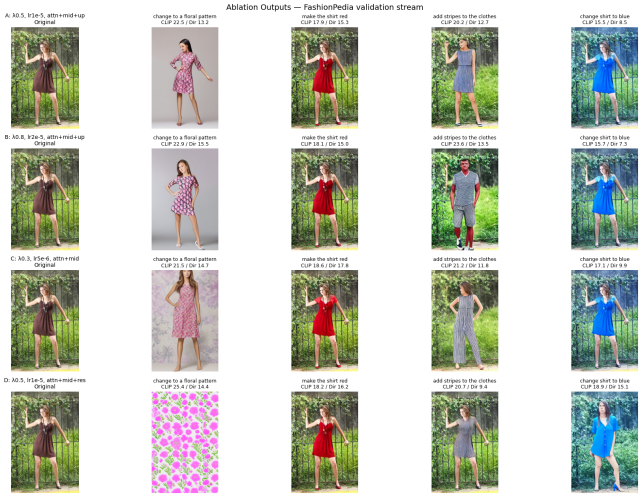

Figure 11: "Ablation Model Performance Configurations – on FashionPedia Test Set"

with modifications to penalize background shifts and training on more epochs would likely perform the absolute best.

7 Evaluation

We evaluate the quality of the edits using the following metrics:

- **CLIP Similarity (↑)**: Checks how well the edited image matches the meaning of the text prompt.
- **CLIP Directional Similarity (↑)**: Checks whether the change from the original image to the edited image follows the same semantic direction as the edit instruction.
- **LPIPS (↓)**: Quantifys how much of the non-edited regions of the image are preserved.

7.1 Baseline Evaluation (FashionPedia)

We conducted baseline evaluations of the InstructPix2Pix model [Brooks et al., 2023] on images from the FashionPedia dataset in order to establish a comparison performance before we do domain specific fine-tuning. For each image, we applied four fashion edit prompts and generated edited outputs using the pretrained Insturct-Pix2Pix model. We evaluated each edit with two metrics:

(1) CLIP Similarity Score, measuring text–image alignment

(2) CLIP Directional Similarity, measuring whether the change in the image aligns with the semantics of the edit instruction.

Qualitative Observations:

The pretrained InstructPix2Pix model successfully responds to edit instructions, but still shows noticeable variation across samples and prompts.

- Pattern edits generally showed the largest improvements in CLIP score and directional similarity values.
- The color based edits showed a mixed performance as some colors changes performed well but others produced negative directional alignment or minimal improvement.
- The negative delta values suggest that InstructPix2Pix might be generating edits that conflict with the text prompt.

Figure 12: Baseline Model on FashionPedia: "make the clothes this person is wearing red"

Figure 13: Baseline Model on FashionPedia: "change the clothes to a floral pattern"

- The results show that the variability depends on the sample largely as the same prompt behaves differently depending on the pose, lighting or original clothing complexity. This indicates that the baseline pretrained InstructPix2Pix model is not reliable in changing fashion specific attributes.
- Looking at the outputted images, it shows that the pattern and color edits (specifically patterns) alter large regions instead of localized edits. This might explain why the pattern edits score higher but the results are inconsistent as they apply changes to non clothing regions. The color edits also sometimes fail to isolate the clothing.

7.2 Fine-Tuned Model (DeepFashion2)

Table 5: Per-prompt CLIP Similarity, Directional Similarity, and LPIPS for the Fine-Tuned Model

Prompt	CLIP Score (↑)	Directional Similarity (↑)	LPIPS (↓)
add stripes to the clothes	23.25	10.28	–
change shirt to blue	22.18	6.18	–
change to a floral pattern	20.10	13.50	–
make the shirt red	23.46	13.94	–
Mean	**22.25**	**10.98**	**0.3901**

In this evaluation, we test the fine-tuned model on the held-out DeepFashion2 test set, where segmentation masks are available. During inference, the model receives the same fashion editing prompts but now applies edits only within the predicted garment region, preventing any modification of skin tone, background, or pose.

7.2.1 Quantitative Results: **Clip Scores** The model achieves a mean CLIP similarity score of **22.25**, indicating a strong text-image alignment across all prompts. In particular:

- Color edit prompts show the highest CLIP scores, reflecting the model's strong ability to follow explicit color change instructions.
- Previously in the baseline model (**Figure 1**) we noticed poor results with pattern prompts, however the model performed relatively well on patterns with a score of 23.25 for adding stripes and 20.10 for adding floral pattern. This is expected as pattern synthesis interacts more in depth with texture, lighting and garment shape.

Directional Similarity Directional similarity captures how well the direction of the edit aligns with the semantic direction implied by the text. The model obtained a mean directional similarity of **10.98**, with:

- Pattern edits producing the strong directional alignment (13.50 for floral change), indicating a highly learned transformation for structured based modifications.
- Pure recoloring edits showed lower values, reflecting that recoloring is a more localized edit and products smaller global feature shifts.

LPIPS Although per-prompt LPIPS was not computed for every sample, the mean LPIPS score of **0.3901** reflects moderate pixel level change concentrated in the clothing region, which is consistent with mask-aware behavior. This is lower than typical values for globally edited diffusion models, showing that the model preservers identity, facial details, and background while modifying only the clothing.

Overall, our metrics support that the model performs strong text aligned edits while maintaining strong high structure and perceptual fidelity.

7.2.2 Qualitative Results: The resulting behavior demonstrates highly effective mask-aware garment editing:

Highly Accurate Color Transformations

For prompts like "make the shirt red" or "change shirt to blue": The model produces consistent, saturated colors that remain realistic under various lighting conditions. It adapts to garment wrinkles, preserving shading and structure while recoloring fabric naturally. This is strongly supported by the high CLIP scores for these edits.

Patterns

The fine-tuned model reliably produces floral or striped patterns. The model avoids global recoloring and instead applies patterns only to the masked regions. The pattern alignment also remains stable. This correlates with the high dimensional similarity values for the pattern-edit prompts.

Identity and Background Preservation

Because mask-aware training explicitly penalizes undesired background changes:

- Facial features, hair, and skin tones remained relatively untouched
- The backgrounds stayed perfectly unchanged.

This is a substantial improvement over the baseline, which frequently recolored large portions of the image.

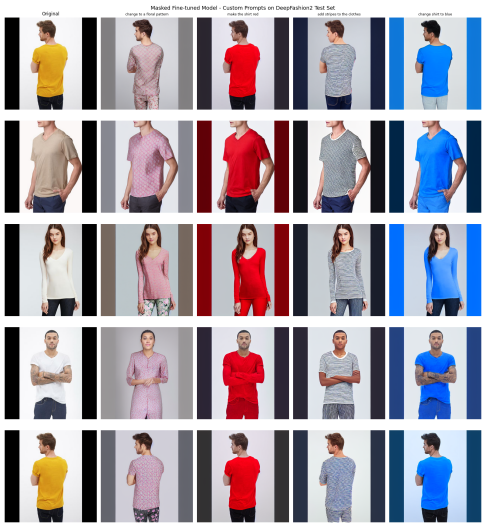

Figure 14: Fine Tuned Model On DeepFashion2 Dataset

As we can we see in **Figure 14**, we observe consistent performance across a variety of poses including frontal, side-view and back-view. The model also generalizes across both female and male clothing styles, indicating it did not overfit to a narrow clothing subset.

Failures

- Very fine stripes occasionally blur.
- The combination of highly detailed patterns with blurry or busy backgrounds cause a small struggle in localization of edits.

Overall, the fine-tuned model produces high quality, clothing localized edits using segmentation masks. It demonstrates semantic alignment, pattern understanding, and preservation of non-edited regions, significantly outperforming the baseline.

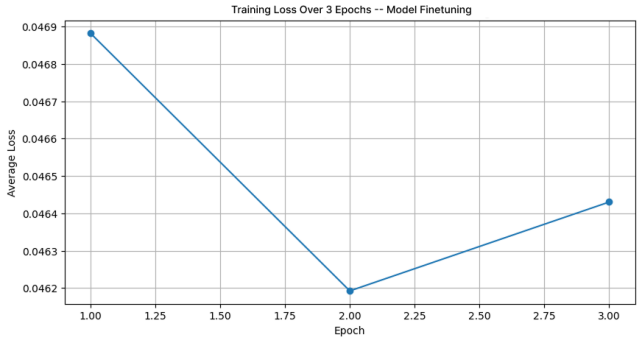

Figure 15: "Training Loss over Epochs - DeepFashion2 Fine-tuning"

While the core denoising objective, $\|\epsilon - \epsilon_\theta\|^2$, follows the standard diffusion loss which is used in Stable Diffusion and InstructPix2Pix, the spatially structured formulation that we introduce is not present in the original literature presented. In particular, this division of the loss into a tight garment zone, an expanded context zone, and a strict background-preservation zone constitutes a strong, novel adaptation inspired by the trimap-based matting, but not applied to diffusion editing. Unlike prior work that relies heavily on global $\lambda_{\text{preserve}}$ or CLIP-based perceptual guidance, or loss embeds region-specific weighting into the noise prediction, which largely enables the model to learn fine-grained fashion edits, not needing explicit percpetual and text-alignment gradients. This mask-aware loss leads to more precise control over where edits occur, which overall distinguishes this method from existing diffusion editing objectives. Plus, as seen in Figure 15, our loss converges down from Epoch 1 to 2, and slightly goes back up in Epoch 3 due to randomness, and with more fine-tuning and training, we are confident this novel approach and configuration would converge well.

7.3 Fine-Tuned Model (FashionPedia)

Table 6: Baseline vs. Fine-Tuned Model Performance on FashionPedia

Metric	Baseline	Fine-Tuned
CLIP Score (↑)	22.81	18.59
Directional Similarity (↑)	6.98	10.52
LPIPS (↓)	0.5467	0.2882
Mean CLIP Score (↑)		18.59
Mean Directional Similarity (↑)		10.52
Mean LPIPS (↓)		0.2882

Our next evaluation assesses how well the fine-tuned model generalizes to unseen fashion images using FashionPedia. Unlike DeepFashion2 [Ge et al., 2019], FashionPedia contains higher variance lighting conditions, diverse background environments, and more extreme poses [Zhu et al., 2019]. The model never sees FashionPedia during training so it serves as a strong test for edit accuracy.

7.3.1 Quantitative Results: In **Table 6** we see three key differences between the baseline and fine-tuned model when evaluated on FashionPedia dataset:

Clip Score

The fine tuned model recieved a lower CLIP score (18.59 compared to baseline 22.81) This is expected because:

- CLIP Similarity penalized edits that preserve background and identity, whereas the baseline frequently performs strong global hallucinated edits, which artificially inflate CLIP similarity at the cost of realism and localization.

Therefore, the lower CLIP score reflects safer and more localized edits rather than weaker text alignment. This is consistent with prior diffusion model literature [Brooks et al., 2023], that better preservation often reduces CLIP similarity but improves visual quality.

Directional Similarity

Directional similarity improved substantially from baseline to fine-tuned model (6.98 to 10.52), showing that the fine-tuned model better captures **semantic direction** of the edit, especially for modifications like patterns and recoloring.

This indicates that that the edits align more consistently with the requested transformation, and the model learned a fashion specific edit direction that transfers to unseen data.

LPIPS

LPIPS drops significantly from 0.546 t0 0.2882, demonstrating that the fine-tuned model introduced fewer unwanted global changes:

- Less background alternation
- Cleaner garment boundaries
- Smoother transitions

This reduction is a major improvement in perceptual image fidelty compared to the baseline.

Overall, the metrics show a clear pattern that the fine-tuned model performs more targeted, semantically accurate edits with much fewer unwanted image changes.

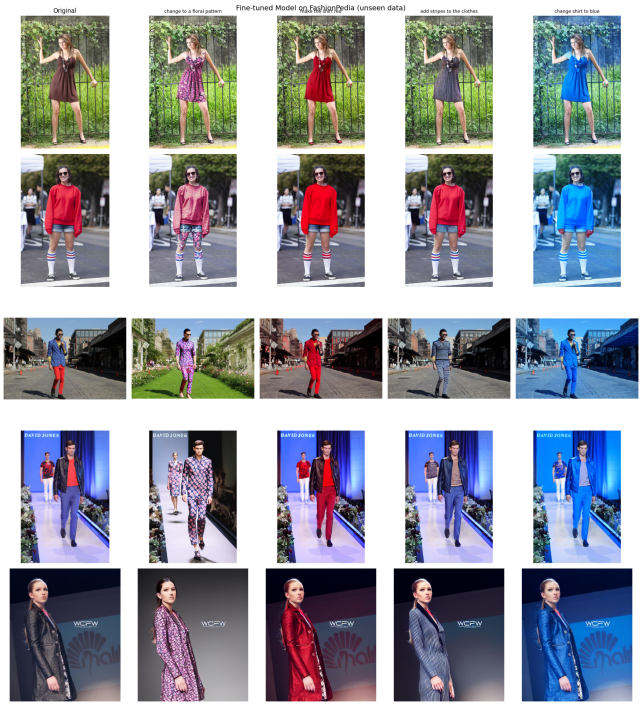

Figure 16: Fine Tuned Model On FashionPedia Dataset (Unseen)

7.3.2 Qualitative Results: Across a wide range of prompts, the fine-tuned model demonstrates marked improvements in both edit precision and garment localization:

Improved Edit Localization

- The model consistently applies edits only to clothing regions.
- Color edits remain tightly bounded to shirts, pants, or jackets even when the background is cluttered. However, the baseline InstructPix2Pix frequently leaked color onto skin or background so the fine-tuned model does a good job at avoiding this.

Impoved Texture and Patterns

The model generated coherent textures even for complex instructions like floral patterns and striped designs. The patterned edits align correctly with clothing folds and body pose, indicating the model learned fashion-specific structure because in the baseline model, texture and patterns are a major factor the model struggled with.

This indicates that the fine-tuned model learned editing behaviors that are transferable and not memorized from DeepFashion2.

Failures

- Cluttered backgrounds sometimes cause localization degrade as it struggles to identify boundaries.
- The model sometimes fails to generate the same shade of color based on the prompt.
- Model occasionally distors facial features.
- Highly patterned clothing occasionally retain traces of the original pattern under the new one.

These are significantly less frequent than in the baseline model.

Overall, the fine-tuned model generalizes strongly to FashionPedia despite never training on it with improvements in directional similarity and LPIPS paired with more localized and semantically accurate edits. It performs well across real-world fashion conditions and demonstrates accurate, localized garment manipulation.

8 Conclusion

In this final project, challenges surrounding text-guided image editing were explored – within the world of fashion. Overall, our study demonstrated that general-purpose diffusion editors including are very limited when applied to fine-grained, apparel-specific transformations. Through a thorough evaluation of dataset construction, parameter tuning, mask-aware training, and ablation-based hyperparameter analysis, we demonstrate that domain specialization is not only helpful but also essential to achieve realistic and semantically aligned image edits in the fashion realm.

First, we begin with the construction of supervised image-instruction-target triplets from the DeepFashion2 dataset. Through pairing images of the same individual wearing different garments and generating edit instructions through caption differences, we created a very scalable and domain-relevant pipeline for producing high-quality training data. Plus, combined with segmentation masks, this overall allowed us to embed structural fashion priors directly into the overall fine-tuning process.

The most important part of improving edit accuracy in this work was our parameter-efficient hyperparameter fine-tuning strategy, where only 23.84% of the U-Net parameters were unfrozen – more specifically, the cross-attention, self-attention, mid-block, and high-resolution up-sampling layers. This overall design preserved the benefits of using the pretrained backbone, while granting the model with enough power to learn fashion-specific texture, pattern, plus boundary information. This final result was a strong balance between compute efficiency and improved visual quality, overall outperforming LoRA-based approaches which were tested in early prototyping.

To now further guide the model towards localized edits, we worked on introducing a hierarchal loss formulation via segmentation masks and bounding-box-derived trimap zones. This overall modification helped in creating a spatially structured training objective, that well explicitly separated garment regions from the context and background. The overall resulting fine-tuned model demonstrated a way better preservation of identity, pose, and background. Plus, there were highly accurate color transformations and fabric pattern edits.

The evaluation across both the DeepFashion2 and unseen FashionPedia dataset reveals that the fine-tuned model consistently improves directional similarity, LPIPS, and qualitative edit fidelity as compared to the pretrained baseline. In particular, the model greatly excels at respecting clothing boundaries, maintaining necessary texture coherence, plus avoiding background spillover that occured in the original InstructPix2Pix [Brooks et al., 2023]. Most importantly, the improvement generalizes to diverse lighting, poses, plus garment types, demonstrating that this model has learned fashion-centric editing as opposed to simply overfitting to the training dataset.

Next, we conducted an ablation study with 4 different configurations to uncover how learning rate, preservation weighting, and selective unfreezing of UNet blocks shape the edit quality. These overall experiments illuminate the tradeoffs between the expressiveness, stabiloty, and over-regulation, which show that no single configuration performs the overall best. The ablations showed that tailoring hyperparameters is key, and some of the configurations performed better than others.

Overall, there are still several limitations of this study. Super intricate textures can strongly blur with strong edits, plus cluttered backgrounds can confuse the model's overall localization ability, plus subtle changes can also go off-prompt. For instance, some issues we saw were that the model sometimes changes aspects about an individual's character rather than their outfit, or sometimes makes edits to the background as well (especially noticable on the unseen FashionPedia dataset that has very detailed backgrounds). Overall, with bounding-box-derived trimaps (inspired by [Chuang et al., 2001]) – these can sometimes restrict edits requiring strongly structural garment changes.

This work shows that fashion-focused diffusion editing works best with parameter-efficient fine-tuning, well-framed structured masking, and triplet construction. Our findings also show the overall efficiency of domain specialization in generative editing, with future extensions to further, multi-view clothes editings or even integration into real fashion design workflows which can automate processes for numerous industries. By overall refining the alignment between natural language instructions and clothing transformations, this project advances the applicability of text-based diffusion models, all through the power of generative artificial intelligence.

References

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. 2023. InstructPix2Pix: Learning to Follow Image Editing Instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yung-Yu Chuang, Brian Curless, David H. Salesin, and Richard Szeliski. 2001. A Bayesian Approach to Digital Matting. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, II–264.

Yuying Ge, Ruimao Zhang, Lingyun Wu, Xiaogang Wang, Xiaoou Tang, and Ping Luo. 2019. DeepFashion2: A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 5337–5346.

Nupur Goel, Mohit Bansal, Abhishek Gupta, and Vinay Bhargava. 2022. Imagenerator: Text-Guided Image Editing with Diffusion Models. In *Proceedings of the 2022 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2501–2510.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint* (2021). arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

A. Jones, Y. Xu, and Bren School of Environmental Science. 2024. *Examining Cut-and-Sew Textile Waste within the Apparel Supply Chain.* Research Report. University of California, Santa Barbara. https://bren.ucsb.edu/sites/default/files/2024-04/Examining%20Cut-and-Sew%20Textile%20Waste%20within%20the%20Apparel%20Supply%20Chain%204.10.24.pdf Accessed: 2025-11-28.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2021. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. arXiv:2108.01073 [cs.CV] doi:10.48550/arXiv.2108.01073

Reuters. 2025. *Zalando uses AI to speed up marketing campaigns, cut costs.* https://www.reuters.com/business/media-telecom/zalando-uses-ai-speed-up-marketing-campaigns-cut-costs-2025-05-07/ Accessed: 2025-11-28.

SaffalPoosh. 2025. *deepFashion-with-masks.* Accessed: 2025-11-28.

Montaha Sidratul, Azam Sami, Rafid A. K. M. Rakibul Haque, Hasan Md. Zahid, Asif Karim, Md. Hasib Khan, Shobhit K. Patel, Mirjam Jonkman, and Zubaer Ibna Mannan. 2022. MNet-10: A robust shallow convolutional neural network model performing ablation study on medical images assessing the effectiveness of applying optimal data augmentation technique. *Frontiers in Medicine* 9 (2022). doi:10.3389/fmed.2022.924979

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rijeber, Patrick von Platen, Matthieu Ma, et al. 2020. Transformers: State-of-the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (2020), 38–45.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. 2023. MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 586–595. https://richzhang.github.io/PerceptualSimilarity/

Yueru Zhu, Ziwei Yang, Zhenxiang Zhang, Ying Xue, Cao Luo, Congmo Sun, Long Chen, Yizhou Zeng, Xiao-Yong Wu, Yongli Yu, et al. 2019. Fashionpedia: Ontology and Categorization of Fashion Attributes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 9106–9115.

A Google Drive Code Files

We provide access to the full project folder on Google Drive, which contains the README file and all associated Python notebooks. The folder is available at the following link:

Google Drive Project Folder[1].

This project folder includes:

- InstructPix2Pix Baseline Model
- Triplet Generation Scripts
- Mask aware loss, bounding box, and ablation test implementation
- Selective Parameter Efficient Tuning
- Full training and validation loops
- Evaluation pipeline (CLIP, Directional Similarity, LPIPS)
- Dataset Preprocessing and configuration
- Visualization and figure generation code
- All experimental logs, sample outputs and plots.

Additionally, we cite the code we obtain from online as well [Brooks et al., 2023][Goel et al., 2022][Hu et al., 2021].

[1]Project Code, README, and Notebooks: Google Drive Link.
Raw URL: https://drive.google.com/drive/folders/1ptplZw2EdzAYnnyCqZmYF5iVMDja1bQb?usp=sharing

B Statement on the Use of AI

Throughout our coding process, we used Gemini/Cursor to automate a few processes. This includes generating the print statements to see the outputs of our model and boilerplate code involving loading the model plotting the results of the model's outputs. Specifically, when debugging, we wanted clear print statements and information about epochs, so we automated some of the tasks of generating these with AI. However, we wrote all the underlying functions and dataset classes ourselves. We wrote this entire paper in latex on our own with no generative AI assistance. We only used EasyBib, a bibliography generating technology, for assistance with generating citations.