Revisiting EmoEdit: A Proposal for Context-Aware Affective Image Manipulation

Bairui Li [1] **Zhichen Pan** [2]

Abstract

Affective Image Manipulation (AIM) tries to change images so that they trigger specific emotions while keeping the original layout and meaning. The recent framework EmoEdit does this by using a new dataset (EmoEditSet) and an Emotion Adapter based on the Q-Former architecture. EmoEdit works better than traditional style transfer methods, but our early analysis shows several limits in its semantic generalization. The model often falls back to repeated and stereotypical visual cues (for example, balloons for "Amusement"). It is also very sensitive to prompts: using "angry" can make a face look angry, but using the synonym "mad" often barely changes the image. In this course project proposal, we describe the EmoEdit method, analyze its structural and semantic weaknesses, and propose an improvement plan. Our main idea is to use Large Multimodal Model (LMM) guided dynamic instruction generation to improve semantic diversity and context awareness.

1. Introduction

Emotions are closely linked to visual perception. The field of Visual Emotion Analysis (VEA) tries to understand these links. However, actively changing an image to guide a viewer's emotional response, known as Affective Image Manipulation (AIM), is still very difficult. A good AIM system must satisfy two goals that often conflict: (1) **Evoking the intended emotion** with high accuracy, and (2) **Preserving the original image composition** and structure (Yang et al., 2025).

Existing generative models usually fail to balance these goals. In our preliminary tests, text-to-image models like DALL-E 3 can create images with strong emotions, but they often invent new scenes and break the original structure. In contrast, editing models like InstructPix2Pix (IP2P) keep the structure well but often do not understand abstract emotion commands, so they make only small or meaningless changes.

EmoEdit (Yang et al., 2025) tries to fill this gap. It introduces a content-aware framework that adds an "Emotion Adapter" into diffusion models. By learning from a dataset of emotion-instruction pairs, it aims to inject emotional meaning into the generation process.

In this proposal, we plan to reproduce the EmoEdit framework and then address its limits in semantic stereotypes and generalization.

2. Methodology Review

2.1. EmoEdit Framework Overview

The EmoEdit framework has two main parts: the construction of EmoEditSet and the Emotion Adapter architecture. Together, they define how emotional meaning is extracted, represented, and added to the diffusion-based editing process. In this section, we describe these parts in a unified way to show how they interact.

Data construction: EmoEditSet. The authors argue that raw emotion labels (for example, "Sadness", "Awe") are too abstract for direct image editing. Instead, they design a multi-stage pipeline to extract concrete and fine-grained visual cues that can be used during diffusion editing.

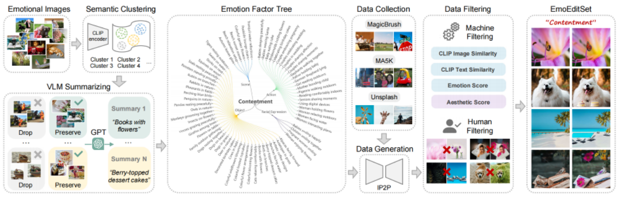
The process starts with CLIP-based emotion attribution. Images in EmoSet are encoded using CLIP ViT-L/14 embeddings. Rather than using only categorical emotion labels, images are clustered *within each emotion* to find hidden groups. These clusters reveal recurring visual patterns, such as "dark forests" or "harsh lighting" for "Fear", that the model can later use as editing handles. This clustering step is motivated by the high variation inside each emotion class. For example, two "Awe" images may be very different (grand landscapes vs. cathedral interiors), which cannot be captured by label supervision alone.

A multimodal LLM (GPT-4V) is then prompted to summa-

[1] College of Engineering, Peking University, Beijing, China
[2] School of Electronic Engineering and Computer Science, Peking University, Beijing, China.

rize the discovered clusters. The prompts are designed to reduce hallucinated features and instead focus on common elements across cluster images. The output is a tree-like **Emotion Factor Tree**, where coarse emotional descriptions split into fine-grained factors that can be turned into concrete editing commands.

Each emotion factor is then converted into natural-language editing instructions (for example, "Add warm sunlight" or "Increase color vibrancy"). InstructPix2Pix (IP2P) generates image-edit pairs for these instructions. Because automatic generation is noisy, EmoEdit applies a multi-stage filtering pipeline. It uses: (i) *CLIP image similarity* to ensure structure preservation, (ii) *CLIP text similarity* to check alignment with the instruction, (iii) an *aesthetic score* to remove low-quality results, and (iv) an *emotion score* from an external classifier to check emotional correctness. This pipeline increases dataset precision, but it also inherits biases from the emotion classifier and the CLIP feature space. Later, we argue that this bias is a key reason for the "visual cliches" in the edited results.



(a) Emotion Attribution (b) Data Construction

Figure 1. The pipeline for constructing EmoEditSet. CLIP-based clustering extracts emotion factors, which are turned into editing instructions and filtered by multiple semantic metrics.

Emotion Adapter architecture. A main innovation of EmoEdit is the Emotion Adapter. It injects emotional control signals into a frozen diffusion U-Net. The adapter is inspired by the Q-Former architecture used in BLIP-2 (**?**). It provides a lightweight but expressive way to merge textual emotion representations with image features.

The adapter uses learnable queries $q \in \mathbb{R}^{N \times d}$ as an intermediate bottleneck between the emotion text embedding e_t and the U-Net's visual latents. These queries help the model extract task-relevant emotional features without changing the base U-Net. This design helps preserve structure during editing.

The adapter has two attention stages. First, the queries attend to e_t and produce an emotion-conditioned latent representation:

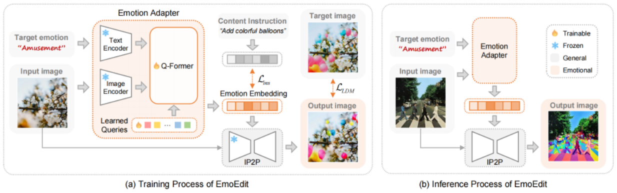
$$A_s = \text{softmax}\left(\frac{[q;e_t]W_q^s([q;e_t]W_k^s)^T}{\sqrt{d_k}}\right)[q;e_t]W_v^s. \quad (1)$$

This step turns the abstract emotional intent into a compact set of latent tokens. Next, the emotion-aware queries interact with U-Net features e_i through cross-attention:

$$A_c = \text{softmax}\left(\frac{A_s W_q^c(e_i W_k^c)^T}{\sqrt{d_k}}\right)e_i W_v^c, \quad (2)$$

so that emotional edits depend on the spatial and semantic context of the input image.

The adapter is added to the deeper layers of the U-Net, which contain high-level semantics rather than low-level texture. This choice helps avoid over-editing of fine structure. It encourages changes in lighting, mood, or global composition, which match the goals of affective manipulation.



(a) Training Process of EmoEdit (b) Inference Process of EmoEdit

Figure 2. The architecture of EmoEdit. Emotion-conditioned queries change U-Net feature maps through cross-attention.

Training objectives. The model is trained with a combined objective:

$$\mathcal{L}_{total} = \mathcal{L}_{LDM} + \lambda\mathcal{L}_{ins}. \quad (3)$$

The standard diffusion loss keeps good reconstruction and stops the adapter from collapsing the latent distribution:

$$\mathcal{L}_{LDM} = \mathbb{E}_{\varepsilon(x),c_i,c_e,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \varepsilon(c_i), c_e)\|_2^2\right]. \quad (4)$$

A key idea of EmoEdit is that emotional editing should be *operational*, not only about emotion classification. Thus, the adapter learns to align emotional embeddings with instruction embeddings:

$$\mathcal{L}_{ins} = \frac{1}{M}\|c_e - \mathcal{E}_{txt}(t_{ins})\|_2^2. \quad (5)$$

This encourages the model to learn actionable emotional changes, such as "brighten the scene to evoke hopefulness", instead of directly mapping images to emotion labels. However, instruction embeddings also reflect dataset bias. This helps explain why the model often produces stereotypical objects such as balloons or flowers.

3. Critical Analysis and Limitations

EmoEdit is an important step for Affective Image Manipulation, but our extended review and early tests show several limits. These limits come from the dataset design, the adapter architecture, and the training objective.

3.1. Limitations Acknowledged by EmoEdit

The authors point out that the Emotion Factor Tree is limited by the discrete, human-defined clusters from CLIP. Emotional expression in images is continuous and also depends on culture and context. Any fixed taxonomy will miss part of the full range of affective cues. In addition, the filtering step depends on a pre-trained emotion classifier. This adds systematic bias: the system prefers visual patterns that the classifier can easily detect. As a result, there is a feedback loop between dataset generation and classifier constraints.

3.2. Observed Semantic Collapse and Generalization Issues

Beyond these known issues, we see a type of **Semantic Collapse**. The model tends to converge to high-frequency and stereotypical cues during inference. The “Balloon Problem” and “Flower/Cake Problem” are clear examples. EmoEdit often encodes emotional change as the simple presence or absence of certain objects. It uses fewer subtle changes like lighting, color temperature, shading, or texture. This behavior suggests that the adapter learns a narrow mapping between emotion embeddings and high-level semantic tokens in the U-Net. It then skips more fine-grained visual channels.

Because the Emotion Adapter changes only high-level U-Net latents, it also struggles with strong contextual constraints. For example, when we try to add “bright fireworks” to an indoor scene, the result often has clear artifacts. This suggests weak grounding between the adapter’s emotional cues and the scene’s geometry and semantics. We also observe a degeneracy in embedding space: emotion embeddings c_e tend to cluster very tightly inside each category after training. This reduces the effective diversity of emotional transformations. Many images with the same target emotion receive almost identical edits.

3.3. Prompt Sensitivity and Cross-Modal Inconsistency

Another important limit is prompt sensitivity and instability. EmoEdit reacts strongly to small changes in the text prompt. For example, the words “angry” and “mad” are close in meaning, but they cause very different outputs. “Angry” leads to strong facial expression changes, while “mad” often causes almost no edit.

We think this comes from two sources. First, the **text encoder granularity** is uneven. The CLIP text encoder does not cover all affective words equally well. Rare synonyms may lie in poorly trained regions of the embedding space. Second, the **adapter capacity** is limited. A small number of learnable queries may not be enough to separate fine-grained lexical differences. This prompt instability reduces the reliability of EmoEdit in real settings, where users may use varied language.

The current training scheme also lacks cross-modal consistency constraints. There is no direct mechanism to keep the instruction t_{ins}, the edited image $\hat{x}$, and a VLM’s reading of that image aligned. Without this triangulation, the adapter can overfit to shallow emotion cues instead of learning robust affective changes grounded in multimodal meaning.

4. Proposed Research Plan

Based on these limits, we propose a three-phase research plan. It aims to improve dataset generation, semantic diversity, and training stability. Our goal is a context-aware affective editing system that avoids stereotypical visual cues and produces more subtle emotional changes.

4.1. Phase I: Reproduction and Diagnostic Analysis

In **Phase 1**, we will reproduce the official EmoEdit implementation and set up a strong baseline. We will first check the reported quantitative metrics (PSNR, SSIM, Emo-A). Then we will qualitatively confirm Semantic Collapse using a test set with diverse scenes, including portraits, landscapes, indoor scenes, and night scenes.

We will also design a Cross-Emotion Consistency Test. This test will measure whether edits for different emotions collapse into similar visual cues. In addition, we will visualize the embedding space and write a detailed reproduction report. The report will document failure cases and show how the adapter behaves under adversarial or tricky prompts.

4.2. Phase II: LMM-Guided Dynamic Instruction Generation

In **Phase 2**, we will replace the static Emotion Factor Tree with a dynamic, image-conditioned instruction generator. This generator will be built using Large Multimodal Models (for example, GPT-4o or LLaVA-Next). For each training image, we will prompt an LMM to first describe the scene. Then it will propose an editing instruction that targets a given emotion and respects the scene context. We will also ask the LMM to avoid cliches such as balloons, fireworks, or flowers.

This design brings three benefits. First, the instructions reflect the specific content of each input image. Second, they reduce repetitive object insertion. Third, they focus more on subtle affective cues such as lighting, contrast, color temperature, spatial tension, and composition flow.

Instead of depending only on a standard emotion classifier, we will use VLM-based emotion verification. For each edited image, we will extract a caption that describes the emotion. We will then compare this caption with the target emotion using text similarity and VLM-based scoring. This dynamic pipeline increases the entropy of EmoEditSet and allows the adapter to learn a richer and more continuous emotional space.

4.3. Phase III: Diversity-Regularized and Consistency-Aware Training

In **Phase 3**, we will add diversity-regularized training to fight overuse of common emotional cues. We also want to encourage a wider set of emotional strategies. We plan to use an object detector (for example, Grounding-DINO) to detect frequently inserted objects. Based on this, we will add a diversity penalty:

$$\mathcal{L}_{div} = \alpha \sum_{o \in O} f(o)^2, \tag{6}$$

where $f(o)$ is the frequency of object o in edits for the same emotion. This term discourages overuse of stereotypical objects.

We will also encourage higher entropy in the adapter's output embeddings and thus prevent collapse:

$$\mathcal{L}_{ent} = -\beta H(A_c). \tag{7}$$

To keep the original scene meaning, we propose a VLM-based scene consistency loss:

$$\mathcal{L}_{scene} = \gamma \cdot \mathrm{KL}\big(P_{\text{scene}}(x) \parallel P_{\text{scene}}(\hat{x})\big), \tag{8}$$

which encourages the original and edited images to share similar scene-level descriptions.

The final objective becomes:

$$\mathcal{L}_{total} = \mathcal{L}_{LDM} + \lambda\mathcal{L}_{ins} + \alpha\mathcal{L}_{div} + \beta\mathcal{L}_{ent} + \gamma\mathcal{L}_{scene}. \tag{9}$$

We expect the trained model to insert fewer stereotypical objects, show richer emotional variation, and achieve better scene-aware consistency. It should also be more robust across diverse contexts.

5. Conclusion

EmoEdit brings content-aware editing to AIM but still shows semantic rigidity and stereotypical emotional expressions. We propose to move from static Emotion Factor Trees to dynamic, context-aware instruction generation using modern VLMs. Combined with diversity and consistency regularization during training, this may lead to an editing model that evokes emotions through more diverse and subtle changes. In this way, we can move beyond simple object insertion and towards richer affective control.

References

Brooks, T., Holynski, A., and Efros, A. A. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402, June 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Yang, J., Feng, J., Luo, W., Lischinski, D., Cohen-Or, D., and Huang, H. Emoedit: Evoking emotions through image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24690–24699, June 2025.

(Yang et al., 2025) (Brooks et al., 2023) (Rombach et al., 2022) (Radford et al., 2021)