OpenCIR: Conditional Image Repainting with Open Condition Mixture

Shuchen Weng†, Xiaocheng Gong†, Haojie Zheng†
Xinlong Wang, Si Li‡, and Boxin Shi, *Senior Member, IEEE*

***Abstract*—In this paper, we introduce OpenCIR, a fully-functional Conditional Image Repainting (CIR) model designed for local image editing. Given an image and a combination of conditions related to geometry, texture, and color, CIR models are required to repaint instances and seamlessly composite them with the original images. Previous CIR models suffer from limited object categories, restricted condition modalities, and demanded geometry precision. In contrast, leveraging the generative priors from pre-trained models, OpenCIR could repaint open object categories. Equipped with redesigned condition injection modules and the condition extension strategy, OpenCIR is able to understand open condition modalities. Adopting the contour refinement strategy, OpenCIR allows users to specify instances with open geometry precision. In addition, we contribute the OPEN-CIR dataset, which includes detailed annotations, tailored for the comprehensive training and evaluation of the OpenCIR model. Extensive experiments demonstrate that OpenCIR outperforms relevant state-of-the-art methods, achieving superior visual quality, and more favorable results by human evaluators.**

***Index Terms*—Image synthesis, diffusion model, cross-modality**

I. INTRODUCTION

RECENT advances in image editing techniques have significantly progressed in manipulating specific image properties, *e.g.*, color [7], [35], [77], [85], texture [16], [29], [41], [80], instance contour [53], [70], [78], [83], and overall scene composition [17], [40], [59], [73]. To capture users' editing requests more accurately, there is growing interest in guiding image editing through a combination of multiple cross-modality conditions [28], [52], [81], [84]. These breakthroughs lead to applications that lower the expertise barrier for users, *e.g.*, imitating the style of a famous painter [71].

In pursuit of stronger controllability, generative models evolve to more effectively disentangle editing objectives, determining *where* to place instances within edited images, *what* those instances are, and *how* they should appear. In this context, **Conditional Image Repainting** (CIR) emerges as an innovative task [63], [64], [74]–[76], demonstrating significant potential to address aforementioned issues. Given conditions related to geometry (where), texture (what), and color (how), CIR models repaint corresponding visual content and composite it seamlessly with the original image.

Although CIR models have achieved notable improvements in producing visually pleasing images, they still face three primary challenges: *(i)* **Limited object categories.** Current models are typically restricted by their capacity and available training data, leading to a focus on repainting specific instance categories (*e.g.*, clothes [75], birds [74], and cars [64]). *(ii)* **Restricted condition modalities.** CIR models are designed to interpret each type of condition in a single modality (*e.g.*, text for the color condition [76]). This limits the user's ability to express complex and nuanced objectives (*e.g.*, color variations). *(iii)* **Demanded geometry precision.** Existing CIR models presuppose that edited regions are provided exactly the same as the repainted instance contour [63], requiring users to provide detailed and fine-grained annotations. This is not aligned with user-friendly design principles.

To overcome the challenges in CIR models and unlock their full potential, we introduce the **OpenCIR**, a fully-functional conditional image repainting framework, tailored for effectively decoupling the color and texture properties of the editing goals. As a great variety of application scenarios shown in Fig. 1, the "open" is reflected in three aspects: *(i)* **Open object categories.** Unlike previous CIR models [63], [75], [76], OpenCIR does not rely on category labels of the geometry condition. Utilizing the generative priors from the pre-trained generative model [57], OpenCIR demonstrates a generalization capability to repaint various instances (*e.g.*, abundant repainted categories in Fig. 1). *(ii)* **Open conditional modalities.** We develop attention-based and spatially-adaptive injection modules for color and texture conditions, respectively. Equipped with the condition extension strategy, OpenCIR supports diverse color (*e.g.*, patches and histogram) and texture (*e.g.*, distinct drawing styles) conditions, allowing users to selectively manipulate a specific property of the instance in Fig. 1. *(iii)* **Open geometry precision.** To handle the feasible boundary deviation, OpenCIR incorporates the contour refinement strategy, utilizing user-provided conditions to adaptively estimate appropriate instance contour. This strategy filters out unnecessary repainted regions (*e.g.*, deviated boundary around the fingers and balloon) and produces harmonized repainted results in Fig. 1.

For training and evaluating our model, we further contribute the OPEN-CIR dataset, which consists of 5M training samples.

S. Weng, H. Zheng, and X. Wang are with Beijing Academy of Artificial Intelligence, Beijing 100083, China. Email: {scweng, wangxinlong}@baai.ac.cn.

H. Zheng is also with School of Software and Microelectronics, Peking University, Beijing 100871. Email: suimu@stu.pku.edu.cn.

X. Gong and S. Li are with School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China. Email: {xiaochengkung, lisi}@bupt.edu.cn.

B. Shi is with State Key Laboratory of Multimedia Information Processing and National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, Beijing 100871, China. Email: shiboxin@pku.edu.cn.

† Equal contribution. ‡ Corresponding author.

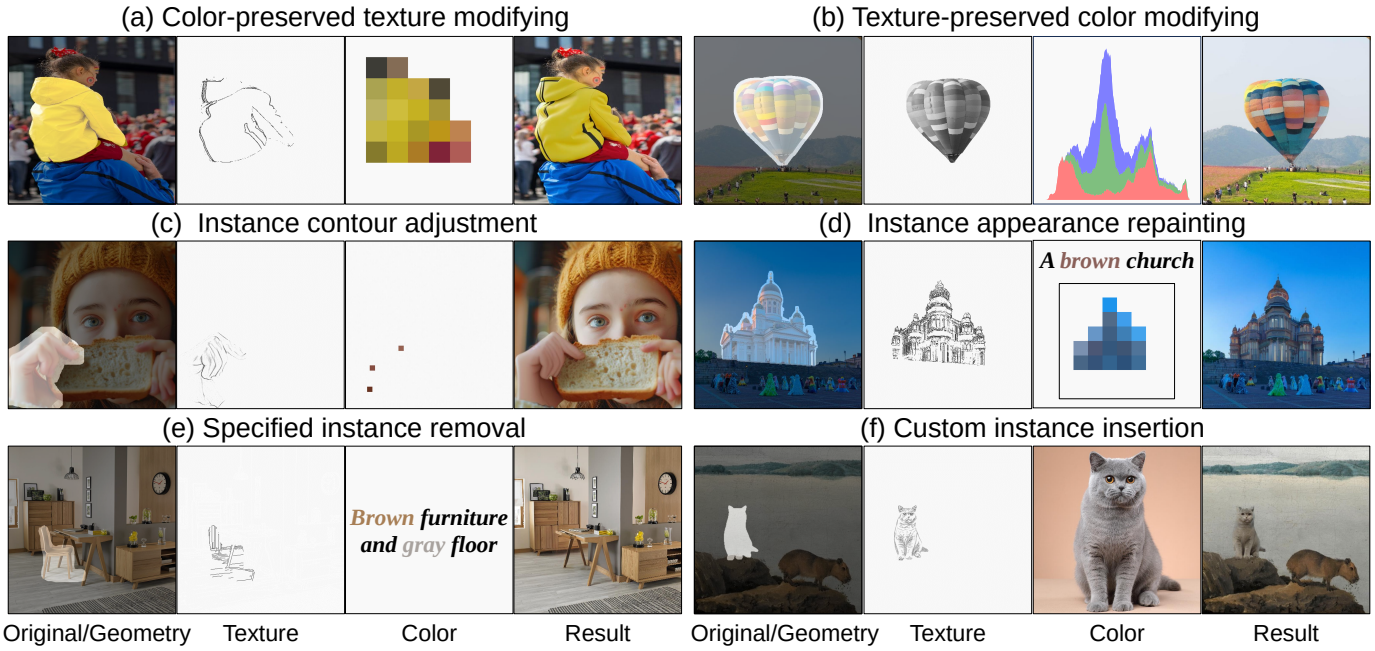

Fig. 1. Illustration of the fully-functional capabilities of the OpenCIR model. Given an image and a combination of conditions with open modalities and geometry precision, OpenCIR repaints instances across open object categories and composites them seamlessly with original images. Typical application scenarios include: (a)/(b) Preserving the color/texture property while modifying the texture/color condition for the specific instance. (c)/(d) Editing the contour/appearance of the instance by modifying both color and texture conditions. (e)/(f) Removing/inserting specified instances in/into the original image with user-provided conditions.

Each sample includes instance masks with varying precision as the geometry condition, along with comprehensive annotations of color and texture in various modalities. We summarize our contributions as follows:

- We present a comprehensive dataset covering diverse scenarios and propose a fully-functional CIR model capable of repainting open object categories.
- We redesign condition injection modules with "open" principles, ensuring the CIR model's compatibility across diverse and mixed modalities.
- We present condition extension and contour refinement strategies, enabling the model to decouple properties and support open modalities and precision of conditions.

II. Related Works

A. Low-level image property editing

Recent advancements in low-level image editing works have significantly improved the ability to modify specific properties of an image, *e.g.*, image colorization [8], [61] adds colors to old photos, texture transfer [12], [33] changes the style of images, and editable lighting [11], [65] adjusts the lighting in the scene. Additionally, considerable efforts have been made to restore lost details in images, *e.g.*, image super-resolution [13], [46], image denoising [43], [67], and image enhancement [15], [49], as well as removing specific elements from images, *e.g.*, reflection removal [26], [86], image demoiréing [20], [47], and image dehazing [21], [44]. Since these methods focus on modifying specific image properties, they fall short in further editing high-level scene semantics, *e.g.*, object categories.

B. High-level image semantic editing

High-level image editing methods explore various strategies to modify object categories [36], [54], composite objects from different images [9], [66], and remove objects entirely [45], [82]. As generative models evolve from VAEs [23], [38] to GANs [19], [51], INNs [1], [2], and diffusion models [24], [60], the conditions required from users have become increasingly user-friendly. In the context of the image-to-image translation task, Pix2Pix [32], [69] relies on paired data for training. Subsequently, CycleGAN [88] showcases the ability to train models using unpaired data. Furthermore, FlexIT [10] takes a further leap by incorporating open-vocabulary text descriptions to specify the target. Recent diffusion models have shown remarkable progress in image editing with multiple conditions. RePaint [48] employs the pre-trained DDPM [24] as the generative prior and uses contextual regions to condition the editing process. Meanwhile, SDEdit [50] edits images by first adding noise to an image and then iteratively denoising it, guided by a model within the SDE framework. Although advanced ControlNet [84] designs modules to enable multi-modal conditioning editing, the effective decoupling of fine-grained editing objectives (*e.g.*, independent control over texture, color, and placement) remains less explored.

C. Conditional image repainting

Conditional Image Repainting (CIR) is fundamentally a high-level semantic image editing task that decouples low-level image properties, *e.g.*, color, texture, geometry, and scene. CIR models [76] are proposed to "free" the users from requiring professional skills while maintaining the "freedom"

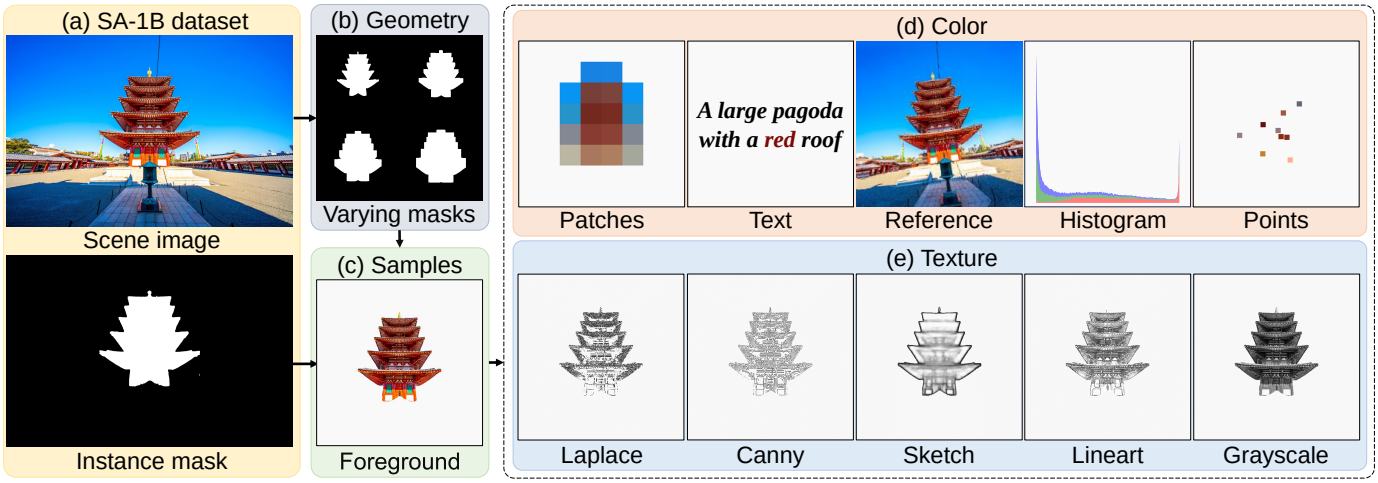

Fig. 2. Illustration of data annotation of the OPEN-CIR dataset.

to realize their ideas for editing an image. The earlier attempt starts from synthesizing realistic person images [75], which synthesizes the foreground person before composites it with the background for harmonization. To support more flexible color descriptions, Weng *et al*. [74] develop the semantic-bridge attention mechanism, enabling CIR models to interpret text descriptions. To break the two-stage dependency in [74], [75], UniCoRN [63] introduces a unified architecture that achieves more visually pleasing results. In particular, LuminAIRe [64] enhances the illumination consistency by introducing physically-correct illumination modeling. However, current CIR models are only capable of repainting limited object categories, interpreting restricted condition modalities, and adopting demanded geometry precision. These limitations inspire us to develop the fully-functional OpenCIR.

III. DATASET

We present the OPEN-CIR dataset, tailored for the comprehensive training and evaluation of CIR models. This dataset consists of 5M training samples and 1K testing samples, significantly enriching the diversity of object categories.

Data collection. We begin creating our dataset by selecting high-resolution images and their corresponding instance masks from the SA-1B dataset [39] (Fig. 2 (a)). We crop each salient instance guided by its mask, adding a randomly determined margin for variability. Instances that are too small are excluded to ensure each annotated instance has a resolution of at least 512×512. To further create binary instance masks with varying precision, we then apply multiple Gaussian filters with predefined parameters to each cropped mask (Fig. 2 (b)).

Condition annotation. During the annotation phase, we use cropped foreground instances (Fig. 2 (c)) to annotate conditions across various modalities, aiming to present detailed and nuanced objectives for the CIR task. For color conditions (Fig. 2 (d)), we adopt various approaches: transforming instances into 8×8 color patches, applying various image augmentations to synthesize reference images, calculating the color histogram with 256 bins, and extracting color points through random selection. Additionally, we carefully select hyper-parameters of the BLIP-2 [42], utilizing it for generating text descriptions[1]. For texture conditions (Fig. 2 (e)), we use four distinct algorithms to create variant texture representations that potentially emulate distinct drawing styles, *i.e*., Laplace [18], Canny [5], Sketch [62], and Lineart [6]. As the complete texture condition, we further extract grayscales by converting the color space from RGB to Lab and then omitting ab channels.

Dataset structure. Consequently, each sample in the OPEN-CIR dataset includes essential components for training: cropped foreground instances and corresponding background, geometry conditions (*i.e*., binary instance masks with varying precision), multi-modal color conditions (*i.e*., color patches, text descriptions, reference images, color histograms, and color points), and diverse texture conditions (*i.e*., Laplace, Canny, Sketch, Lineart, and grayscale). OPENCIR dataset covers common instance categories, including landscapes, animals, persons, buildings, furniture, *etc*.

IV. OPENCIR MODEL

This section starts with an overview of our framework (Sec. IV-A). Following this, we provide a comprehensive illustration of condition injection modules (Sec. IV-B). Next, we present the condition extension strategy (Sec. IV-C). The section concludes with the contour refinement strategy (Sec. IV-D).

A. Overview

The framework is based on the latent diffusion model [57]. During the training phase, the model learns to predict the specific noise added at each step of the forward process. In the inference phase, it initiates with the noise and iteratively applies the denoising network in the backward process to generate images with the image decoder. We illustrate the overview of the framework in Fig. 3.

[1]To ensure the quality, we recruit 10 volunteers to evaluate 1% samples, with a pass rate over 93%.

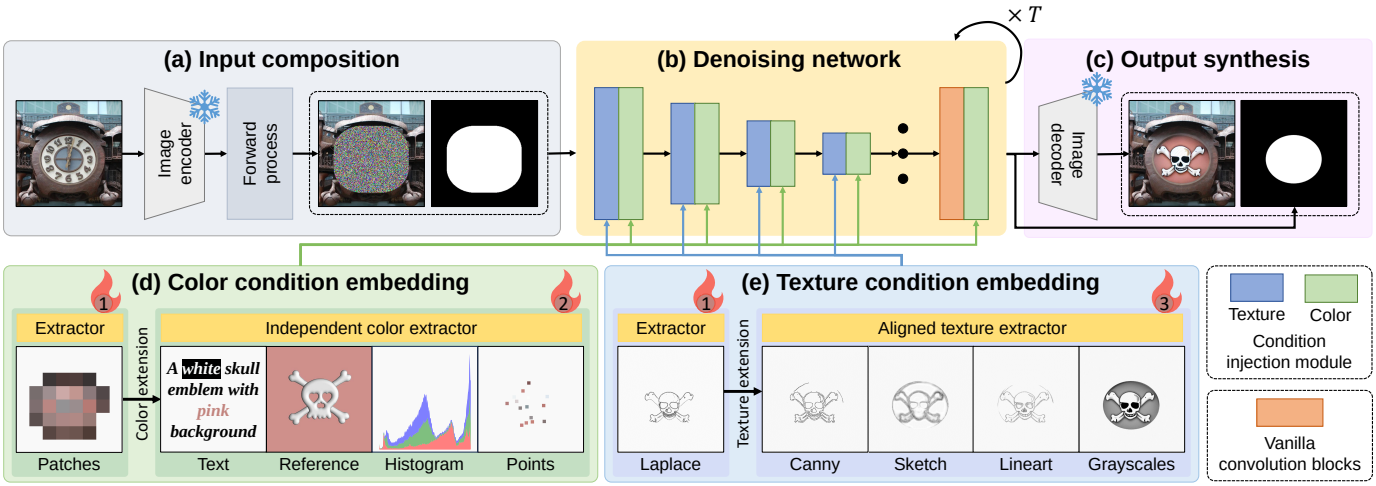

Fig. 3. The framework of the OpenCIR model. (a) The process starts with encoding the original image into the latent code. In the forward process, Gaussian noise is added selectively to repainted regions. The latent code is further concatenated with the geometry condition. (b) The input combination is fed into the denoising network equipped with texture/color injection modules. The denoising process includes T iterations. (c) With the estimated fine-grained geometry condition, the denoised latent code is mixed with background regions. Finally, an image decoder maps the mixed code into the repainted result. (d-e) By training specific parameters according to the sequence number, OpenCIR could effectively decouple the color and texture properties of the editing goals, and further support open condition modalities.

Input composition. We use a pre-trained image encoder $\mathcal{E}$ to encode the original image I^{in} into the latent code as $z_0 = \mathcal{E}(I^{\text{in}})$. After applying the forward process, we further combine the noised latent code z_t with the downsampled geometry condition $\hat{x}^{\text{g}}$ as the input of the denoising network.

Forward process. Given the special property of the forward process in diffusion models [24], [60], the noised latent code z_t could be expressed as a linear mix of the latent code z_0 and a noise variable ϵ:

$$z_t = \sqrt{\alpha_t} z_0 + \sqrt{1-\alpha_t}\epsilon, \tag{1}$$

where $\epsilon \sim \mathcal{N}(0,1)$ is Gaussian noise, and α_t is the parameter for controlling noise levels, and $t \in \{1,\dots,T\}$ is the timestep. We only add noise selectively to the latent code in repainted regions, allowing the use of the clean background and user-provided conditions to repaint the specific instance:

$$\hat{z}_t = z_t \odot \hat{x}^{\text{g}} + z_0 \odot (1 - \hat{x}^{\text{g}}), \tag{2}$$

where the downsampled geometry condition $\hat{x}^{\text{g}} \in \mathbb{R}^{\tilde{h}\times\tilde{w}}$ is a binary mask indicating *where* the repainted instance places, and $\tilde{h}$ and $\tilde{w}$ are the mask sizes.

Denoising network. Our denoising network is based on the U-Net architecture [58], equipped with texture and color injection modules. The texture injection module is used in each downsampling block, which enables the model to determine *what* to repaint. The color injection module is applied in all blocks to guide our OpenCIR in accurately presenting *how* the instance appears.

Backward process. Given a combination of conditions x^{g}, x^{t}, and x^{c}, our denoising network ϵ_θ iteratively refines the latent code $\hat{z}_t$ until it converges:

$$\hat{z}_{t-1} = \frac{1}{\sqrt{a_t}}\big(\hat{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\hat{z}_t, t, x^{\text{g}}, x^{\text{c}}, x^{\text{t}})\big) + \sigma_t\epsilon, \tag{3}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$ means the amount of noise and σ_t is the standard deviation of the noise. After T iterations, we obtain the denoised latent code $\hat{z}_0$.

Output synthesis. The output of the denoising network includes the denoised latent code $\hat{z}_0$ and the estimated fine-grained geometry condition $\bar{x}^g$. To handle open geometry precision, we precisely filter out unnecessary repainted regions by re-using the Eq. (2), where $\hat{x}^{\text{g}}$ is replaced by $\bar{x}^{\text{g}}$. After that, denoting the filtered result as $\hat{z}'_0$, we use a pre-trained image decoder $\mathcal{D}$ to map it back into the pixel space and present the synthesized image as $I^{\text{out}} = \mathcal{D}(\hat{z}'_0)$.

B. Condition injection modules

We redesign color and texture condition injection modules with "open" principles, ensuring the CIR model's compatibility across diverse and mixed modalities, as shown in Fig. 3 (b).

Texture injection module. With the OpenCIR model, users could provide texture conditions to describe *what* to repaint. To understand distinct drawing styles of texture conditions, we utilize a stack of convolution layers to extract unified spatial tensors from the texture condition. After that, to adaptively preserve the correspondence between texture semantics and repainted regions, we adopt additional convolution layers to separately obtain refined texture feature f_i^{t} corresponding to i-th downsampling block. We formulate the spatially-adaptive texture injection module as:

$$\hat{f}_i = \big(1 + \text{Conv}(f_i^{\text{t}})\big) \odot \frac{f_i - \mu_i}{\sigma_i} + \text{Conv}(f_i^{\text{t}}), \tag{4}$$

where $\hat{f}_i$ and f_i represent feature maps after and before modulating, respectively. μ_i and σ_i denote the per-channel the mean and standard deviation of f_i, and $\odot$ means the pixel-wise multiplication.

Color injection module. OpenCIR allows users to provide color conditions to define the *how* repainted instances appear.

In contrast to texture conditions that have clear correspondences, the color distribution is inherently ambiguous and requires estimation. Therefore, we adopt the ViT architecture [14] as the color extractor to encode color conditions. This consists of six stacked Transformer layers, where each layer incorporates two layer normalizations, a self-attention layer, and a feed-forward network, equipped with residual connections. The color extractor encodes color conditions into unified color tokens $x^{\mathrm{c}} \in \mathbb{R}^{N^{\mathrm{l}} \times N^{\mathrm{e}}}$, where N^{l} is the sequence length and N^{e} is the number of embedding channels. Next, we introduce the attention-based color injection module as:

$$\hat{f}_i = \mathrm{Softmax}\left(M + \frac{QK^{\top} + A}{\sqrt{N^{\mathrm{k}}}}\right) V, \tag{5}$$

Where Q and K, V are transformed features from the feature map f_i and the color condition x^{c}, respectively. N^{k} denotes the number of embedding channels. A and M are optional cues and represent pre-defined matrices: A indicates optional correspondences between color conditions and repainted regions while M means the regions targeted for color injection. Specifically, for conditions that assign colors to specific regions (*e.g.*, color patches and color points), we define $A \in \mathbb{L}^{(H \times W) \times N^{\mathrm{l}}}$, where $\mathbb{L} \in \{0, 1\}$. For conditions representing the color of specific instances (*e.g.*, color histograms), we use $M = x^{\mathrm{g}}$. For conditions that lack explicit correspondences and describe the entire image (*e.g.*, reference images and text descriptions), both M and A are all-zero matrices.

Condition mixture injection. When users provide multiple color conditions, each color condition is injected as Eq. (5). Denoting $\hat{f}^{\mathrm{c}}_{i,j}$ as the modulated feature map in the i-th block with the j-th color condition, the multi-modal color conditions mixture could be formulated as:

$$\hat{f}_i = \mathrm{Linear}\big(\frac{1}{\sum_j W_j} \sum\nolimits_j W_j \hat{f}^{\mathrm{c}}_{i,j}\big), \tag{6}$$

where W_j is the hyper-parameter to weigh the impact of each condition. If any color condition is absent, the corresponding W_j is adjusted to zero. Since texture conditions are mostly derived from user drawings, where a consistent style is generally employed for image repainting and mixing disparate texture styles offers limited additional benefit, we do not consider texture condition mixture.

C. Condition extension strategy

In addition to compatible condition injection modules, we present the corresponding training strategy to effectively decouple properties and support open condition modalities. As shown in Fig. 3 (d-e), our condition extension strategy includes following three steps.

Base model training. Considering color patches represent the mean value of each local region and Laplace indicates the changing trend of the global image, we firstly use these conditions to train a base model to decouple the color and texture properties of the editing goals. To preserve the generative priors of pre-trained models, we only train the condition injection modules and color/texture extractors, and freeze remaining parameters of the denoising network (*e.g.*, vanilla convolution blocks). For denoising, we employ the MSE loss to minimize the discrepancy between the learned and target image distributions for arbitrary denoised latent code, formulated as:

$$\mathcal{L}_{\mathrm{dm}} = \mathbb{E}_{t, z_0, \epsilon \sim \mathcal{N}(0,1)}\big[\|\epsilon - \epsilon_\theta(z_t, t, x^{\mathrm{g}}, x^{\mathrm{c}}, x^{\mathrm{t}})\|_2\big]. \tag{7}$$

Color condition extension. We further extend color condition modalities based on the well-trained base model, presented in Fig. 3 (d). Specifically, we further fix parameters of the trained texture extractor to preserve the extracted texture property, remaining the color property to be represented. Next, we introduce and train independent color extractors with the same Transformer architecture for each color condition. During extension, we drop color conditions with 10% probability, otherwise randomly select $N^{\mathrm{c}} \in \{1, 2, 3\}$ modalities as color conditions for training according to Eq. (7).

Texture condition extension. We extend texture condition modalities after color conditions, illustrated in Fig. 3 (e). Specifically, we introduce additional convolution layers to capture various texture patterns. After that, these patterns are translated into Laplace's texture space in a contrastive learning manner [34]. This alignment process fills missing textures and removes irrelevant details (*e.g.*, brightness in grayscales). Next, we fix the parameters of all independent color extractors to maintain the color representation. We train the remaining parameters for each texture condition as Eq. (7). This ensures color conditions correctly match corresponding regions under the guidance of texture conditions with distinct drawing styles.

D. Contour refinement strategy

Previous CIR models [63], [74]–[76] assume that edited regions are provided exactly the same as the repainted instance contour. Towards open geometry precision, we introduce the contour refinement strategy to the OpenCIR model. As presented in Fig. 3 (a-c), we adopt additional channels to the denoising network to integrate downsampled user-provided geometry conditions $\hat{x}^{\mathrm{g}}$ and to estimate the fine-grained instance contour $\bar{x}^{\mathrm{g}} \in \mathbb{R}^{H \times W}$. With the training process, the precision of the estimated instance contour improves based on user-provided conditions. This is because *(i)* the geometry condition, despite its open precision, provides the fundamental instance contour of the expectation; and *(ii)* the color and texture conditions inherently contribute to defining the expected instance contour. We further adopt the Dice loss to identify the fine-grained repainted regions as:

$$\mathcal{L}_{\mathrm{dc}} = 1 - 2\frac{|\bar{x}^{\mathrm{g}} \cap x^{\mathrm{g}}|}{|\bar{x}^{\mathrm{g}}| + |x^{\mathrm{g}}|}, \tag{8}$$

where $|\cdot|$ means size of the set and $\cap$ presents intersection of two sets. Therefore, OpenCIR could adaptively understand geometry conditions with open precision. The total loss is a combination:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{dm}} + \lambda \mathcal{L}_{\mathrm{dc}}, \tag{9}$$

where $\lambda = 0.01$ is a hyper-parameter.

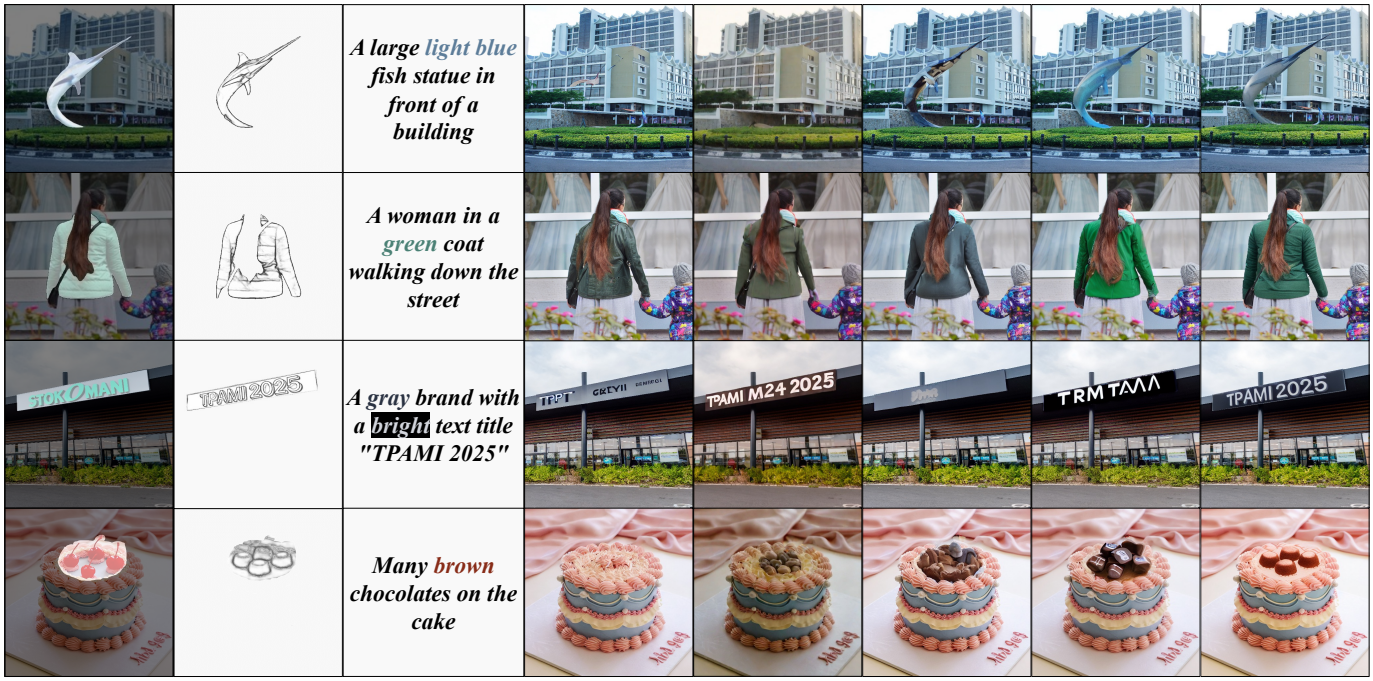

Fig. 4. Qualitative comparison with state-of-the-art methods. **Left:** Original images and conditions related to geometry (where), texture (what), and color (how). **Right:** Qualitative comparison results.

V. Experiment

A. Training details

We initialize the parameters of OpenCIR from SD1.5[2] and intend to repaint images with 512×512 resolution. The base model training, color condition extension, and texture condition extension take approximately 100, 60, and 60 hours on 4 NVIDIA Tesla V100 graphic cards. We adopt the Adam optimizer [37] with a learning rate of 5×10^{-5}. During inference, we utilize the DDIM sampling strategy [60] with 50 sampling steps.

B. Quantitative evaluation metrics

Four metrics are used to quantitatively evaluate the performance of each method: *(i)* We use the Fréchet inception distance (FID) score [22] to measure the distance between the distribution of repainted results and real images. *(ii)* We adopt Structural Similarity Index Measure (SSIM) [72] to evaluate whether repainted images have corresponding visual structure. *(iii)* We show Peak Signal-to-Noise Ratio (PSNR) [31] to measure the visual perceived quality between repainted regions and real images. *(iv)* We calculate CLIP score [56] to evaluate whether the repainted images are aligned with text descriptions in the CLIP space.

C. Comparison with state-of-the-art methods

We compare OpenCIR with relevant state-of-the-art image editing methods, including Stable Diffusion [57], SDXL [55], SDEdit [50], SmartBrush [81], InstructPix2Pix [3], ControlNet [84], UniControlNet [87], and T2I-Adapter [52]. Among these methods, we reimplement and retrain SmartBrush [81] due to its code and weights being unavailable. As InstructPix2Pix [3] requires paired training data that our dataset cannot provide, we only evaluate its pre-trained models. For the remaining methods, we use the publicly released models and fine-tune them on our OPEN-CIR dataset to eliminate the bias brought by differences in training datasets[3]. During comparison, we follow the original configurations of aforementioned methods. Specifically, Stable Diffusion [57], SDXL [55],

TABLE I
QUANTITATIVE COMPARISON EXPERIMENT RESULTS (WITH SKETCH AND TEXT DESCRIPTIONS). THROUGHOUT THE PAPER, ↑ (↓) MEANS HIGHER (LOWER) IS BETTER. BEST SCORES ARE HIGHLIGHTED IN **BOLD**.

Comparison with state-of-the-art methods					
Methods	FID ↓	SSIM (%) ↑	PSNR ↑	CLIP (%) ↑	Pref. (%) ↑
Stable Diffusion	3.74	62.79	17.78	91.67	8.24
Stable Diffusion*	4.87	61.55	15.75	89.96	6.96
SDXL	8.17	56.67	13.80	86.17	1.88
SDXL*	5.92	57.78	14.50	89.09	5.38
SDEdit	10.06	33.25	14.01	76.63	3.25
SDEdit*	8.72	32.07	13.91	77.63	4.13
SmartBrush*	4.31	62.22	16.90	91.13	7.13
InstructPix2Pix	11.59	45.18	13.37	78.86	3.63
ControlNet	4.19	63.35	16.48	90.98	6.00
ControlNet*	3.80	63.98	16.90	91.69	7.13
UniControlNet	9.04	19.91	9.19	80.67	4.75
UniControlNet*	7.94	28.75	10.34	83.31	3.25
T2I-Adapter	12.19	31.07	10.12	77.77	2.50
T2I-Adapter*	13.83	31.86	10.21	77.05	2.88
Ours (OpenCIR)	**3.37**	**66.65**	**18.92**	**93.13**	**32.89**

[2]https://huggingface.co/runwayml/stable-diffusion-v1-5

[3]We mark the fine-tuned method with an asterisk (*).

SDEdit [50], SmartBrush [81], and InstructPix2Pix [3] only rely on text descriptions, while ControlNet [84], UniControlNet [87], and T2I-Adapter [52] utilize a combination of Sketch and text descriptions. Although our OpenCIR supports the open condition modalities, we restrict the provided conditions to Sketch and text descriptions for a fair comparison.

Qualitative comparisons. We only show representative comparison methods in Fig. 4 due to space limitations. In Fig. 4, the first two rows show reconstruction results, and the remaining two rows present editing results. Stable Diffusion [57] has difficulty in interpreting text descriptions, resulting in the omission of instances (first row in Fig. 4, the missing fish). SDXL* [55] faces challenges in synthesizing the structure of instances according to user-provided Sketch guidance (second row in Fig. 4, the absence of creases in clothes). SmartBrush* [81] fails to accurately create the texture of specific instances (third row in Fig. 4, the text of the brand). ControlNet* [84] struggles to match the fine-grained details of instances to the conditions (fourth row in Fig. 4, misaligned chocolates in count and shape). In contrast, OpenCIR repaints specific instances with user-provided geometry, texture, and color conditions.

Quantitative comparisons. We show comprehensive quantitative results in Tab. I, where unmasked original images serve as the ground truth for calculating each metric. OpenCIR achieves the best score on all quantitative evaluation metrics, demonstrating its superior performance. Note that SDEdit [50], InstructPix2Pix [3], UniControlNet [87], and T2I-Adapter [52] are global image editing approaches, which places them at a disadvantage for editing specific instance within an image. The reason all methods report high CLIP scores is due to text descriptions being annotated by BLIP-2 [42], which selectively filters out descriptions with low CLIP similarity during the dataset processing phase (condition annotation details in Sec. III).

User study. In addition to qualitative and quantitative comparisons, we further conduct a user study experiment to find out whether images synthesized by our model are more favored by human observers compared to other state-of-the-art methods. We present participants with input conditions, original images, and candidates images synthesized using comparison methods and our OpenCIR. Participants are asked to select the most visually pleasing result according to input conditions. The experiment is published on Amazon Mechanical Turk (AMT), where we randomly select 100 samples from the testing set of the OPEN-CIR dataset. The experiment results are polled independently by 25 volunteers. As shown in Tab. I, our model achieves the highest preference score.

D. Comparison with previous CIR models.

Compared to relevant methods, the principal contribution of the CIR models lies in understanding multi-modal conditions while decoupling low-level image properties. As the advancement of CIR models, we compare with previous CIR models that supports interpreting text descriptions (*i.e.*, unified model [76] and two-phase model [74]). Since previous models are typically designed to repaint specific instance categories (*e.g.*, landscape and birds) using limited text descriptions. This limits their applicability for evaluation on our OPEN-CIR dataset. To make a fair comparison, we retrain our OpenCIR on the Caltech UCSD Birds [68] dataset and the COCO-Stuff [4] dataset. Considering that previous CIR models use the Gaussian noise as the texture condition that lacks content-related texture prior, we introduce two types of texture conditions to our OpenCIR when comparison: *(i)* using "null" as the texture condition, setting all elements of texture features to zero; *(ii)* using Canny [5] as the texture condition. We present the qualitative comparison results in Fig. 5, which demonstrates that OpenCIR achieves the most realistic compositing results, even without texture guidance. Quantitatively, OpenCIR achieves the highest scores across all four metrics on both datasets, as shown in Tab. II. Notably, providing Canny [5] as the texture condition can lead to further performance gains.

TABLE II
EVALUATION SCORES ON TWO ADDITIONAL DATASETS, QUANTITATIVELY COMPARED TO PREVIOUS CIR MODELS.

Methods	FID ↓	SSIM ↑	PSNR ↑	CLIP (%) ↑
Caltech UCSD Birds dataset				
Two-phase model	12.16	61.44	16.56	88.94
Unified model	10.54	62.60	17.77	91.06
OpenCIR (null)	3.78	86.29	23.18	93.13
OpenCIR (Canny)	**3.53**	**88.81**	**26.15**	**94.59**
COCO-Stuff dataset				
Two-phase model	18.37	32.60	12.72	87.79
Unified model	15.27	33.03	13.07	88.55
OpenCIR (null)	4.01	65.31	19.85	93.34
OpenCIR (Canny)	**3.74**	**67.44**	**21.28**	**94.19**

E. Ablation study

We disable several modules to establish four baselines for studying the impact of the corresponding modules. We also explore the effects of varying the conditional dropout probability and the maximum number of selected modalities. Since some of our ablations aims to evaluate the effectiveness of condition mixture injection (*e.g.*, W/o CMI), we conduct experiments using Laplace as the texture condition and randomly select color conditions for evaluation. In Tab. III, we present the evaluation scores. In Fig. 6, we show reconstruction results in the first two rows and editing results in the remaining two rows for the module disabling ablation.

W/o Condition Extension Strategy (CES). We disable the condition extension strategy and train all color conditions in a single phase. As a result, the model cannot decouple color and texture properties, leading to inadequate representation (first row in Fig. 6, withered and discolored petals).

W/o Condition Injection Modules (CIM). We discard optional cues in the color injection module and replace texture injection module with modules in ControlNet [84]. This leads to suboptimal color and texture representation (second row in Fig. 6, abrupt color and irregular texture of the knight).

W/o Condition Mixture Injection (CMI). We feed "null" into the color extractor when a corresponding color condition

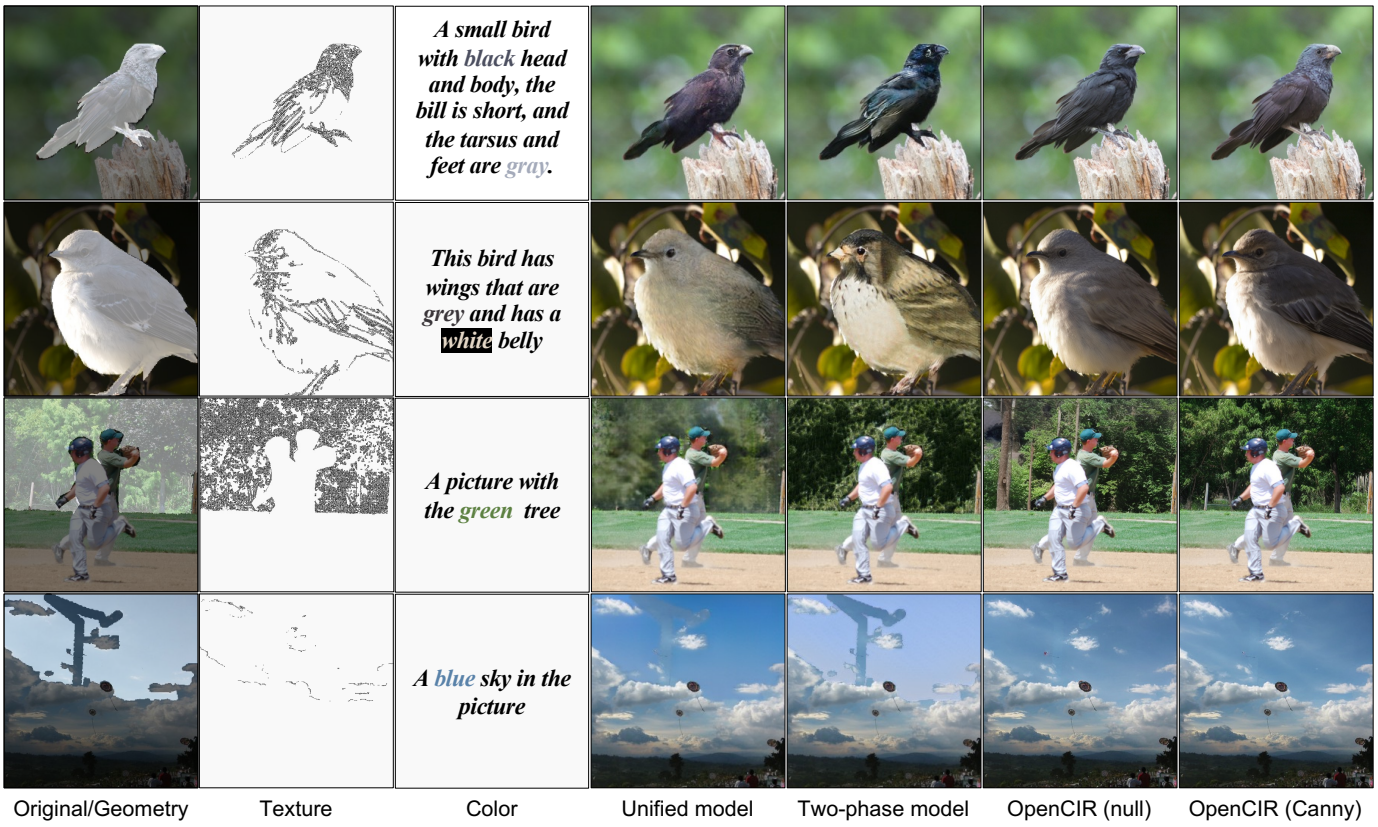

Fig. 5. Qualitative comparison results with previous CIR models. **Left:** Original images and conditions related to geometry (where), texture (what), and color (how). **Right:** Qualitative comparison results.

is absent, instead of discarding it by setting the weight to zero. Consequently, colors of repainted results become inaccurate (third row in Fig. 6, shifted colors of the wash painting).

W/o Contour Refinement Strategy (CRS). The pivotal component (Dice loss) is removed from the strategy, designed to estimate fine-grained instance contour. As a result, the boundaries between instances and background regions become jagged (fourth row in Fig. 6, the contour of the teddy bear).

Hyperparameter Sensitivity Analysis. We leverage the classifier-free guidance [25] with a dropout probability of 10% during the color condition extension, enabling joint training of unconditional and conditional models. This dropout probability aligns with the Stable Diffusion [57] and UniControlNet [87]. Empirically, we adopt the maximum number of selected modalities MixNum $= 3$ to achieve optimal condition mixture performance. To analyze the sensitivity, we conduct ablation study experiments by adjusting these hyperparameters. As the results shown in Tab. III, model performance degrades when deviating from the default values.

TABLE III
QUANTITATIVE ABLATION EXPERIMENT RESULTS (WITH LAPLACE AND RANDOMLY COLOR CONDITIONS).

Module Disabling Ablation				
Methods	FID ↓	SSIM (%) ↑	PSNR ↑	CLIP (%) ↑
W/o CIM	1.82	71.68	22.35	96.27
W/o CMI	1.85	71.29	22.20	96.17
W/o CES	2.31	70.94	21.43	95.53
W/o CRS	2.38	71.04	21.26	95.49
Hyperparameter Sensitivity Analysis				
Prob $= 5\%$	1.79	71.71	22.34	96.34
Prob $= 20\%$	1.77	71.98	22.36	96.37
MixNum $= 2$	1.82	71.68	22.30	96.28
MixNum $= 4$	1.76	71.78	22.42	96.40
Ours (OpenCIR)	**1.52**	**72.47**	**23.13**	**96.94**

F. Effectiveness of decoupling properties

We adopt the condition extension strategy to decouple the color and texture properties of the repainted instance (details see Sec. IV-C). We demonstrate the effectiveness of decoupling properties in following three aspects.

Condition extension. In the process of extending OpenCIR to support new color/texture conditions, we observe that results of the initial iteration tend to lose the corresponding color/texture property while preserving the other one. Specifically, as illustrated in Fig. 7 (left), the initial iteration of the color condition extension reveals that OpenCIR preserves the texture structure but fails to correctly represent colors, leading to colorless results. In addition, as demonstrated in Fig. 7 (right), the initial iteration of the texture condition extension presents a simplistic color distribution that omits instance details. This is because OpenCIR cannot fully understand the new conditions in the initial iteration of the condition extension. Meanwhile, it continues to render the previously mastered the other property.

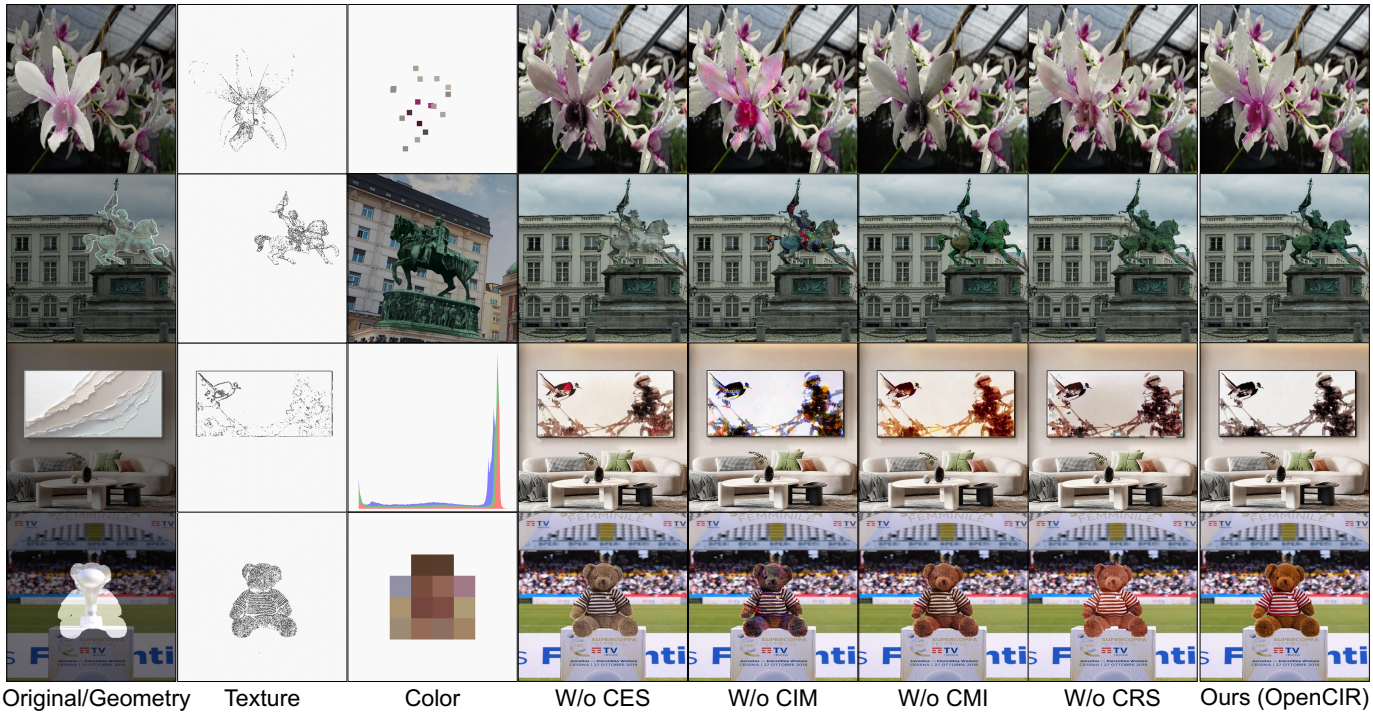

Fig. 6. Ablation study. **Left:** Original images and conditions related to geometry (where), texture (what), and color (how). **Right:** Ablation study results.

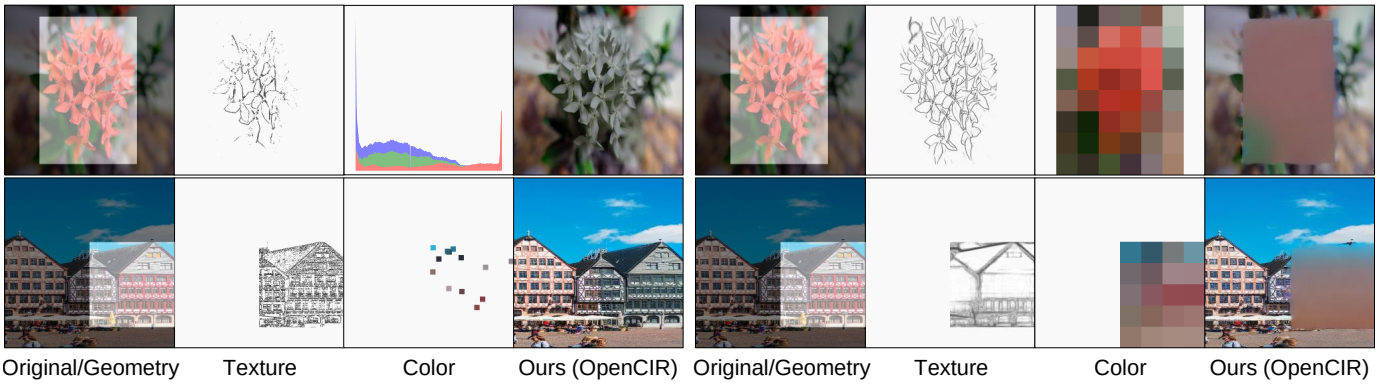

Fig. 7. Initial iteration visualization during color and texture condition extension. **Left:** Color condition extension. **Right:** Texture condition extension.

This serves as evidence that OpenCIR effectively decouples the color and texture properties.

Property alignment. We conduct additional experiments to demonstrate that each condition can exclusively represent the respective color and texture properties after the condition extension. As demonstrated in Fig. 8 (top), despite the text description mentioning the zebra, OpenCIR only applies the relevant black and white colors to the appropriate regions, ignoring the categorical context. In contrast, alternative methods (*i.e.*, ControlNet [84], UniControlNet [87], and T2I-Adapter [52]) render the zebra texture, losing the fidelity to the texture condition. Furthermore, as illustrated in Fig. 8 (bottom), although grayscale provides luminance that potentially indicates candidate colors (*e.g.*, brighter regions implying lighter colors [85]), OpenCIR correctly repaints instances according to the color condition (*e.g.*, applying dark colors in bright regions), along with preserving the texture structure. In this case, traditional colorization methods (*i.e.*, L-CoDe [79], UniColor [30], and L-CAD [7]) have difficulty in assigning corresponding colors according to the user-provided instruction. These examples demonstrate that the extended conditions are aligned with the color and texture spaces of the base model.

Condition modality. We further present qualitative results for repainting the same image under different condition modalities to explore the effects of the condition modalities. As shown in Fig. 9 (left), varying drawing styles may influence texture representation: Sketch-based repainting reveal fewer texture details, whereas those using Laplace highlight high-frequency features. As illustrated in Fig. 9 (right), distinct color conditions lead to varied color distributions: text descriptions provide optional color ranges for corresponding regions, while color patches explicitly define the average color for each repainted region. Notably, each condition affects only its corresponding property (*i.e.*, color or texture), preserving other aspects of the reconstructed image.

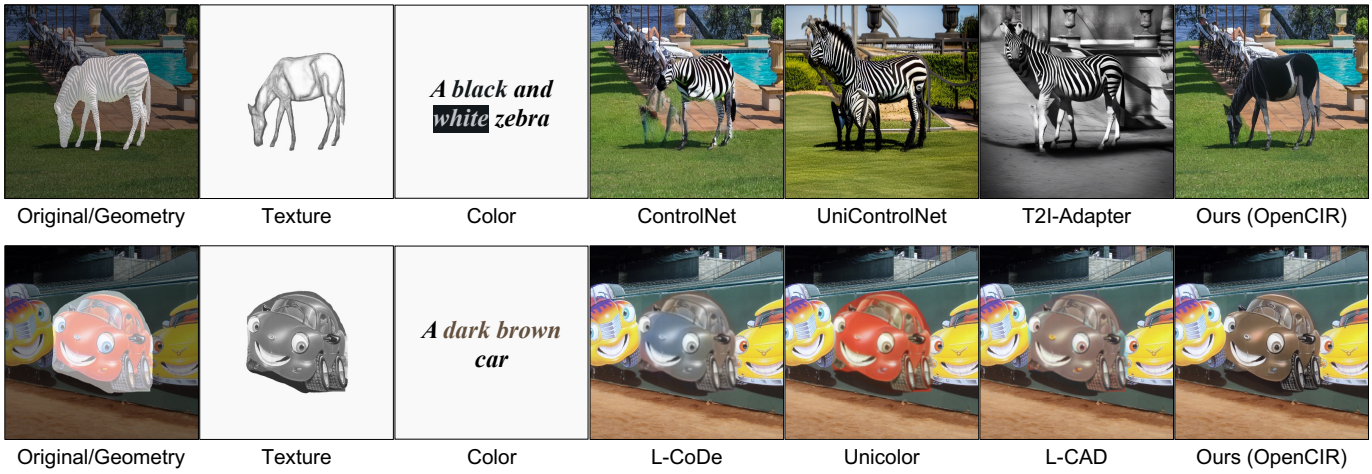

Fig. 8. Examples of property alignment results. **Top:** Color property alignment. **Bottom:** Texture property alignment.

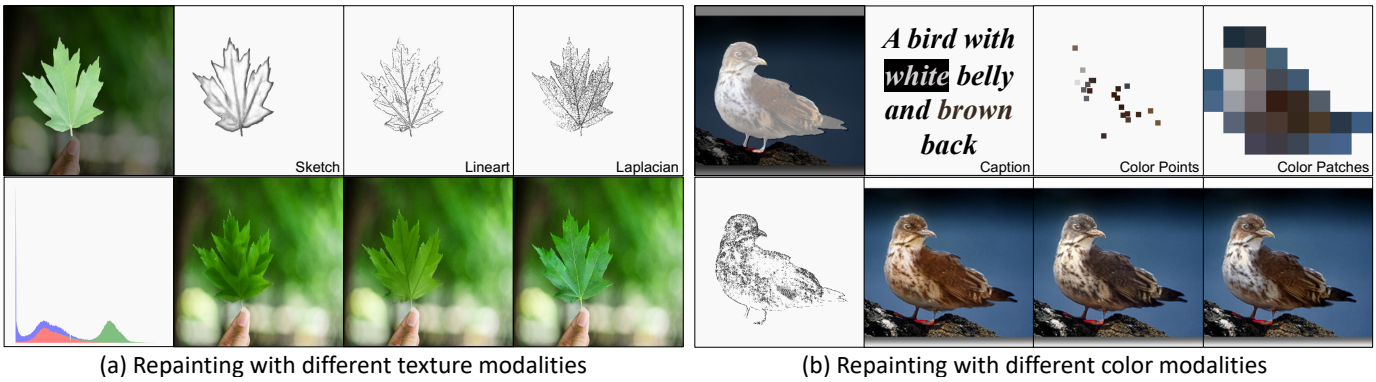

Fig. 9. Examples of image repainting results with different condition modalities. **Left:** Different texture conditions. **Right:** Different color conditions.

G. Controllability and Robustness

To demonstrate the fully-functional capabilities of the Open-CIR, we additionally conduct experiments to demonstrate its controllability and robustness under various conditions, including scenarios with conflicting conditions.

Controllability to open conditions. By providing distinct conditions related to geometry (where), texture (what), and color (how), we show six typical application scenarios of Open-CIR in Fig. 10 to repaint open object categories with open condition modalities and geometry precision: (a-b) For a given instance, we preserve its color or texture while modifying the opposite one to manipulate the specific property; (c) For multiple instances requiring individual contour adjustments, we use geometry conditions covering the instance contours to adjust their boundaries into unique shapes; (d) For the specific instance, we repaint its appearance using diverse color and texture conditions along with the same geometry condition to produce various repainted results; (e) For candidate instances to be removed, we match the geometry condition to the instance contours to remove individual instances; (f) In a scene lacking a subject, we introduce various instances by specifying where to place with the geometry condition, what the instance is with the texture condition, and how it appears with the color condition, thereby enriching the same background with different instances.

Robustness to conflicting conditions. Users may provide conflicted conditions when repainting images. We demonstrate OpenCIR's handling of such scenarios through four examples. As shown in Fig. 11 (top), although the two conditions describe different colors, they have different emphases. On the left, color points primarily focus on boxes, while text descriptions describe the overall colors. On the right, the color histogram describes the global color distribution, while text descriptions assign specific colors to the truck head. As a result, OpenCIR successfully integrates both. As illustrated in Fig. 11 (bottom), both color conditions focus on the same regions. On the left, both reference images and color points focus on the body of the car. On the right, both color patches and reference images aim to assign colors to the clothing. As a result, the repainting results exhibit a subtractive mixture effect, *e.g.*, mixing purple and green to produce brown. In all cases, OpenCIR produces visually pleasing results.

H. Limitation

While OpenCIR demonstrates strong capabilities in repainting photo-realistic images, it still faces challenges in repainting out-of-distribution images (*e.g.*, cartoons). This is primarily because these images have distinct condition distributions (*e.g.*, colors and textures). A promising approach to overcome this issue involves collecting training samples specific

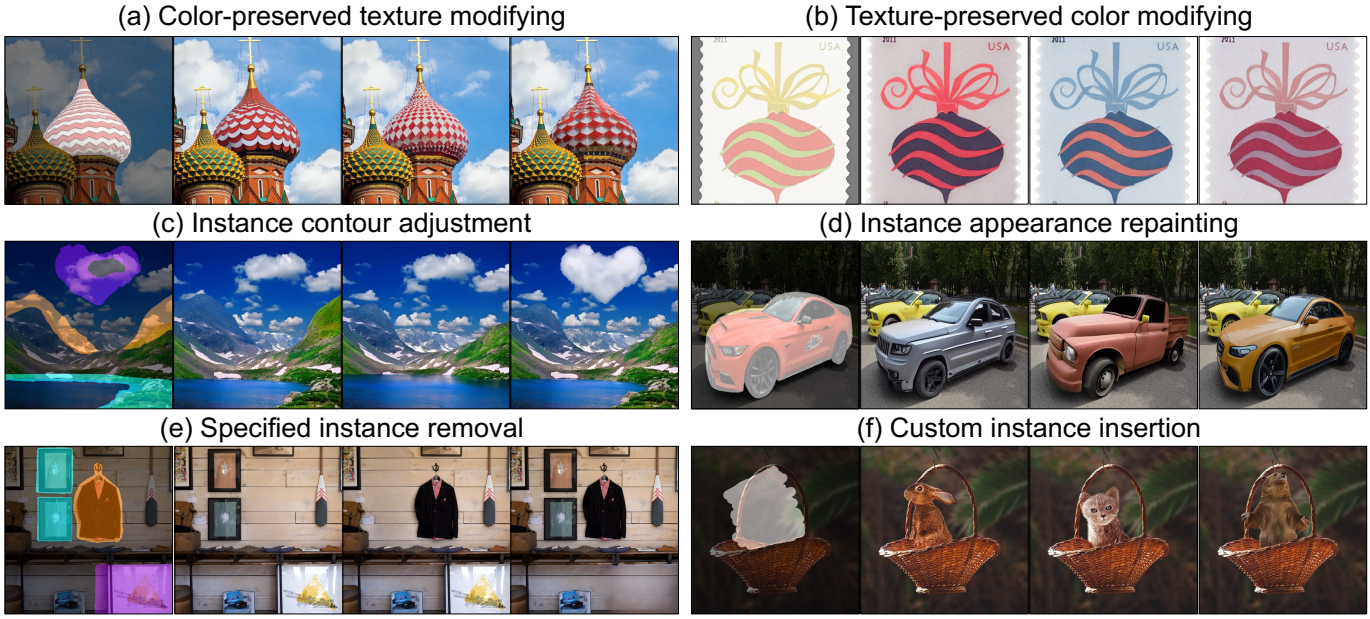

Fig. 10. Controllability study results of OpenCIR demonstrate its fully-functional capabilities across typical application scenarios. For each set, the original image is on the left, highlighting potential instances for repainting, followed by three variations of the repainted results.

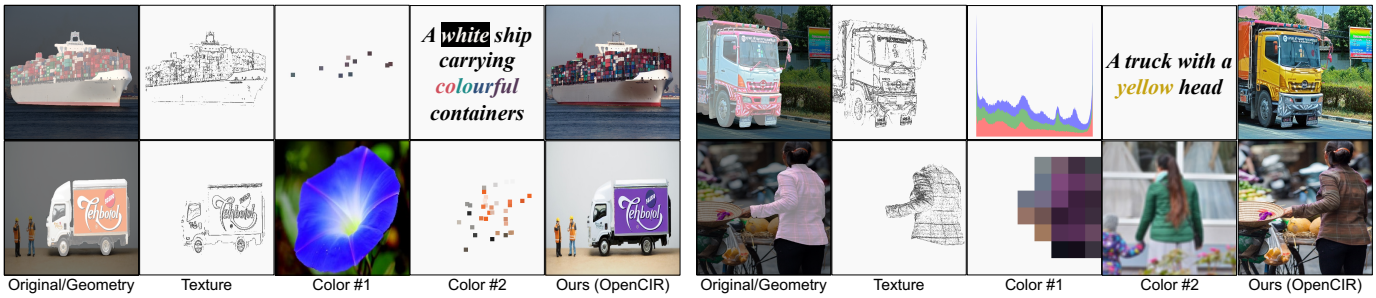

Fig. 11. Examples of repainting results with conflict conditions. **Top:** Conditions with different emphases. **Bottom:** Conditions focus on same image regions.

to the target domain, followed by fine-tuning the model or incorporating extra parameters [27]. We plan to explore this avenue in future work with the goal of improving the model's adaptability to a wider range of image domains.

VI. Conclusion

In this paper, we present OpenCIR, a fully-functional CIR model that aims at repainting specified visual content and then seamlessly compositing them into original images. OpenCIR utilizes generative priors of pre-trained generative models, redesigns condition injection modules with "open" principles, and adopts the condition extension and contour refinement strategies. These improvements enable OpenCIR to repaint open object categories with open condition modalities and geometry precision. For the comprehensive training and evaluation of OpenCIR, we additionally contribute a large-scale dataset with diverse and detailed condition annotations. Extensive experiments demonstrate that OpenCIR outperforms relevant state-of-the-art image editing methods.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant No. 62136001, 62088102 and Beijing Municipal Science & Technology Commission, Administrative Commission of Zhongguancun Science Park (Grant No. Z241100003524012).

References

[1] Lynton Ardizzone, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. In *ICLR*, 2018.
[2] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019.
[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2Pix: Learning to follow image editing instructions. In *CVPR*, 2023.
[4] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018.
[5] John Canny. A computational approach to edge detection. *TPAMI*, 1986.
[6] Caroline Chan, Frédo Durand, and Phillip Isola. Learning to generate line drawings that convey geometry and semantics. In *CVPR*, 2022.
[7] Zheng Chang, Shuchen Weng, Peixuan Zhang, Yu Li, Si Li, and Shi Boxin. L-CAD: Language-based colorization with any-level descriptions using diffusion priors. In *NeurIPS*, 2023.
[8] Zheng Chang, Shuchen Weng, Peixuan Zhang, Yu Li, Si Li, and Boxin Shi. L-CoIns: Language-based colorization with instance awareness. In *CVPR*, 2023.
[9] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. DoveNet: Deep image harmonization via domain verification. In *CVPR*, 2020.

[10] Guillaume Couairon, Asya Grechka, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. FlexIT: Towards flexible semantic image translation. In *CVPR*, 2022.

[11] Mohammad Reza Karimi Dastjerdi, Jonathan Eisenmann, Yannick Hold-Geoffroy, and Jean-François Lalonde. EverLight: Indoor-outdoor editable hdr lighting estimation. In *ICCV*, 2023.

[12] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. StyTr2: Image style transfer with transformers. In *CVPR*, 2022.

[13] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 2015.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[15] Huiyuan Fu, Wenkai Zheng, Xiangyu Meng, Xin Wang, Chuanming Wang, and Huadong Ma. You do not need additional priors or regularizers in retinex-based low-light image enhancement. In *CVPR*, 2023.

[16] Tsu-Jui Fu, Xin Eric Wang, and William Yang Wang. Language-driven artistic style transfer. In *ECCV*, 2022.

[17] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ICLR*, 2022.

[18] Rafael C Gonzales and Paul Wintz. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 1987.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[20] Bin He, Ce Wang, Boxin Shi, and Ling-Yu Duan. FHDe2Net: Full high definition demoireing network. In *ECCV*, 2020.

[21] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *TPAMI*, 2010.

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a nash equilibrium. In *NIPS*, 2017.

[23] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2016.

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[25] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[26] Yuchen Hong, Qian Zheng, Lingran Zhao, Xudong Jiang, Alex C Kot, and Boxin Shi. Panoramic image reflection removal. In *CVPR*, 2021.

[27] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2021.

[28] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. In *ICML*, 2023.

[29] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[30] Zhitong Huang, Nanxuan Zhao, and Jing Liao. UniColor: A unified framework for multi-modal colorization with transformer. In *SIGGRAPH Asia*, 2022.

[31] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 2008.

[32] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[33] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[34] Chanyong Jung, Gihyun Kwon, and Jong Chul Ye. Exploring patch-wise semantic relation for contrastive learning in image-to-image translation tasks. In *CVPR*, 2022.

[35] Xiaoyang Kang, Tao Yang, Wenqi Ouyang, Peiran Ren, Lingzhi Li, and Xuansong Xie. DDColor: Towards photo-realistic image colorization via dual decoders. In *ICCV*, 2023.

[36] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *CVPR*, 2023.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[38] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[39] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023.

[40] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *CVPR*, 2023.

[41] Gihyun Kwon and Jong Chul Ye. CLIPstyler: Image style transfer with a single text condition. In *CVPR*, 2022.

[42] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

[43] Junyi Li, Zhilu Zhang, Xiaoyu Liu, Chaoyu Feng, Xiaotao Wang, Lei Lei, and Wangmeng Zuo. Spatially adaptive self-supervised learning for real-world image denoising. In *CVPR*, 2023.

[44] Runde Li, Jinshan Pan, Zechao Li, and Jinhui Tang. Single image dehazing via conditional generative adversarial network. In *CVPR*, 2018.

[45] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. MAT: Mask-aware transformer for large hole image inpainting. In *CVPR*, 2022.

[46] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *ICCV*, 2021.

[47] Bolin Liu, Xiao Shu, and Xiaolin Wu. Demoireing of camera-captured screen images using deep convolutional neural network. *arXiv preprint arXiv:1804.03809*, 2018.

[48] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022.

[49] Long Ma, Tengyu Ma, Risheng Liu, Xin Fan, and Zhongxuan Luo. Toward fast, flexible, and robust low-light image enhancement. In *CVPR*, 2022.

[50] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2021.

[51] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[52] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2I-Adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. 2024.

[53] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z. Qureshi, and Mehran Ebrahimi. EdgeConnect: Generative image inpainting with adversarial edge learning. In *ICCVW*, 2019.

[54] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *SIGGRAPH*, 2023.

[55] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*.

[56] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[57] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[58] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[59] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023.

[60] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[61] Jheng-Wei Su, Hung-Kuo Chu, and Jia-Bin Huang. Instance-aware image colorization. In *CVPR*, 2020.

[62] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *ICCV*, 2021.

[63] Jimeng Sun, Shuchen Weng, Zheng Chang, Si Li, and Boxin Shi. UniCoRN: A unified conditional image repainting network. In *CVPR*, 2022.

[64] Jiajun Tang, Haofeng Zhong, Shuchen Weng, and Boxin Shi. LuminAIRe: Illumination-aware conditional image repainting for lighting-realistic generation. In *NeurIPS*, 2023.

[65] Jiajun Tang, Yongjie Zhu, Haoyu Wang, Jun Hoong Chan, Si Li, and Boxin Shi. Estimating spatially-varying lighting in urban scenes with disentangled representation. In *ECCV*, 2022.

[66] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *CVPR*, 2017.

[67] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *CVPR*, 2018.

[68] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[69] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018.

[70] Yaxiong Wang, Yunchao Wei, Xueming Qian, Li Zhu, and Yi Yang. Sketch-guided scenery image outpainting. *TIP*, 2021.

[71] Zhizhong Wang, Lei Zhao, and Wei Xing. StyleDiffusion: Controllable disentangled style transfer via diffusion models. In *ICCV*, 2023.

[72] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.

[73] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. ELITE: Encoding visual concepts into textual embeddings for customized text-to-image generation. In *ICCV*, 2023.

[74] Shuchen Weng, Wenbo Li, Dawei Li, Hongxia Jin, and Boxin Shi. Conditional image repainting via semantic bridge and piecewise value function. In *ECCV*, 2020.

[75] Shuchen Weng, Wenbo Li, Dawei Li, Hongxia Jin, and Boxin Shi. MISC: Multi-condition injection and spatially-adaptive compositing for conditional person image synthesis. In *CVPR*, 2020.

[76] Shuchen Weng and Boxin Shi. Conditional image repainting. *TPAMI*, 2023.

[77] Shuchen Weng, Jimeng Sun, Yu Li, Si Li, and Boxin Shi. CT^2: Colorization transformer via color tokens. In *ECCV*, 2022.

[78] Shuchen Weng, Yi Wei, Ming-Ching Chang, and Boxin Shi. Instance contour adjustment via structure-driven CNN. In *ECCV*, 2022.

[79] Shuchen Weng, Hao Wu, Zheng Chang Chang, Jiajun Tang, Si Li, and Boxin Shi. L-CoDe: Language-based colorization using color-object decoupled conditions. In *AAAI*, 2022.

[80] Shuchen Weng, Peixuan Zhang, Zheng Chang, Xinlong Wang, Si Li, and Boxin Shi. Affective image filter: Reflecting emotions from text to images. In *ICCV*, 2023.

[81] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. SmartBrush: Text and shape guided object inpainting with diffusion model. In *CVPR*, 2023.

[82] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018.

[83] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Free-form image inpainting with gated convolution. In *ICCV*, 2019.

[84] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[85] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

[86] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. In *CVPR*, 2018.

[87] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-ControlNet: All-in-one control to text-to-image diffusion models. In *NeurIPS*, 2024.

[88] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

Shuchen Weng is a research staff in Beijing Academy of Artificial Intelligence. He received PhD degree from the Peking University in 2024. His research interests include cross-modality (mainly language-based) content creation and manipulation.

Xiaocheng Gong is a MS student in School of Artificial Intelligence, Beijing University of Posts and Telecommunications. His current research interests include cross-modality content manipulation and multimodal information extraction.

Haojie Zheng is a joint PhD student in School of Software and Microelectronics, Peking University and Beijing Academy of Artificial Intelligence. His research focuses on cross-modal content creation and manipulation, particularly in language-based modalities.

Xinlong Wang is a technical lead at Beijing Academy of Artificial Intelligence (BAAI), leading and founding the vision and multimodal research center. He received my PhD degree from The University of Adelaide, supervised by Prof. Chunhua Shen. Before that he obtained his Bachelor's degree from Tongji University. He was a recipient of the Google PhD Fellowship. His research interests lie in the area of computer vision and foundation models.

Si Li received the Ph.D. degree from the Beijing University of Posts and Telecommunications in 2012. She is currently an associate Professor with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications. Her current research interests include multimodal artificial intelligence and machine learning.

Boxin Shi received the BE degree from the Beijing University of Posts and Telecommunications, the ME degree from Peking University, and the PhD degree from the University of Tokyo, in 2007, 2010, and 2013. He is currently a Boya Young Fellow Associate Professor (with tenure) and Research Professor at Peking University, where he leads the Camera Intelligence Lab. Before joining PKU, he did research with MIT Media Lab, Singapore University of Technology and Design, Nanyang Technological University, National Institute of Advanced Industrial Science and Technology, from 2013 to 2017. His papers were awarded as Best Paper, Runners-Up at CVPR 2024, ICCP 2015, and selected as Best Paper candidate at ICCV 2015. He is an associate editor of TPAMI/IJCV and an area chair of CVPR/ICCV/ECCV. He is a senior member of IEEE.

Supplementary Material of OpenCIR: Conditional Image Repainting with Open Condition Mixture

Shuchen Weng†, Xiaocheng Gong†, Haojie Zheng†
Xinlong Wang, Si Li‡, and Boxin Shi, *Senior Member, IEEE*

APPENDIX

A. Task differences

In Tab. S1, we summarize the supported modalities and functionalities of relevant image editing methods, existing CIR models, and our approach. ✓/✗indicates the model supports/lacks the specified modality/functionality. We categorize modalities into three groups: text descriptions (*e.g.*, instructions expressed in various languages), visual elements (*e.g.*, reference images, semantic maps, and other image-like conditions), and statistical concepts (*e.g.*, attributes, color histograms, and motion points). In addition, model functionalities are also divided into three categories: local editing (*e.g.*, modifying specific regions while preserving remaining regions entirely unchanged), composable controllability (*e.g.*, utilizing multiple modalities simultaneously for image editing), and property decoupling (*e.g.*, modifying specific low-level properties of the image, such as color, texture, and geometry). Compared to relevant methods, the principal contribution of the CIR models lies in understanding multi-modal conditions while decoupling low-level image properties.

TABLE S1
DIFFERENCES BETWEEN CIR MODELS AND RELEVANT METHODS.

Method	Modality group			Functionality categories		
	Text descriptions	Visual elements	Statistical concepts	Local editing	Composable controllability	Property decoupling
RePaint [10]	✗	✗	✗	✓	✗	✗
DragGAN [13]	✗	✗	✓	✗	✗	✓
DragDiffusion [17]	✗	✗	✓	✗	✗	✓
PaintByExample [23]	✗	✓	✗	✓	✗	✗
InstructPix2Pix [3]	✓	✗	✗	✗	✗	✗
Pix2PixZero [14]	✓	✗	✗	✗	✗	✗
Imagic [8]	✓	✗	✗	✗	✗	✗
RichText [6]	✓	✗	✓	✗	✓	✗
SDEdit [11]	✓	✓	✗	✗	✓	✗
UniControlNet [27]	✓	✓	✗	✗	✓	✗
T2I-Adapter [12]	✓	✓	✗	✗	✓	✗
SceneComposer [24]	✓	✓	✗	✗	✓	✗
Stable Diffusion [16]	✓	✓	✗	✓	✗	✗
SDXL [16]	✓	✓	✗	✓	✗	✗
SmartBrush [22]	✓	✓	✗	✓	✓	✗
BlendDiffusion [2]	✓	✓	✗	✓	✓	✗
ControlNet [25]	✓	✓	✗	✓	✓	✗
Composer [7]	✓	✓	✓	✓	✓	✗
MISC [21]	✗	✓	✓	✓	✓	✓
UniCoRN [18]	✗	✓	✓	✓	✓	✓
LuminAIRe [19]	✗	✓	✓	✓	✓	✓
Repainting [20]	✓	✓	✗	✓	✓	✓
Ours (OpenCIR)	✓	✓	✓	✓	✓	✓

B. Additional qualitative results.

We present additional comparison experiment results with Stable Diffusion [16], SDXL [15], SDEdit [11], SmartBrush [22], InstructPix2Pix [3], ControlNet [25], UniControlNet [27], and T2I-Adapter [12] on OPEN-CIR dataset in Fig. S1 and Fig. S2, where we transpose the layout to make the figure larger. The first four columns show the comprehensive qualitative comparisons presented in Fig. 4 of the main paper. The remaining four columns show additional reconstruction results and editing results. Zoom-in details are additionally provided in Fig. S3 and Fig. S4.

In addition, we show more qualitative comparison results with previous CIR models in Fig. S5, whose details can be found in Sec V-D of the main paper. More ablation study results are shown in Fig. S6, whose details are presented in Sec. V-E of the main paper.

C. Additional comparisons with state-of-the-art methods

We additionally compare OpenCIR with a broader range of models, including Repaint [10], SD3 [5], SD3.5 [1], and FLUX [9]. Since the official inpainting pipeline for SD3.5 [1] is unavailable, we implement it following the approach of SDXL [15], denoted as SD3.5-Fill. Similarly, we denote the inpainting pipeline for FLUX [9] as FLUX-Fill. As ControlNet pipelines for SD3.5 and FLUX do not support the Sketch condition used in Sec. V-C of the main paper, we select the most similar Canny as the provided condition, denoted as SD3.5-Canny and Flux-Canny, respectively. During comparison, we follow the original configurations of the aforementioned methods when editing images. Specifically, Repaint [10] takes no

S. Weng, H. Zheng, and X. Wang are with Beijing Academy of Artificial Intelligence, Beijing 100083, China. Email: {scweng, wangxinlong}@baai.ac.cn.

H. Zheng is also with School of Software and Microelectronics, Peking University, Beijing 100871. Email: suimu@stu.pku.edu.cn.

X. Gong and S. Li are with School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China. Email: {xiaochengkung, lisi}@bupt.edu.cn.

B. Shi is with State Key Laboratory of Multimedia Information Processing and National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, Beijing 100871, China. Email: shiboxin@pku.edu.cn.

† Equal contribution. ‡ Corresponding author.

TABLE S2
ADDITIONAL QUANTITATIVE COMPARISON EXPERIMENT RESULTS. THROUGHOUT THE PAPER, ↑ (↓) MEANS HIGHER (LOWER) IS BETTER. BEST SCORES ARE HIGHLIGHTED IN **BOLD**.

Additioanl comparison with state-of-the-art methods				
Methods	FID ↓	SSIM (%) ↑	PSNR ↑	CLIP (%) ↑
Repaint	9.18	72.73	17.25	85.39
SD3	3.56	68.34	18.15	92.03
SD3.5-Fill	4.70	73.16	16.82	91.12
FLUX-Fill	2.93	73.64	20.01	93.16
SD3.5-Canny	8.25	32.77	10.87	81.79
FLUX-Canny	10.93	26.54	9.69	74.89
OpenCIR-Canny	**2.11**	**78.73**	**19.31**	**94.67**

additional condition for handling open categories; SD3 [5], SD3.5-Fill [1], and FLUX-Fill [9] use only text descriptions; and SD3.5-Canny [1] and FLUX-Canny [9] use both Canny edges and text descriptions.

Qualitative comparisons. As shown in Fig. S7, the first two rows show reconstruction results, and the remaining two rows present editing results. Repaint [10] tends to remove the instance or generate an unknown one (first row in Fig. S7, the disappearing motorcycle). SD3 [5] struggles to repaint the instance with the appropriate contour (second row in Fig. S7, strange house shape). SD3.5-Fill [1] and FLUX-Fill [9] repaint high-quality instances, but lack structural similarity to the input texture condition (third row in Fig. S7, the unnatural text of the brand). SD3.5-Canny [1] and FLUX-Canny [9] repaint instances based on texture guidance, but significantly modify the background appearance (fourth row in Fig. S7, the bright room). Instead, OpenCIR faithfully repaints specific instances according to the user-provided conditions.

Quantitative comparisons. As comprehensive quantitative results shown in Tab. S2, OpenCIR achieves the best score on all quantitative evaluation metrics. Note that SD3.5-Canny and Flux-Canny are global image editing approaches, which places them at a disadvantage for editing specific instance within user-provided images.

D. Architecture of the texture condition injection module

We present the architecture of the texture condition injection module in Fig. S8 (a). Specifically, it consists of a texture extraction network (Fig. S8 (b)) followed by four stacked texture injection blocks (Fig. S8 (c)). The extraction network employs a stack of convolutional layers and SiLU activations [4]. Each injection block contains two Spatially-Adaptive Texture Injection (SATI) modules (Fig. S8 (d)). As described in Eq. (4) of the main paper, SATI modules modulate image features with corresponding texture features and instance masks to understand distinct drawing styles of texture conditions for specified regions.

E. Differences with ControlNet

While ControlNet [26] has shown strong performance in controllable image generation, it primarily adds features from an additional network branch, which may not effectively disentangle fine-grained editing objectives (*e.g.*, determining where to place instances, what those instances are, and how they should appear) for the condition image repainting task. To address this limitation, we employ distinct strategies and tailored injection mechanisms: *(i)* **Training strategy.** We utilize a contour refinement strategy guided by instance masks to precisely control the placement (where), and a condition extension strategy to decouple texture (what) and color (how) properties. *(ii)* **Texture injection.** To adaptively integrate diverse drawing styles from texture conditions, we inject texture features via feature modulation, rather than direct feature addition as commonly used in ControlNet [26]. This allows for a more nuanced integration suitable for stylistic variations. *(iii)* **Color injection.** To handle potentially ambiguous color correspondences, our module incorporates explicit instance-color cues, rather than relying solely on cross attention as ControlNet [26]. This provides precise instance-specific color mapping. As a result, our OpenCIR achieves leading performance across all four metrics and user study results.

F. Discussion of detail correlation

We observe that OpenCIR tends to correlate user-provided drawing details with the repainting details. Consequently, achieving results with high detail may require correspondingly elaborate hand-drawn texture conditions. A potential approach is to design a "detail scale factor" and further refine the training data distribution. This could decouple detail correspondence between user-provided condition and repainted results, enabling detailed generation even from coarser sketches and offering users greater flexibility. We plan to explore this possibility in future work.

G. Dataset sample

As illustrated in Sec. III of the main paper, we collect and process a large-scale dataset tailored for the comprehensive training and evaluation of CIR models. To further demonstrate its diversity of object categories, we show more foreground examples in Fig. S9.

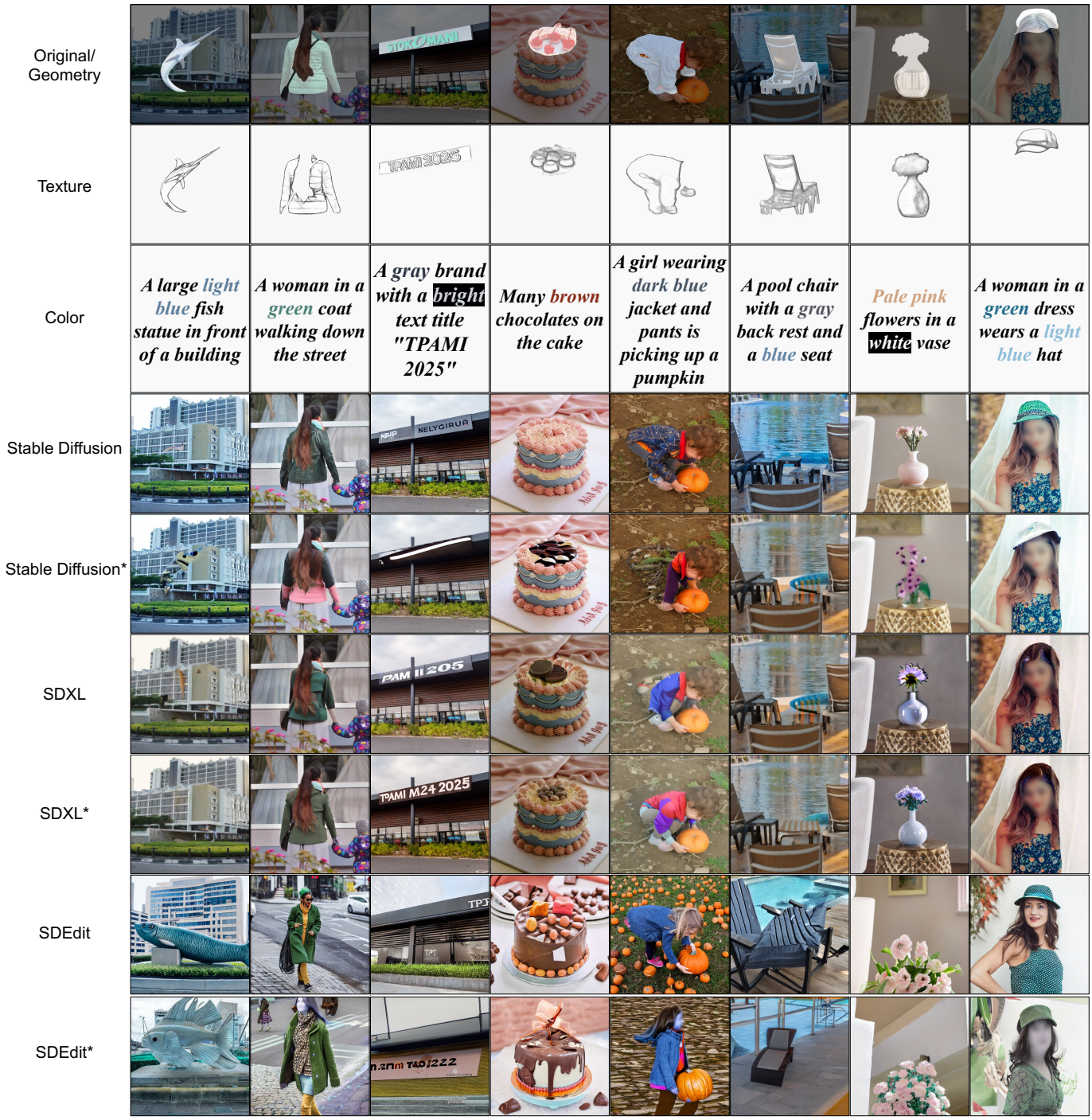

Fig. S1. Qualitative comparison with state-of-the-art image editing methods. **Top:** Original images and conditions related to geometry (where), texture (what), and color (how). **Bottom:** Qualitative comparison results.

Fig. S2. Qualitative comparison with state-of-the-art image editing methods.

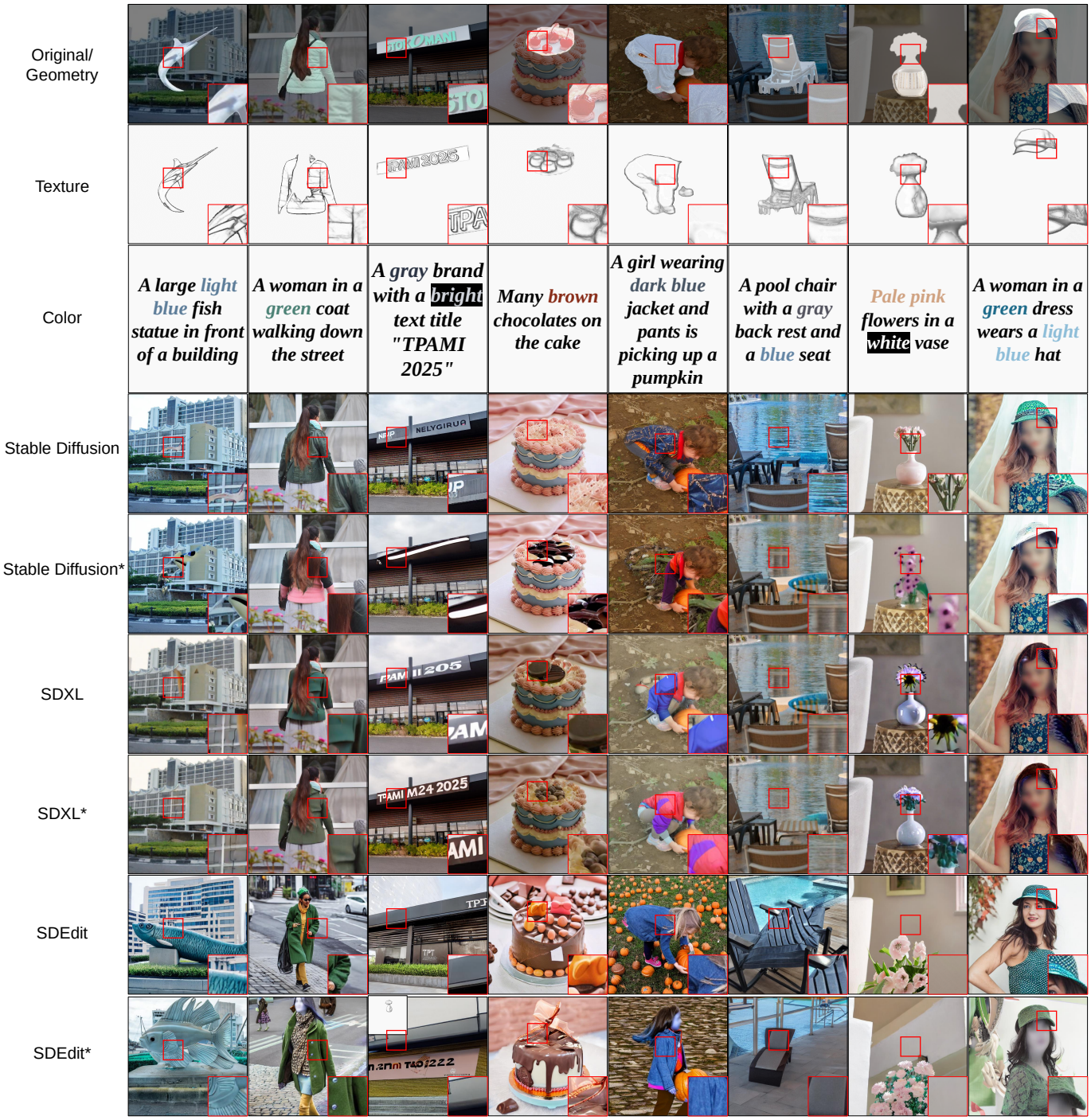

Fig. S3. Zoom-in details of qualitative comparison with state-of-the-art image editing methods. **Top**: Original images and conditions related to geometry (where), texture (what), and color (how). **Bottom**: Qualitative comparison results.

Fig. S4. Zoom-in details of qualitative comparison with state-of-the-art image editing methods.

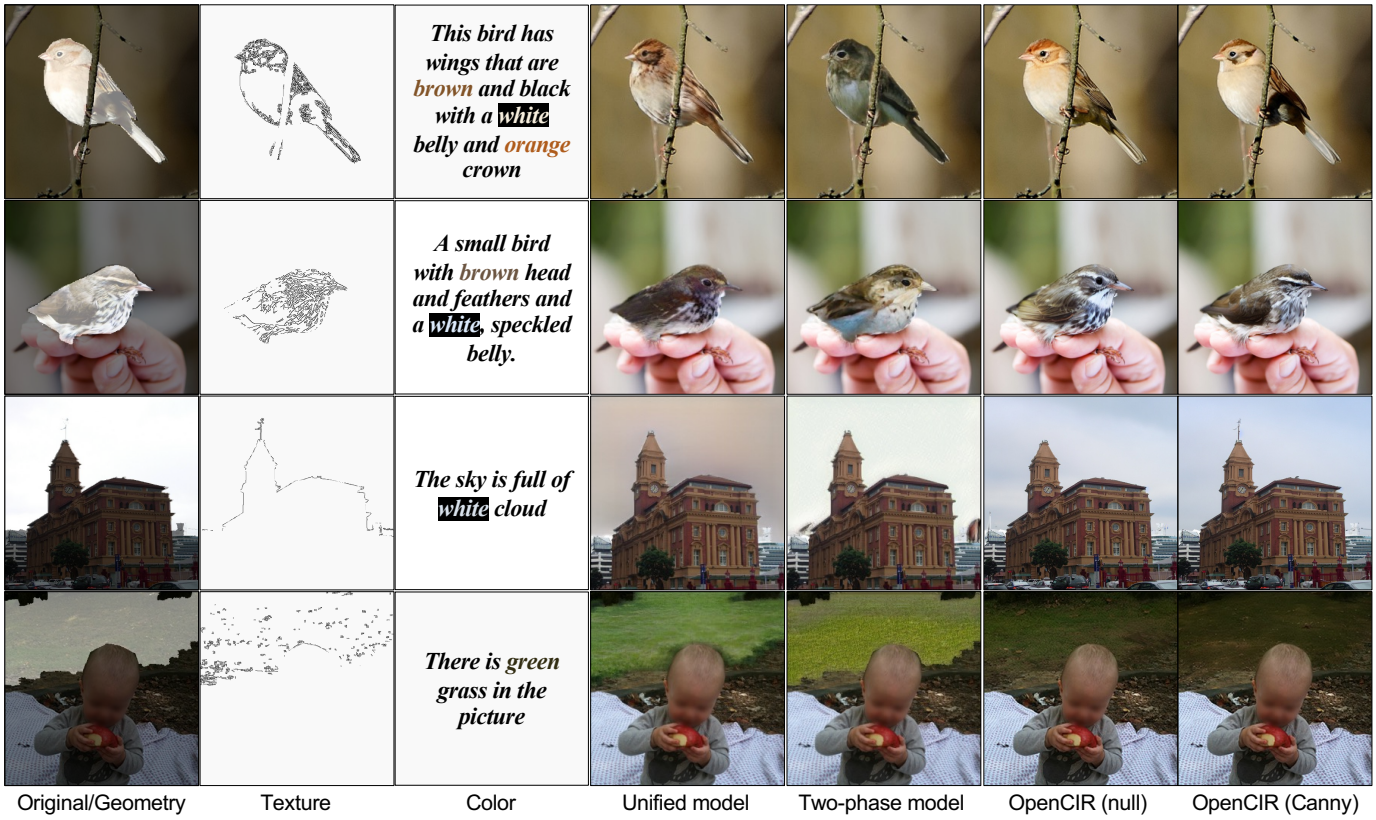

Fig. S5. Qualitative comparison results with previous CIR models. **Left:** Original images and conditions related to geometry (where), texture (what), and color (how). **Right:** Qualitative comparison results.

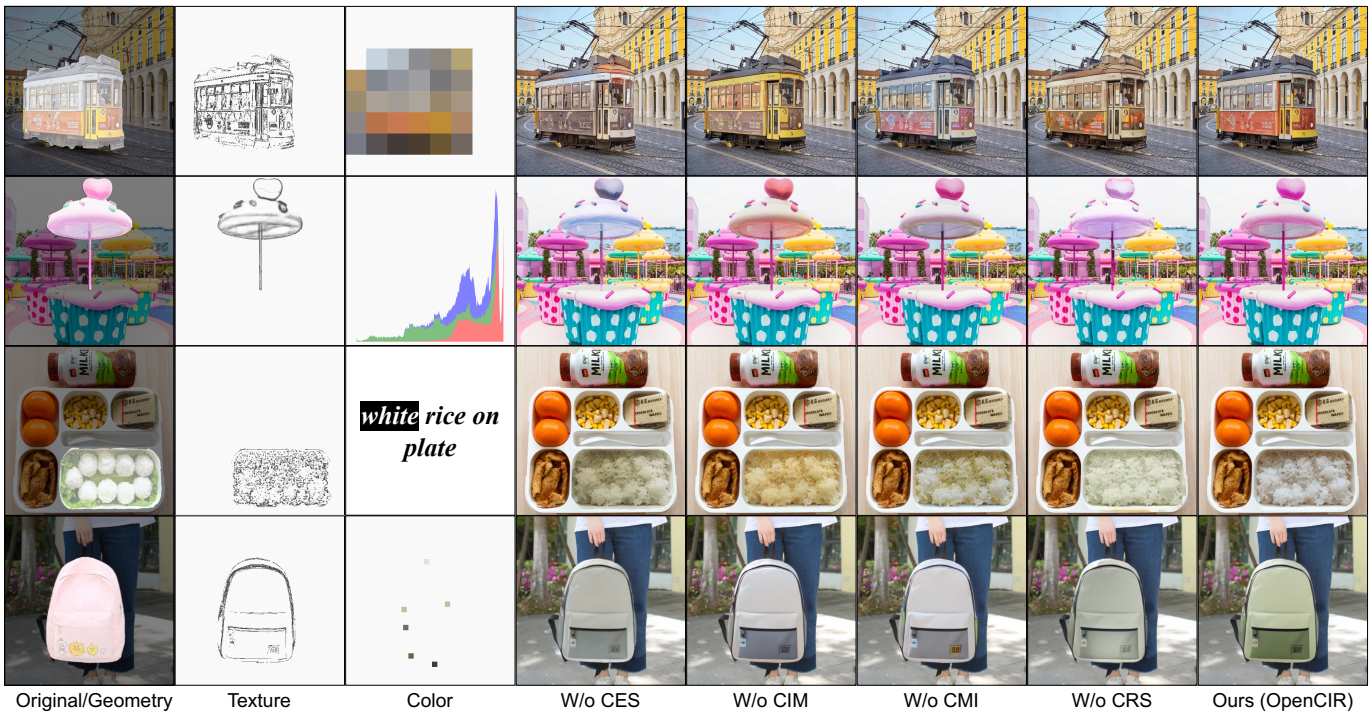

Fig. S6. Ablation study with different variants of the proposed method. **Left:** Original images and conditions related to geometry (where), texture (what), and color (how). **Right:** Ablation study results.

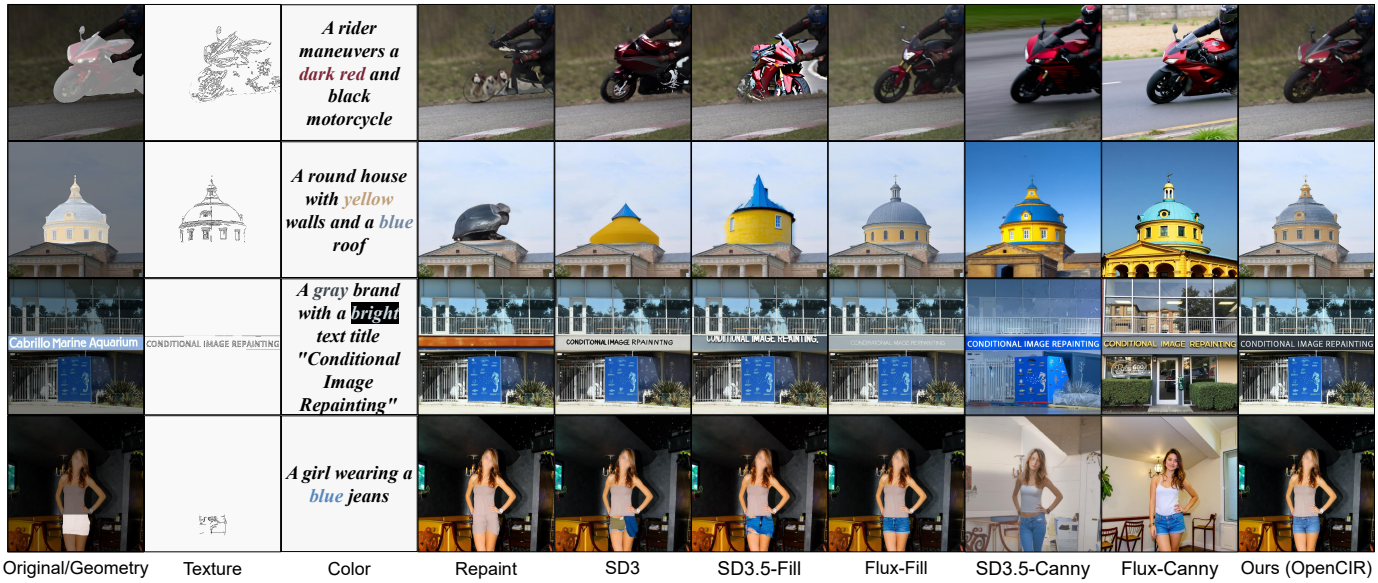

Fig. S7. Additional qualitative comparison with state-of-the-art image editing methods. **Left**: Original images and conditions related to geometry (where), texture (what), and color (how). **Right**: Qualitative comparison results.

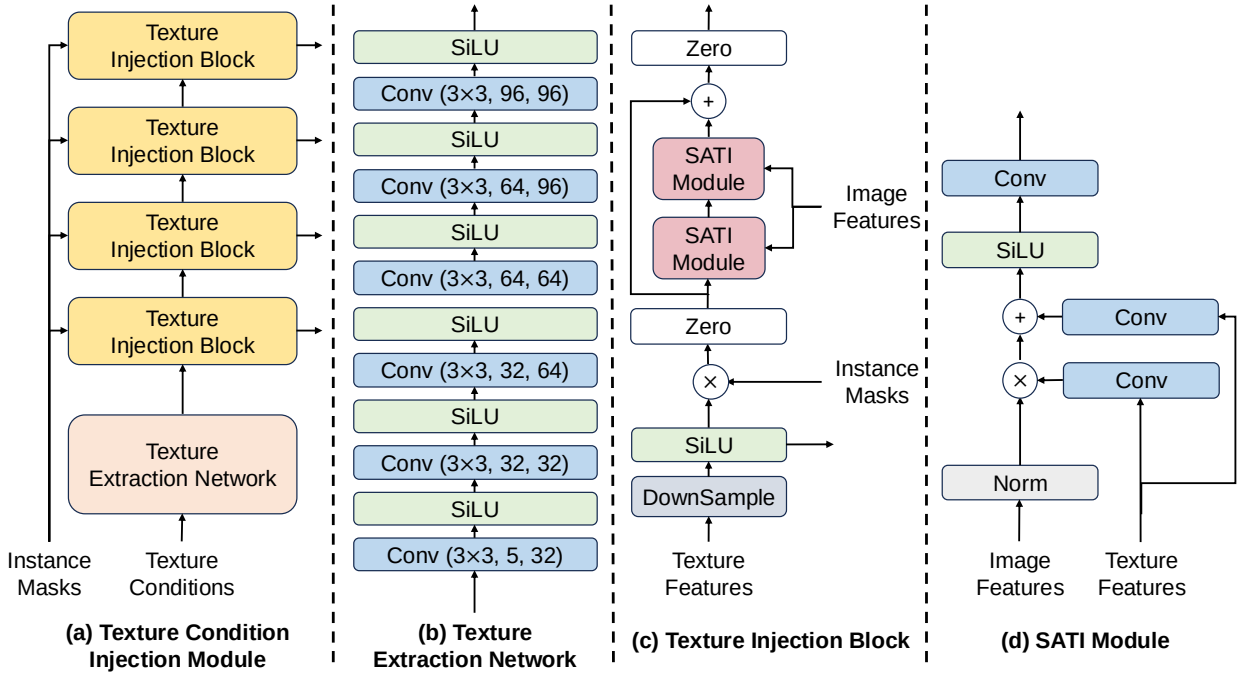

Fig. S8. Detailed architecture of the texture condition injection module.

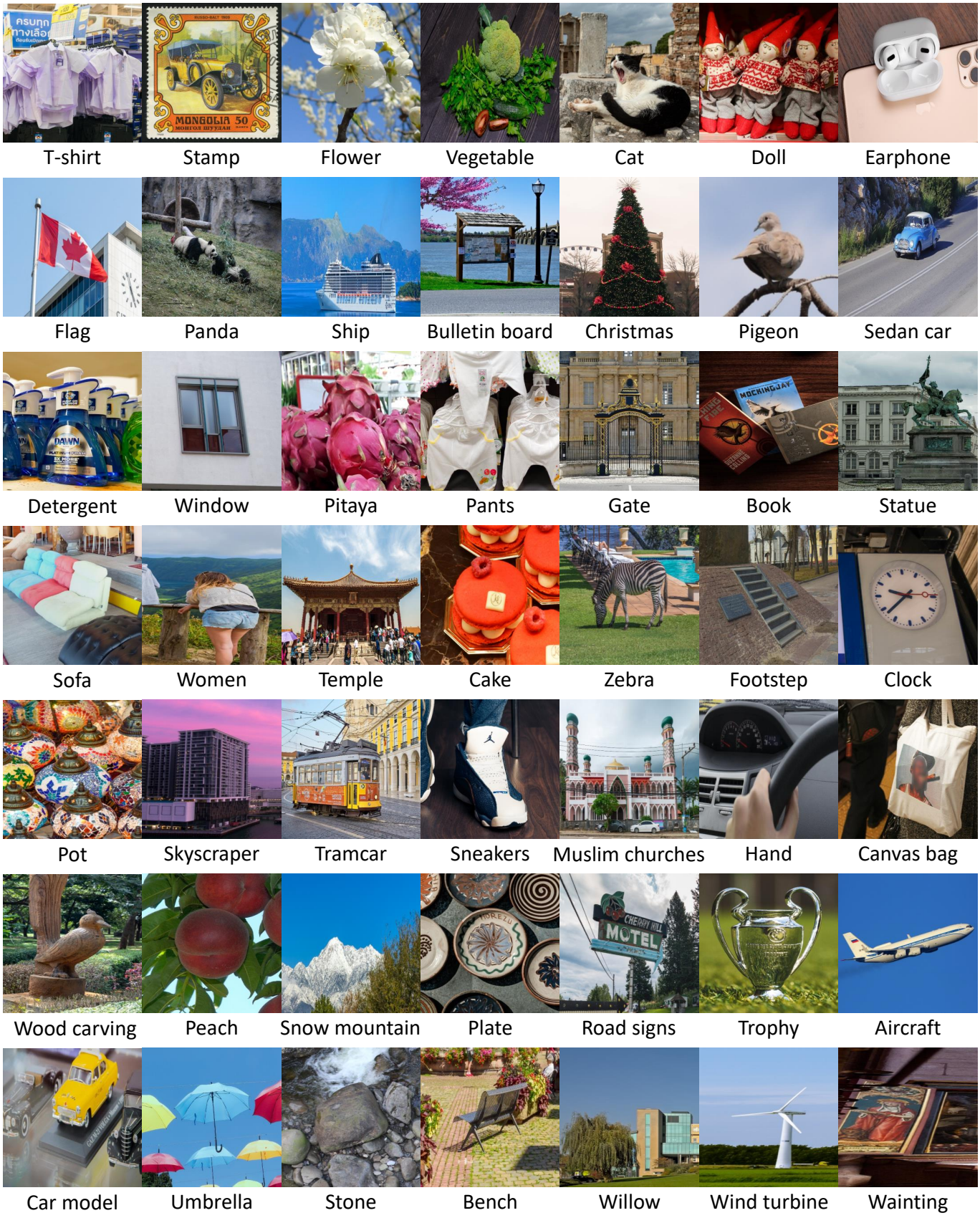

Fig. S9. More foreground examples of the OPENCIR dataset.

REFERENCES

[1] Stability AI. Stable diffusion 3.5. https://github.com/Stability-AI/sd3.5, 2024. Accessed: 2025-03-21.

[2] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, 2022.

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2Pix: Learning to follow image editing instructions. In *CVPR*, 2023.

[4] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 2018.

[5] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*.

[6] Songwei Ge, Taesung Park, Jun-Yan Zhu, and Jia-Bin Huang. Expressive text-to-image generation with rich text. In *ICCV*, 2023.

[7] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. In *ICML*, 2023.

[8] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *CVPR*, 2023.

[9] Black Forest Labs. FLUX. https://github.com/black-forest-labs/flux, 2024.

[10] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022.

[11] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2021.

[12] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2I-Adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

[13] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023.

[14] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *SIGGRAPH*, 2023.

[15] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*.

[16] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[17] Yujun Shi, Chuhui Xue, Jiachun Pan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. *arXiv preprint arXiv:2306.14435*, 2023.

[18] Jimeng Sun, Shuchen Weng, Zheng Chang, Si Li, and Boxin Shi. UniCoRN: A unified conditional image repainting network. In *CVPR*, 2022.

[19] Jiajun Tang, Haofeng Zhong, Shuchen Weng, and Boxin Shi. LuminAIRe: Illumination-aware conditional image repainting for lighting-realistic generation. In *NeurIPS*, 2023.

[20] Shuchen Weng, Wenbo Li, Dawei Li, Hongxia Jin, and Boxin Shi. Conditional image repainting via semantic bridge and piecewise value function. In *ECCV*, 2020.

[21] Shuchen Weng, Wenbo Li, Dawei Li, Hongxia Jin, and Boxin Shi. MISC: Multi-condition injection and spatially-adaptive compositing for conditional person image synthesis. In *CVPR*, 2020.

[22] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. SmartBrush: Text and shape guided object inpainting with diffusion model. In *CVPR*, 2023.

[23] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *CVPR*, 2023.

[24] Yu Zeng, Zhe Lin, Jianming Zhang, Qing Liu, John Collomosse, Jason Kuen, and Vishal M Patel. Scenecomposer: Any-level semantic image synthesis. In *CVPR*, 2023.

[25] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[26] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[27] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-ControlNet: All-in-one control to text-to-image diffusion models. In *NeurIPS*, 2024.