RefRetouch: Personalized Image Retouching without Test-time Fine-tuning

Temesgen Muruts Weldengus[1], Binnan Liu[1], Fei Kou[2], Youwei Lyu[2], Jinwei Chen[2], Qingnan Fan[2†], and Changqing Zou[3,1†]

[1] Zhejiang University
[2] vivo BlueImage Lab
[3] Zhejiang Lab
tememuruts@zju.edu.cn, changqing.zou@zju.edu.cn, fqnchina@gmail.com

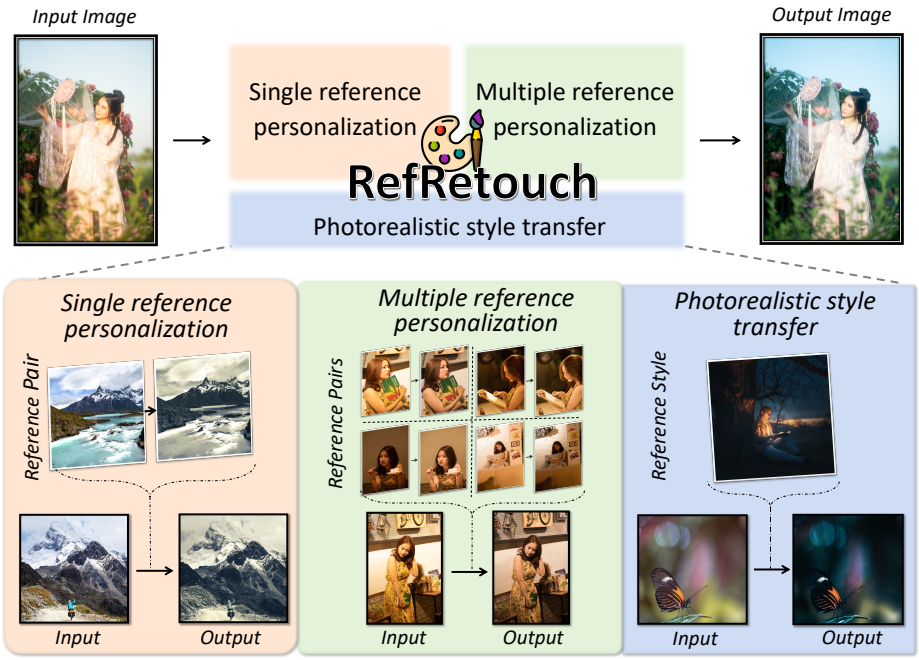

Fig. 1: Our proposed method effectively personalizes image retouching from single or multiple references of similar or diverse styles at inference time, with no test-time finetuning. It also supports photorealistic style transfer out of the box.

Abstract. Personalized image retouching aims to adapt retouching styles of individual users from reference examples, but existing methods often require user-specific fine-tuning or fail to generalize effectively. To address

†Corresponding authors.

these challenges, we introduce **RefRetouch**, a general framework for personalized image retouching that instantly adapts to user retouching styles without any test-time fine-tuning. It employs an *asymmetric auto-encoder* to encode the retouching style from paired examples into a content disentangled latent representation that enables faithful transfer of the retouching style to new images. To adaptively apply the encoded retouching style to new images, we further propose *retrieval-augmented retouching* (RAR), which retrieves and aggregates style latents from reference pairs most similar in content to the query image. With these components, **RefRetouch** enables superior and generic content-aware retouching personalization across diverse scenarios, including single-reference, multi-reference, and mixed-style settings, while also generalizing out of the box to photorealistic style transfer.

1 Introduction

Image retouching enhances photographs to improve aesthetics or convey a specific narrative, but it is an inherently subjective task [20, 37], with user preferences varying widely. Existing methods [12, 14, 31, 45, 47], while automating the process, often learn fixed styles and require large datasets and retraining to adapt to new users, which limits their practical application. To address this, image retouching approaches need to be able to adapt to user preferences using only a few reference examples, enabling personalized retouching at test time without the need for retraining or extensive datasets.

Several works have explored generic personalized image retouching by conditioning on user-provided reference images [20, 24, 37]. While these methods demonstrate promising results, their effectiveness is fundamentally constrained by their weak representation of users' retouching styles. Specifically, the methodologies adopted by these approaches such as style classification [37], metric learning [20], and simple auto-encoding [24] struggle to capture retouching styles in a manner that is both disentangled from image content and transferable to unseen images. Additionally, they lack the ability to integrate information from multiple reference images, which prevents coherent retouching style fusion from multiple user references and constrains their ability to handle personalized image retouching scenarios involving diverse or mixed styles.

To overcome these limitations, we introduce **RefRetouch**, a generic personalized image retouching framework that accurately captures a user's retouching style from a small set of reference examples and adaptively applies it to new images based on their visual content. Central to our approach is an *Asymmetric Auto-Encoder* which encodes the retouching style between an original image and its retouched version into a high-level latent representation. The architecture comprises a twin encoder network with shared weights that processes the input and its corresponding retouched target, capturing the color and tone transformations while abstracting away the semantic content. This resulting latent representation is then used as a condition to control a lightweight decoder, implemented as a conditional multilayer perceptron (MLP) operating in the color

domain, which faithfully reconstructs the retouched image from the original input. This design disentangles retouching style from image semantics, enabling precise and consistent transfer of the encoded retouching style to new query images without any test-time fine-tuning.

Having encoded the retouching style of a pair of images into disentangled latent representations, we introduce *retrieval-augmented retouching (RAR)* to personalize and adaptively retouch new images using style information derived from a user's preference examples. RAR performs content-aware retouching by retrieving the top-K most relevant reference pairs from the user's preference set based on visual similarity to the new input image. The corresponding retouching style latent codes of these retrieved examples are then aggregated via a weighted average, producing a composite retouch style that reflects the most relevant color and tone transformations for the current input. This aggregated retouching style latent, along with the input image, is then passed to the conditional MLP decoder to obtain the final retouched output. Through this retrieval-based conditioning, our method adapts the retouching style not only to the user's overall preferences but also to the content and context of each input image.

RefRetouch supports personalized image retouching using one or multiple reference examples with the same or varying styles, as shown in Figure 1. Moreover, it generalizes seamlessly to photorealistic style transfer with an unpaired single reference image [15, 19, 46], without any retraining, by simply reusing the new input image as a pseudo reference input. This demonstrates the model's ability to encode retouching style even from semantically different images. In summary, our contributions are as follows:

- A generic framework for personalized image retouching to instantly adapt a user's retouching style from few reference examples.
- An Asymmetric Auto-Encoder that encodes user preferences into a content-disentangled retouching style latent representation.
- A retrieval-augmented retouching (RAR) module that enables content-adaptive image retouching personalization by combining relevant reference styles.

2 Related Work

Image retouching Numerous approaches have been proposed to automate image retouching, ranging from traditional enhancement methods to modern learning-based techniques. Many works [12, 22, 23, 30, 35, 45, 47] optimize classical enhancement operators using paired datasets of input and expert-retouched images. Others adopt convolutional networks or multilayer perceptrons to directly map input images to retouched outputs [6, 14, 18, 41, 43]. While effective at modeling user's retouching style, these methods apply a fixed retouching style at test time, limiting their applicability in real-world scenarios where user preferences are diverse and inherently subjective. White-box approaches [17, 31, 36] aim to improve user control by exposing interpretable pipelines and allowing manual adjustments to auto-retouched results. However, they still rely on models trained to mimic a specific expert's style and require active user interaction

Fig. 2: Image retouching style transfer from a single reference pair. *Top*: input–output reference pair defining the retouching style. *Second row*: query input images. *Third row*: our method's results. *Bottom*: ground-truth retouched images.

in the editing process. Methods such as [11, 39, 44] demonstrate that fine-tuning a model on one or a few style image pairs can effectively transfer user preferences to new images. Nonetheless, these methods require re-training for each new user or style, which limits their practicality in real-world settings. In contrast to all the above methods, our approach enables test-time adaptation to user preferences without any model retraining by conditioning directly on a small set of user-provided reference examples.

Generic personalized retouching Several methods [9, 21, 40] train a model on diverse retouching styles to produce multiple outputs from a single model at inference time. However, they do not support personalization via user-provided reference examples. Recent approaches use multi-modal large language models for text-guided image retouching [4, 10, 29], but textual descriptions are often too vague and abstract to precisely capture a user's nuanced retouching preferences. Moreover, the subtle adjustments of color and tone in image retouching are difficult to describe using text. Methods like StarEnhancer [37], MSM [24], and PieNet [20] offer test-time personalization with a small set of user-preferred examples but they suffer from weak representations and ineffective adaptation to varying image content. Photorealistic style transfer [15, 19, 26, 46] can perform single-reference color transfer but cannot handle multi-reference personalization or content-aware adaptability. Generic image editing frameworks [2, 5, 13, 27, 34, 42] use visual in-context learning to unify various tasks but perform poorly compared to models specialized for image retouching.

3 Method

3.1 Asymmetric Auto-Encoder

In generic personalized image retouching, the first step is to effectively represent a user's retouching style from the reference example pairs. StarEnhancer [37] learns a style representation by training a classifier on a fixed set of retouching styles, but this closed-set formulation limits its ability to generalize to unseen styles. PieNet [20] uses metric learning [16] to map images with similar retouching effects close in embedding space. However, the contrastive signal used in this setup is too coarse to capture the fine-grained color and lighting variation that characterizes individual user's retouching preferences. MSM [24] adopts a U-Net with a conditional decoder to predict the residual between input and retouched images via reconstruction loss, but the resulting representation entangles style with content, preventing robust retouching style transfer to new images with different visual compositions.

To address these limitations, we propose an **asymmetric auto-encoder** whose encoder and decoder are not mirror-symmetric, both in architecture and in the spaces they operate on. The model encodes retouching style into a content-disentangled latent representation, enabling faithful reconstruction of the retouched image from its original input. As illustrated in Figure 3, our model comprises a style encoder E_θ, which extracts the retouching style from input–output image pairs, and a lightweight decoder D_ϕ, which reconstructs the retouched image conditioned on the original input and the inferred retouching style representation. Given an original image $x \in \mathcal{X}$ and its user-retouched version $y \in \mathcal{Y}$, the encoder computes a latent vector $z = E_\theta(x, y)$, which captures high-level retouching attributes such as color and tone shifts. The decoder then synthesizes the retouched output as $\hat{y} = D_\phi(x, z)$.

The encoder is implemented as a Siamese network based on the SigLIPv2 backbone [38], processing both the original and retouched images. Features pooled from both branches are concatenated and passed through a projection layer to form the final retouching style representation $z \in \mathbb{R}^d$. To effectively adapt the encoder to retouching tasks, we add trainable LoRA weights, enabling efficient fine-tuning for retouching style representation learning.

The decoder is designed as a lightweight conditional MLP that operates in color space, mapping per-pixel values from $\mathbb{R}^3 \to \mathbb{R}^3$. This design is motivated by two key factors. First, it encourages the decoder to rely on the retouching style latent z rather than the model parameters during reconstruction. It forces the latent representation to encode color and lighting transformations that are disentangled from image content, since the difference between input and retouched images is a variation in color and lighting. Second, prior works [8, 14, 45, 47] have demonstrated that operating in color space yields higher image fidelity compared to spatial-domain operators, while also being resolution-agnostic and computationally efficient.

The architecture of the conditional MLP decoder is shown in Figure 3. It consists of an input layer followed by N conditional MLP blocks, inspired by

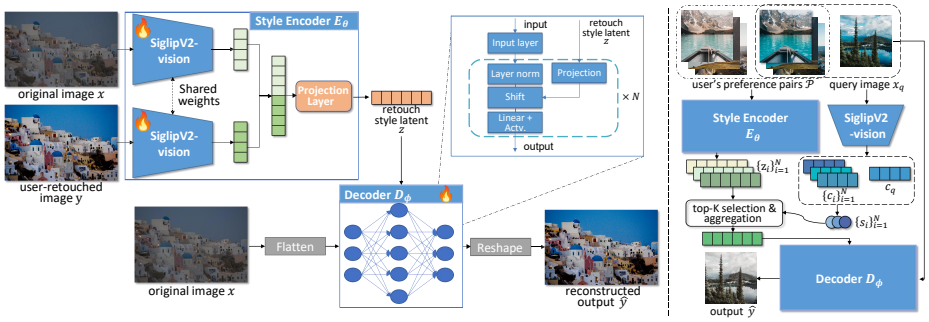

Fig. 3: Illustration of our RefRetouch pipeline. The asymmetric auto-encoder comprises a LoRA-adapted Siamese encoder and a lightweight conditional MLP decoder. The encoder processes input-output pairs to learn a content-disentangled retouching style latent, the conditional MLP decoder then applies this latent representation to reconstruct the retouched image from the original input, operating in color space to preserve content. The retrieval-augmented retouching (RAR) module enables contextual retouching personalization by selecting and combining the most relevant latents from multiple reference images.

feature conditioning in modern transformer [32]. To incorporate the retouching style latent, we project z and add it to the hidden features of each MLP block. While we also experimented with more complex conditioning strategies such as adaptive layer normalization and cross-attention, the simple additive method performed best in our setting, offering a good balance between simplicity and effectiveness.

To train the encoder–decoder pair, we minimize an ℓ_1 reconstruction loss between the predicted and ground-truth retouched images:

$$\mathcal{L}_{\text{recon}} = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\| D_\phi(x, E_\theta(x, y)) - y \|_1 \right], \tag{1}$$

where $\mathcal{D}$ is a paired image retouching dataset. This objective encourages the encoder to encode retouch-specific transformations, while the decoder learns to apply these transformations faithfully to the original image. Once trained, our method can encode the user's preference pairs $\mathcal{P} = \{(x_i, y_i)\}_{i=1}^{K}$, into retouching style latents, precisely capturing the user's preferences.

3.2 Retrieval-Augmented Retouching

After representing a user's preferences through the retouching style latents, the next step is to apply it to a new query image. StarEnhancer [37] and PieNet [20] average per-pair style embeddings to create a single user preference vector, conditioning retouching on this global code. However, this ignores the content-dependent nature of image retouching, as users apply different styles to images based on their content (e.g., overexposed vs. underexposed scenes). MSM [24] models content-dependent retouching using a transformer that predicts the query style from the reference content/style embeddings and the query content embedding, and is trained through a meta-learning framework. However, this frame-

work relies on unpaired user images, where pseudo input–output pairs are constructed by synthetically degrading the unpaired images. As a result, the target outputs often exhibit relatively homogeneous color and tone distributions, while variation is primarily introduced only on the input side. This constrains the learned mapping space, encouraging the model to reproduce the output distribution of the training images rather than learning a robust content-aware style prediction capability.

To address these limitations, we propose **retrieval-augmented retouching (RAR)**, which applies retouching styles by retrieving reference examples that are most relevant to a given query image, as illustrated on the right side of Figure 3. Given a query image x_{q}, we first extract its content embedding c_{q}, together with the content embeddings $\{c_i\}$ of the input images in the reference. Specifically, we use the LoRA-adapted SigLIPv2 backbone from the style encoder and take its pooled features, which are shown to capture retouching-relevant content: color and tone cues, as detailed in Section 4.4. For each reference pair, we also encode its retouching style latent z_i using the style encoder, representing the user's retouching preference. We then compute the relevance between the query image and each reference input by measuring the cosine similarity between their content embeddings: $s_i = \frac{\mathbf{c}_{\mathrm{q}} \cdot \mathbf{c}_i}{\|\mathbf{c}_{\mathrm{q}}\| \|\mathbf{c}_i\|}$. Based on these similarity scores, we retrieve the top-K nearest reference examples, denoted as $\mathcal{N}_K$, and aggregate their corresponding retouching style latents through a softmax-weighted combination:

$$w_i = \frac{\exp\left(\frac{s_i}{\tau}\right)}{\sum_{j \in \mathcal{N}_K} \exp\left(\frac{s_j}{\tau}\right)}, \qquad z_{\mathrm{q}} = \sum_{i \in \mathcal{N}_K} w_i \, z_i, \tag{2}$$

where $\tau > 0$ is the temperature controlling the sharpness of the weighting. The personalized retouching result is then obtained by the conditional decoder as,

$$\hat{y}_{\mathrm{q}} = D_\phi(x_{\mathrm{q}}, z_{\mathrm{q}}), \tag{3}$$

applying the aggregated style z_{q} to the query image. By retrieving and weighting styles based on content similarity, RAR adapts user preferences to each image's characteristics rather than enforcing a single global style for every input image.

4 Experiments

4.1 Dataset

To train the asymmetric auto-encoder in our **RefRetouch**, we collect 800 free presets from the Adobe Lightroom community [1] and apply them to 95,000 images sampled from the LAION dataset [33]. This process provides a diverse set of real-world retouching transformations while maintaining a clear separation between training and evaluation domains. To ensure fair benchmarking, we exclude all evaluation benchmark images and presets from the training set, in contrast to previous methods that incorporated benchmark content or styles during training [20, 37]. Additional details on the dataset construction, filtering, and preset selection are provided in the supplementary material.

4.2 Experimental Setup

We use SigLIPv2 as the backbone of the Siamese style encoder and add LoRA adapters (rank 16) to all linear layers in the attention blocks. The projection layer takes the concatenation of the features pooled from the two branches and outputs a 2048-dimensional retouching style latent. The conditional MLP decoder operates per pixel in color space. It consists of an input layer that maps an RGB pixel to a 128-dimensional hidden feature, followed by three conditional MLP blocks with hidden dimensions [256, 512, 3], as shown in Figure 3. All intermediate layers use ReLU activations, while the final layer uses a sigmoid to constrain outputs to $[0, 1]$. The retouching style latent is injected into all hidden layers of the MLP decoder after projecting it using a Linear+ReLU to match the target layer's hidden dimension. This model is then trained for $180,000$ steps with a batch size of 8 on 384×384 random crops of the training images. For content embeddings in retrieval-augmented retouching (RAR), we reuse the LoRA-adapted SigLIPv2 backbone and extract its pooled features. Then, we compute cosine similarity between the query and references' input embeddings to retrieve the most relevant references using Equation (2). Unless stated otherwise, we set the softmax temperature to $\tau = 0.1$, and $K = 3$ in all the experiments.

4.3 Benchmarks

We evaluate image retouching personalization across three increasingly challenging settings: (1) single-style retouching, (2) multi-style retouching with consistent edits, and (3) multi-style retouching with inconsistent, implicitly defined edits.

VCIRB (Single-Style Retouching) To evaluate personalization under a single, consistent retouching style, we construct the Visually Consistent Image Retouching Benchmark (VCIRB) by applying 25 unseen Adobe Lightroom presets to images sampled from LAION [33]. Since a preset may produce inconsistent results across diverse content [11,19], we cluster images into 25 semantically and color-tone coherent groups using SigLIPv2 embeddings and *ab*-channel histograms in CIELAB space. Presets are applied within these clusters to ensure visual consistency in the retouched outputs. We compare against several generic personalized image retouching baselines: StarEnhancer [37], MSM [24], VisualCloze [27] (a generic in-context generative editing model), PhotoArtAgent [4], Seedream4.0 [34], and NanoBanana [13]. PhotoArtAgent, Seedream 4.0, and NanoBanana are evaluated only on single-pair personalization, as they are constrained by a short effective context window, where using more than two images as input leads to degraded performance. We also include retrained variants of MSM and VisualCloze, trained on our meta-learning dataset in which 800 Lightroom presets are applied to 95,000 LAION images clustered by semantic content and tonal similarity, following the same protocol as VCIRB (see supplementary material for dataset construction details).

PPR10K-Groups (Multi-Style, Consistent Retouching) In real-world scenarios such as event or portrait photography, users often retouch related images into a consistent output style. The PPR10K dataset [28] groups portrait

images of the same subjects captured in different scenes, along with their expert-retouched versions, where experts are instructed to apply consistent adjustments across all images within a group. This inherent style consistency allows a few reference samples from a group to effectively capture the user's retouching preferences, while the variation in scene content provides a natural testbed for evaluating a model's context-awareness. We select 21 PPR10K groups, retouched by expert A, with more than 12 images each and sample 1, 3, or 6 reference pairs per group and test on the remaining images. We compare with the same baselines as in VCIRB, testing each model's ability to handle multiple references in a multi-style setting.

MIT-FiveK and PPR10K Users (Multi-Style, Inconsistent Retouching) Another common use case is adapting to a user's retouching style from historical edits, which is often diverse and inconsistent. This setting is widely used in prior personalization works [24, 31, 45]. We follow the standard protocol on MIT-Adobe FiveK [3] and PPR10K [28] to split the datasets into training and test splits. From each training split, we sample 20, 50, and 100 input–output reference pairs and evaluate on the test sets (498 for FiveK and 2,286 for PPR10K). Unlike prior work [24], we explicitly sample references based on color and tone diversity to minimize distribution mismatch between reference and test images. These fixed references are then used for all methods. Further details of the sampling strategy are provided in the supplementary material. To compare our method, we select baselines capable of handling more than 20 references. While VisualCloze can theoretically accommodate any number of references, its computational complexity increases rapidly as the number of references grows, since it directly processes the patchified 2D latent encodings of the reference pairs using a large diffusion transformer. Therefore, we exclude it from comparison in this setting. Additionally, we include two state-of-the-art, training-based image personalization methods: AdaInt [45] and RSFNet [31] to assess and compare their performance when training data is limited. Both are trained on the reference images for 100 epochs using their default configurations. Note that we exclude StarEnhancer from the evaluation on the MIT-FiveK data because the model was already trained on MIT-FiveK.

All images in VCIRB are resized to 512×512, while the original resolution is maintained for images from MIT-FiveK and PPR10K. For VisualCloze, we pre-resize the images to be divisible by 16 using a right crop, and evaluate them at this resolution to prevent loss due to VAE downsampling and patchification. PSNR, SSIM, and LPIPS are used to evaluate image retouching personalization performance across all three benchmarks. The reported metrics are averaged over all test images and across groups (if applicable).

4.4 Results

Reconstruction As shown in Table 1, we compare the reconstruction fidelity of RefRetouch with MSM [24], an approach that learns a retouching style latent through a reconstruction objective. We evaluate both the original MSM checkpoint and another version trained on the same dataset as ours for a fair com-

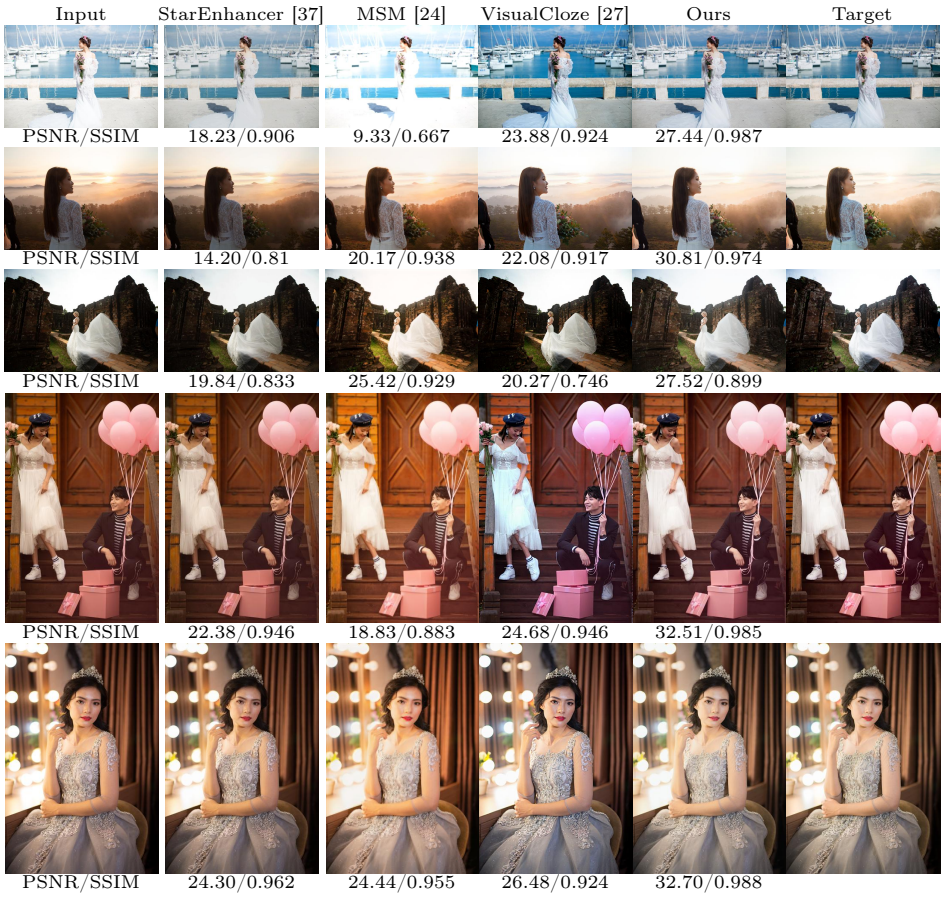

Fig. 4: PPR10K groups performance comparison when three references are used.

parison. Across the VCIRB benchmark and paired examples from MIT-Adobe FiveK [3] and PPR10K [28], RefRetouch consistently outperforms both MSM variants. These results demonstrate that our model (i) encodes retouching styles effectively into a low-dimensional latent vector and (ii) faithfully reconstructs the retouched output from the original image and the learned latent representation.

Single-reference retouching personalization Reconstruction alone is not sufficient; the learned retouching style latent must be disentangled from image content to enable style transfer to new query images, a key requirement for personalized retouching. To evaluate this, we extract the retouching style latent from a single reference pair and apply it to unseen query images. As shown in Figure 2, RefRetouch accurately transfers the retouching style from the reference to the query images using only the latent representation. These results indicate that the encoded latent representation is effectively disentangled from image content, enabling it to capture the retouching style in the reference pair while remaining invariant to scene semantics. Quantitative results on the VCIRB benchmark in Table 3 further support this finding; our method significantly outperforms

Table 1: Reconstruction fidelity comparison. † indicates that the baseline model is trained on our dataset.

Dataset	Method	PSNR↑	SSIM↑	LPIPS↓
VCIRB	MSM [24]	27.28	0.900	0.099
	MSM† [24]	29.61	0.923	0.069
	Ours	**32.42**	**0.974**	**0.035**
MIT-FiveK	MSM [24]	28.94	0.851	0.137
	MSM† [24]	26.32	0.803	0.199
	Ours	**32.95**	**0.972**	**0.028**
PPR10K-A	MSM [24]	29.28	0.943	0.075
	MSM† [24]	29.71	0.936	0.086
	Ours	**34.14**	**0.984**	**0.019**

Table 2: Photorealistic style transfer quantitative results. Metrics follow [19]. Higher Content and Style Scores indicate better structure preservation and style fidelity, respectively.

Method	Content Score↑	Style Score↑
NeuralPreset [19]	0.856	0.555
DeepPreset [15]	0.913	0.616
D-LUT [25]	0.787	0.657
PhotoArtAgent [4]	0.860	0.530
PhotoWCT2 [7]	0.749	0.916
NanoBanana [13]	0.658	0.610
Seedream4.0 [34]	0.454	0.461
Ours	0.742	0.917

existing generic retouching transfer approaches. For evaluations with multiple reference pairs, we use retrieval-augmented retouching to aggregate style cues from relevant references. However, increasing the number of references results in only marginal improvements on VCIRB, which is expected since all images in a given VCIRB group are retouched with the same preset, making a single reference sufficient to capture the style.

Multi-reference personalization with consistent retouching styles Table 3 provides the performance on the PPR10K-groups dataset, and similarly, our method consistently outperforms existing generic retouching personalization approaches. Notably, performance improves as more reference examples are added, in contrast to the high inconsistency seen in other methods. This highlights the effectiveness of our retrieval-augmented retouching in leveraging multiple references for content-aware retouching, adapting the style based on the input image's context. Qualitative comparisons using three reference examples are presented in Figure 4.

Personalization with diverse and inconsistent retouching styles The results in Table 4 show the performance of personalizing to MIT-FiveK and PPR10K users' retouching style. Our approach surpasses all generic personalized retouching baselines and achieves performance comparable to, or even exceeding, methods explicitly trained on the specific individual user references [31, 45], despite being entirely tuning-free. This highlights the strong personalization capability of our approach and its ability to generalize to challenging image retouching personalization scenarios. Consistent with the PPR10K-groups evaluation, performance improves with additional reference images, highlighting the model's ability to leverage multiple examples, especially in cases involving diverse retouching styles.

Photorealistic style transfer Our method naturally supports photorealistic style transfer, in which the color and tonal characteristics of an unpaired style reference are applied to a content image while preserving its structure and semantics [15, 19, 46]. This is achieved by forming a pseudo reference pair, reusing the content image as the input and the style image as the target, and then applying our standard pipeline without any task-specific tuning. The visual results in Figure 5 and the quantitative results in Table 2, evaluated using

Table 3: Image retouching personalization comparison on the VCIRB and PPR10K groups benchmarks. The best results are shown in bold. † indicates methods that are re-trained (or fine-tuned) on our dataset.

Dataset	Method	PSNR↑			SSIM↑			LPIPS↓		
		1	**2**	**4**	**1**	**2**	**4**	**1**	**2**	**4**
VCIRB	StarEnhancer [37]	21.00	20.98	21.27	0.864	0.863	0.868	0.123	0.123	0.120
	MSM [24]	17.69	18.17	18.53	0.822	0.830	0.836	0.164	0.155	0.150
	MSM† [24]	24.35	25.03	25.28	0.900	0.904	0.905	0.097	0.094	0.093
	VisualCloze [27]	19.81	18.60	12.05	0.765	0.731	0.378	0.173	0.191	0.591
	VisualCloze† [27]	23.91	24.63	24.70	0.83	0.834	0.825	0.077	0.072	0.079
	PhotoArtAgent [4]	20.03	-	-	0.83	-	-	0.158	-	-
	NanoBanana [13]	11.61	-	-	0.387	-	-	0.570	-	-
	Seedream4.0 [34]	11.8	-	-	0.422	-	-	0.488	-	-
	Ours	**29.13**	**29.57**	**29.82**	**0.963**	**0.964**	**0.965**	**0.048**	**0.047**	**0.046**
		1	**3**	**6**	**1**	**3**	**6**	**1**	**3**	**6**
PPR10K-groups	StarEnhancer [37]	19.16	18.83	19.36	0.875	0.857	0.866	0.102	0.108	0.119
	MSM [24]	18.86	18.82	18.59	0.877	0.873	0.878	0.154	0.159	0.155
	MSM† [24]	19.27	19.62	18.16	0.876	0.880	0.866	0.149	0.148	0.170
	VisualCloze [27]	15.04	14.21	7.74	0.453	0.432	0.246	0.22	0.261	0.844
	VisualCloze† [27]	21.12	21.42	20.98	0.832	0.832	0.812	0.097	0.093	0.112
	PhotoArtAgent [4]	20.01	-	-	0.888	-	-	0.093	-	-
	NanoBanana [13]	12.89	-	-	0.48	-	-	0.434	-	-
	Seedream4.0 [34]	9.34	-	-	0.278	-	-	0.607	-	-
	Ours	**22.51**	**23.92**	**25.27**	**0.932**	**0.949**	**0.961**	**0.065**	**0.052**	**0.043**

metrics from [19], demonstrate that our method achieves superior style transfer performance while faithfully preserving content compared to state-of-the-art Photorealistic style transfer methods [7, 15, 19, 25] as well as generic image editing models [13, 34]. This indicates our model's ability to encode retouching style across two semantically different images.

User study We further conduct a user study to evaluate both style transfer fidelity and the visual appeal of photorealistic style transfer, criteria that are difficult to fully capture with quantitative metrics due to the lack of ground truth. As shown in Table 8, our method achieves competitive Style Fidelity while outperforming all baselines in Visual Appeal. The study involves 40 participants. For each trial, participants are presented with the reference style image, the content image, and five corresponding stylized results, and are asked to rank the outputs. The evaluated samples are randomly drawn from a pool of 40 examples to increase coverage and reduce sampling bias. Overall, the results indicate that our approach produces perceptually stronger photorealistic stylization, despite not being explicitly trained for the task.

Retouching-aware feature adaptation We analyze the feature embeddings of the LoRA-adapted SigLIPv2 backbone after training with the asymmetric framework and observe a shift from general semantic representations toward retouching-relevant color and tone representations. In particular, as shown in Figure 6, top-4 retrieval using the adapted SigLIPv2 features returns images with similar color and tone characteristics, whereas retrieval using the frozen pretrained SigLIPv2 features is primarily driven by semantic similarity (details of the retrieval implementation are provided in the supplementary material). This observation motivates our use of the adapted SigLIPv2 backbone as the content

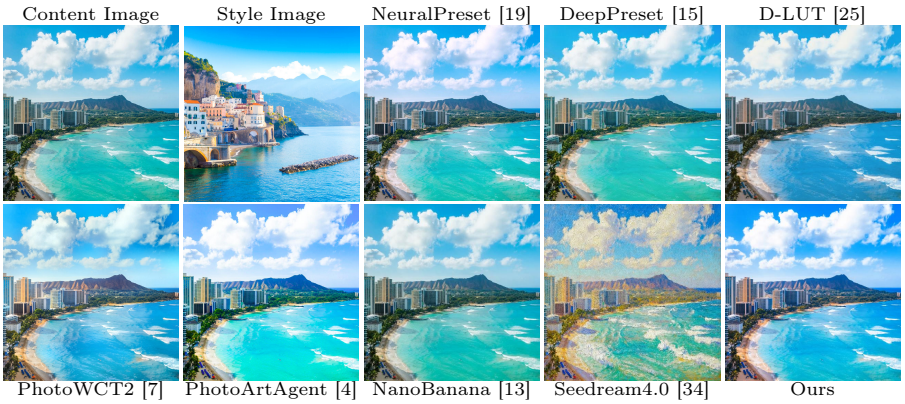

Fig. 5: Photorealistic style transfer comparison. Our method can be used for photorealistic style transfer, applying color and tone from a single unpaired reference out of the box.

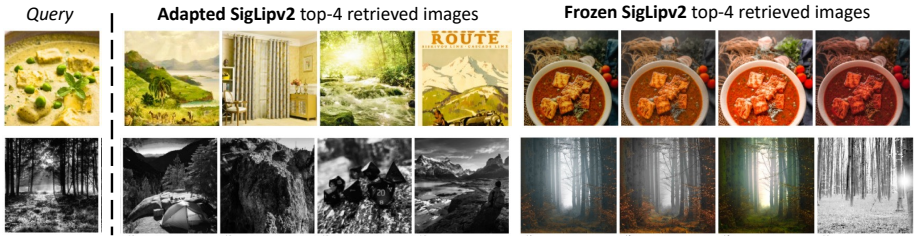

Fig. 6: Feature embedding analysis through top-K retrieval.

encoder for multi-reference personalized retouching, since experts typically apply similar retouching styles to images with similar color and tone distributions rather than to images that are merely semantically similar.

4.5 Ablation Study

To validate the architectural design and evaluate the contribution of individual components in the asymmetric auto-encoder of *RefRetouch*, we conduct ablation studies varying the encoder backbone, decoder architecture, and conditioning method, as summarized in Table 5. All models are trained for 10,000 iterations on a subset of the training data with a batch size of 8. Reconstruction fidelity is evaluated on a held-out validation set using PSNR and SSIM metrics.

Encoder backbone In experiments 1–3, we observe that using a SigLIPv2 encoder with trainable LoRA adapters significantly outperforms both the ResNet-50 and frozen SigLIPv2 variants. This highlights that strong pretrained features alone are insufficient; adaptation to the retouching task is essential for effective retouching style extraction.

MLP decoder Comparing Experiments 1 and 4, we find that the proposed lightweight color-space MLP decoder achieves better reconstruction performance than the UNet-based decoder used in [24], highlighting the effectiveness of our

Table 4: Image personalization with few reference samples on the standard MIT-FiveK and PPR10K benchmark datasets. While MSM, StarEnhancer, and our method does not require user-specific training, AdaInt and RSFNet require training on individual user's reference samples.

Dataset	Method	PSNR↑			SSIM↑			LPIPS↓		
		20	50	100	20	50	100	20	50	100
MIT-FiveK [3]	MSM [24]	22.13	21.79	22.05	0.807	0.806	0.807	0.171	0.174	0.171
	MSM† [24]	20.46	20.48	20.58	0.750	0.753	0.753	0.238	0.236	0.234
	AdaInt [45]	20.12	20.80	22.49	0.824	0.833	0.854	0.111	0.098	0.078
	RSFNet [31]	22.52	22.87	22.66	0.862	0.866	0.86	0.072	0.069	0.072
	Ours	**22.95**	**23.19**	**23.34**	**0.913**	**0.917**	**0.920**	**0.070**	**0.067**	**0.067**
PPR10K [28]	StarEnhancer [37]	20.54	20.57	20.64	0.880	0.878	0.882	0.085	0.086	0.085
	MSM [24]	19.48	19.98	20.06	0.866	0.873	0.873	0.146	0.138	0.134
	MSM† [24]	20.94	21.21	21.38	0.871	0.875	0.878	0.117	0.117	0.116
	AdaInt [45]	21.28	22.08	**22.82**	0.903	0.921	0.932	0.077	**0.065**	**0.058**
	RSFNet [31]	**22.01**	22.08	22.22	0.923	0.924	0.923	**0.069**	0.067	0.065
	Ours	21.75	**22.30**	22.47	**0.935**	**0.941**	**0.942**	0.070	**0.065**	0.063

Table 5: Ablation study for the **asymmetric auto-encoder** in RefRetouch. We compare three encoder backbones: *RN-50* (ResNet-50), *SL-F* (frozen SigLIPv2), and *SL-LoRA* (SigLIPv2 with LoRA adapters). Decoder variants include our proposed conditional *MLP* and the *UNet*-based decoder from [24]. We also evaluate three conditioning strategies: additive (*Add*), adaptive layer normalization (*AdaLN*), and cross-attention (*Cross-attn*).

Exp №	Encoder			Decoder		Cond-Method			Metrics	
	RN-50	*SL-F*	*SL-LoRA*	*MLP*	*UNet*	*Add*	*AdaLN*	*Cross-attn*	PSNR↑	SSIM↑
1	×	×	✓	✓	×	✓	×	×	**32.40**	**0.977**
2	✓	×	×	✓	×	✓	×	×	30.00	0.971
3	×	✓	×	✓	×	✓	×	×	23.41	0.931
4	×	×	✓	×	✓	×	×	×	29.58	0.944
5	×	×	✓	✓	×	×	✓	×	31.93	0.977
6	×	×	✓	✓	×	×	×	✓	22.66	0.880

asymmetric design. This result supports our hypothesis that a simple per-pixel decoder forces the encoder to capture content-disentangled, retouching-relevant features, enabling faithful reconstruction of the enhanced output from the input.

Conditioning method Experiments 1, 5, and 6 further demonstrate that simple additive conditioning injection outperforms adaptive layer norm and cross-attention alternatives. For conditioning the UNet decoder with the retouching style latent, we adopt the approach in [24].

Retrieval-augmented retouching Table 6 shows the image personalization performance on MIT-FiveK, varying the number of relevant references used to compute the retouching style, as in Equation (2). Performance improves with more relevant references, demonstrating the model's ability to utilize multiple inputs. However, using all the references results in poor performance, highlighting the importance of selecting relevant samples and emphasizing the effectiveness of our retrieval-augmented retouching approach.

Content embedding model We compare different content embedding models for representing the query and reference images in the RAR module. The

Table 6: Image personalization with varying top-K, number of relevant references.

Ref	k=1	k=3	k=5	All
20	21.44/0.889	22.95/0.913	23.37/0.918	21.99/0.897
50	21.47/0.899	23.19/0.917	23.56/0.920	21.97/0.898
100	21.94/0.900	23.34/0.920	23.82/0.923	21.94/0.897

Table 7: Comparison of different content embedding models used for reference retrieval in the proposed RAR module. Results are reported in PSNR (dB).

Encoder	PPR10K-Groups			MIT-FiveK		
	1	3	6	20	50	100
SigLIPv2 Adapted	22.51	23.92	25.27	22.95	23.19	23.33
SigLIPv2 Frozen	22.51	23.49	23.75	21.02	21.38	20.99
Lab Histogram	22.51	23.94	24.66	21.94	21.92	21.86
DINOv2	22.51	23.53	24.49	21.11	20.63	20.71
LUT Decoder	22.51	23.92	25.28	22.97	23.21	23.34

Table 8: User study results on photorealistic style transfer. T1 and T2 indicates the percentage of Top-1 and Top-2 positions.

Method	Style Fid.		Vis. Appeal	
	T1(%)	T2(%)	T1(%)	T2(%)
D-LUT	9.8	23.5	13.7	49.0
DeepPreset	15.7	29.4	37.3	66.7
NanoBanana	7.8	17.6	5.9	29.4
PhotoWCT2	41.2	66.7	15.7	35.3
Ours	39.2	80.4	41.2	56.9

results in Table 7 provide the retouching performance on the multi-reference PPR10K-Groups and MIT-FiveK benchmarks. Overall, the LoRA-adapted SigLIPv2 backbone consistently achieves the best performance, outperforming generic semantic encoders such as frozen SigLIPv2 and DINOv2, while also surpassing the hand-crafted Lab histogram descriptor. These results suggest that the asymmetric auto-encoding framework adapts the representation toward retouching-relevant photometric attributes, enabling more effective retrieval of reference images with similar color and tone characteristics.

Limitations Our method currently performs only global retouching, applying a uniform transformation across the entire image. As a result, it cannot selectively adjust specific regions or handle spatially varying edits. In future work, we aim to extend the framework to support region-aware, locally adaptive retouching by incorporating spatial decomposition or region-level conditioning following the approaches in [29, 48].

5 Conclusion

We proposed a generic personalized image retouching framework that adapts to individual user's image retouching style from only a handful of reference pairs without test-time fine-tuning. Experiments across three benchmarks of increasing complexity show that our proposed method consistently outperforms existing generic image retouching personalization methods. In addition, the same pipeline can be used out of the box for photorealistic style transfer, highlighting its versatility and generalization across tasks.

Acknowledgements

This research was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LD24F020007.

References

1. Adobe Inc.: Adobe lightroom community presets (2025), https://lightroom.adobe.com/learn/discover, accessed 2025-05-11
2. Bar, A., Gandelsman, Y., Darrell, T., Globerson, A., Efros, A.: Visual prompting via image inpainting. Advances in Neural Information Processing Systems **35**, 25005–25017 (2022)
3. Bychkovsky, V., Paris, S., Chan, E., Durand, F.: Learning photographic global tonal adjustment with a database of input / output image pairs. In: The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition (2011)
4. Chen, H., Tao, K., Wang, Y., Wang, X., Zhu, L., Gu, J.: Photoartagent: Intelligent photo retouching with language model-based artist agents. arXiv preprint arXiv:2505.23130 (2025)
5. Chen, L., Mao, Q., Gu, Y., Shou, M.Z.: Edit transfer: Learning image editing via vision in-context relations. arXiv preprint arXiv:2503.13327 (2025)
6. Chen, Y.S., Wang, Y.C., Kao, M.H., Chuang, Y.Y.: Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6306–6314 (2018)
7. Chiu, T.Y., Gurari, D.: Photowct2: Compact autoencoder for photorealistic style transfer resulting from blockwise training and skip connections of high-frequency residuals. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 2868–2877 (2022)
8. Conde, M.V., Vazquez-Corral, J., Brown, M.S., Timofte, R.: Nilut: Conditional neural implicit 3d lookup tables for image enhancement. In: Proceedings of the AAAI Conference on Artificial Intelligence (2024)
9. Duan, Z.P., Zhang, J., Lin, Z., Jin, X., Wang, X., Zou, D., Guo, C.L., Li, C.: Diffretouch: Using diffusion to retouch on the shoulder of experts. In: Proceedings of the AAAI Conference on Artificial Intelligence (2025)
10. Dutt, N.S., Ceylan, D., Mitra, N.J.: MonetGPT: Solving puzzles enhances mllms' image retouching skills. arXiv preprint arXiv:2505.06176 (2025)
11. Elezabi, O., Conde, M.V., Wu, Z., Timofte, R.: Inretouch: Context aware implicit neural representation for photography retouching. arXiv preprint arXiv:2412.03848 (2024)
12. Gharbi, M., Chen, J., Barron, J.T., Hasinoff, S.W., Durand, F.: Deep bilateral learning for real-time image enhancement. ACM Transactions on Graphics (TOG) **36**(4), 1–12 (2017)
13. Google AI: Nano banana. https://ai.google.dev/gemini-api/docs/image-generation (2025), accessed: 2025-11-12
14. He, J., Liu, Y., Qiao, Y., Dong, C.: Conditional sequential modulation for efficient global image retouching. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16. pp. 679–695. Springer (2020)
15. Ho, M.M., Zhou, J.: Deep preset: Blending and retouching photos with color style transfer. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2113–2121 (2021)
16. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International workshop on similarity-based pattern recognition. pp. 84–92. Springer (2015)
17. Hu, Y., He, H., Xu, C., Wang, B., Lin, S.: Exposure: A white-box photo post-processing framework. ACM Transactions on Graphics (TOG) **37**(2), 1–17 (2018)

18. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
19. Ke, Z., Liu, Y., Zhu, L., Zhao, N., Lau, R.W.: Neural preset for color style transfer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14173–14182 (2023)
20. Kim, H.U., Koh, Y.J., Kim, C.S.: Pienet: Personalized image enhancement network. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16. pp. 374–390. Springer (2020)
21. Kim, J., Hwang, S., Kim, D.O., Han, C., Park, M.K., Kim, C.S.: Oneta: Multi-style image enhancement using eigentransformation functions. arXiv preprint arXiv:2506.23547 (2025)
22. Kim, W., Cho, N.I.: Image-adaptive 3d lookup tables for real-time image enhancement with bilateral grids. In: European Conference on Computer Vision. pp. 91–108. Springer (2024)
23. Kim, W., Lee, K., Cho, N.I.: Lightweight and fast real-time image enhancement via decomposition of the spatial-aware lookup tables. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11895–11905 (2025)
24. Kosugi, S., Yamasaki, T.: Personalized image enhancement featuring masked style modeling. IEEE Transactions on Circuits and Systems for Video Technology **34**(1), 140–152 (2024). https://doi.org/10.1109/TCSVT.2023.3285765
25. Li, M., Wang, G., Zhang, X., Liao, Q., Xiao, C.: D-lut: Photorealistic style transfer via diffusion process. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 9206–9214. IEEE (2025)
26. Li, Y., Liu, M.Y., Li, X., Yang, M.H., Kautz, J.: A closed-form solution to photorealistic image stylization. In: Proceedings of the European conference on computer vision (ECCV). pp. 453–468 (2018)
27. Li, Z.Y., Du, R., Yan, J., Zhuo, L., Li, Z., Gao, P., Ma, Z., Cheng, M.M.: Visualcloze: A universal image generation framework via visual in-context learning. arXiv preprint arXiv:2504.07960 (2025)
28. Liang, J., Zeng, H., Cui, M., Xie, X., Zhang, L.: Ppr10k: A large-scale portrait photo retouching dataset with human-region mask and group-level consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 653–661 (2021)
29. Lin, Y., Lin, Z., Lin, K., Bai, J., Pan, P., Li, C., Chen, H., Wang, Z., Ding, X., Li, W., et al.: JarvisArt: Liberating human artistic creativity via an intelligent photo retouching agent. arXiv preprint arXiv:2506.17612 (2025)
30. Moran, S., Marza, P., McDonagh, S., Parisot, S., Slabaugh, G.: Deeplpf: Deep local parametric filters for image enhancement. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12826–12835 (2020)
31. Ouyang, W., Dong, Y., Kang, X., Ren, P., Xu, X., Xie, X.: Rsfnet: A white-box image retouching approach using region-specific color filters. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12160–12169 (2023)
32. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
33. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in neural information processing systems **35**, 25278–25294 (2022)

34. Seedream Team, Chen, Y., Gao, Y., Gong, L., Guo, M., Guo, Q., Guo, Z., Hou, X., Huang, W., Huang, Y., et al.: Seedream 4.0: Toward next-generation multimodal image generation. arXiv preprint arXiv:2509.20427 (2025)
35. Serrano-Lozano, D., Herranz, L., Brown, M.S., Vazquez-Corral, J.: Namedcurves: Learned image enhancement via color naming. In: European Conference on Computer Vision. pp. 92–108. Springer (2024)
36. Shi, J., Xu, N., Xu, Y., Bui, T., Dernoncourt, F., Xu, C.: Learning by planning: Language-guided global image editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13590–13599 (2021)
37. Song, Y., Qian, H., Du, X.: Starenhancer: Learning real-time and style-aware image enhancement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4126–4135 (2021)
38. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., et al.: Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. arXiv preprint arXiv:2502.14786 (2025)
39. Tseng, E., Zhang, Y., Jebe, L., Zhang, X., Xia, Z., Fan, Y., Heide, F., Chen, J.: Neural photo-finishing. ACM Trans. Graph. **41**(6), 238–1 (2022)
40. Wang, H., Zhang, J., Liu, M., Wu, X., Zuo, W.: Learning diverse tone styles for image retouching. IEEE Transactions on Image Processing **33**, 310–321 (2023)
41. Wang, R., Zhang, Q., Fu, C.W., Shen, X., Zheng, W.S., Jia, J.: Underexposed photo enhancement using deep illumination estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6849–6857 (2019)
42. Wang, X., Wang, W., Cao, Y., Shen, C., Huang, T.: Images speak in images: A generalist painter for in-context visual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6830–6839 (2023)
43. Wang, Y., Li, X., Xu, K., He, D., Zhang, Q., Li, F., Ding, E.: Neural color operators for sequential image retouching. In: European Conference on Computer Vision. pp. 38–55. Springer (2022)
44. Wu, J., Wang, Y., Li, L., Zhang, F., Xue, T.: Goal conditioned reinforcement learning for photo finishing tuning. Advances in Neural Information Processing Systems **37**, 46294–46318 (2024)
45. Yang, C., Jin, M., Jia, X., Xu, Y., Chen, Y.: Adaint: Learning adaptive intervals for 3d lookup tables on real-time image enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17522–17531 (2022)
46. Yoo, J., Uh, Y., Chun, S., Kang, B., Ha, J.W.: Photorealistic style transfer via wavelet transforms. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9036–9045 (2019)
47. Zeng, H., Cai, J., Li, L., Cao, Z., Zhang, L.: Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(4), 2058–2073 (2020)
48. Zeng, H., Huang, J., Li, J., Xiong, Z.: Region-aware portrait retouching with sparse interactive guidance. IEEE Transactions on Multimedia **26**, 127–140 (2023)