arXiv:2608.06929v1 [cs.CV] 7 Aug 2026

MASKFLOW: PRECISE, CONSISTENT AND SEAMLESS REGIONAL IMAGE EDITING

Rui Xu[1], **Yang Yong**[1], **Shunzi Yang**[2], **Ruihao Gong**[1,2]✉, **Chengtao Lv**[1,3]
[1] SenseTime Research [2] Beihang University [3] Nanyang Technological University
chiaroair@gmail.com yongyang@sensetime.com
yangshunzi@buaa.edu.cn gongruihao@buaa.edu.cn chengtao001@e.ntu.edu.sg
✉ Corresponding author Project page: https://reychiaro.github.io/MaskFlow



(a) The reference-based method edits the wrong label despite the provided mask in the fourth row.

(b) The reference-based method alters unrelated leaves and flowers outside the mask.

(c) The inpainting method leaves visible seams along the mask boundary.

Figure 1: Qualitative comparison of the proposed **MaskFlow** with representative reference-based and inpainting methods. MaskFlow precisely localizes edits to masked regions, consistently preserves background content, and produces seamless transitions across mask boundaries.

ABSTRACT

Regional image editing has attracted considerable attention for its spatial controllability. Although instruction-based and mask-reference-based editing methods can achieve strong semantic alignment, reliable regional control remains challenging, where an edit must be accurately localized and naturally integrated with the preserved context. We propose **MaskFlow**, a training framework for precise localization, consistent background preservation, and seamless boundary transitions. MaskFlow incorporates the mask into the probability path and flow-matching objective, coordinating generation within the editable region with source preservation outside it. The proposed Soft-Poisson de-seaming module further refines the

predicted vector field during both training and sampling to improve the smooth integration of the edited foreground with the preserved background. We also design a data synthesis pipeline to construct MEData, a mask-based image editing dataset for training regional image editing models and facilitating further research. Experiments on natural scenes and infographic images demonstrate consistent improvements over competing methods in both quantitative and qualitative evaluations.

1 INTRODUCTION

Recent advances in diffusion and flow-matching models have substantially improved the fidelity and controllability of image generation (Ho et al., 2020; Rombach et al., 2022; Lipman et al., 2022; Peebles & Xie, 2023; Wu et al., 2025). These developments have stimulated growing interest in conditioning mechanisms and controllable generation, particularly in regional image editing, which aims to modify regions specified by the user while preserving the remaining content (Hertz et al., 2022; Brooks et al., 2023; Zhang et al., 2026).

Existing approaches to regional image editing can be broadly categorized according to how the editable region is specified. Instruction-based methods identify the target content from textual instructions, whereas mask-reference-based methods use explicit masks as spatial references. Benefiting from advances in large language models and vision language models (Radford et al., 2021; Achiam et al., 2023; Team et al., 2023; Bai et al., 2025), instruction-based image editors have demonstrated strong semantic understanding and editing capabilities (Koh et al., 2023; Zhou et al., 2025; Huang et al., 2024; Xu et al., 2025), especially when the target can be described unambiguously. However, as illustrated in Figure 1(a,b), specifying an exact target through language becomes cumbersome in complex scenes such as infographic images (Peng et al., 2025; Tang et al., 2026). These images often contain repeated or visually similar elements, dense layouts, or targets whose locations are difficult to describe precisely. Such spatial ambiguity can substantially reduce localization accuracy. Moreover, instruction-based methods commonly predict noise or vector fields over the entire spatial representation without explicit regional constraints (Wu et al., 2025; Labs, 2025; Deng et al., 2025). Consequently, they may unintentionally alter background content that should remain unchanged.

Mask-reference-based methods alleviate spatial ambiguity by allowing users to specify the editable region explicitly (Avrahami et al., 2023; Wang et al., 2023; Xie et al., 2023; Zhuang et al., 2024; Zhou et al., 2026). Given a source image and a mask that identifies the editable region, existing methods typically encode the mask as an additional image condition and process it in a manner similar to the source image. Common strategies introduce mask information through shared attention layers (Vaswani et al., 2017), additional conditioning branches (Zhang et al., 2023; Mou et al., 2024), or parameter efficient fine-tuning (Hu et al., 2022; Zhang et al., 2026). Despite their improved localization capability, the mask-reference-based methods continue to face challenges in edit fidelity and image integration. Generating the masked region independently can lead to semantic or stylistic drift (Rombach et al., 2022; Xie et al., 2023), which reduces the reliability of the edited content. In addition, the foreground and background frequently follow different generation or preservation processes (Lugmayr et al., 2022; Zhou et al., 2026). Differences between their trajectories can manifest as inconsistencies in color, texture, and local gradients. These inconsistencies may create visible seams along the mask boundary and produce suboptimal results, as shown in Figure 1(c). Taken together, these limitations call for a regional editing framework that provides precise localization, consistent background preservation, and seamless boundary transitions.

To address these challenges, we propose **MaskFlow**, a training framework for regional image editing. Given a source image, an editing instruction, and a mask of arbitrary shape, MaskFlow incorporates the mask into the probability path and flow-matching objective, explicitly modeling generation within the editable region and source preservation outside it. A Soft-Poisson de-seaming module applied during both training and sampling further refines the vector field to produce a smooth boundary.

Specifically, the proposed MaskFlow solves the problems through the following designs. **(1)** We construct editing instructions that omit explicit position descriptions, thereby encouraging the model to identify editable regions from the masks and improving localization precision. **(2)** We introduce probability paths guided by masks and a mask-aware optimization objective that constrains editable

regions throughout denoising. This formulation supports masks with arbitrary shapes and improves regional control in intricate scenes such as infographic images while consistently preserving the background. **(3)** We design a Soft-Poisson de-seaming module that suppresses visible discontinuities between the foreground and background using a Soft-Poisson objective to refine the predicted vector field. This procedure promotes smooth and consistent transitions in color and gradients near the boundary. As demonstrated in Figure 1(a), MaskFlow accurately replaces all text instances specified by the masks. Figure 1(b) further shows that semantically related background content remains unchanged. The results in Figure 1(c) illustrates more natural transitions than methods that directly replace masked regions.

In summary, this work addresses localization, background preservation, and seamless transitions in regional image editing. Our contributions are summarized below.

- We propose **MaskFlow**, a training framework for regional image editors. The probability path guided by the mask and the vector field refinement module jointly enable precise localization, consistent background preservation, and seamless boundary transitions.
- We construct MEData, a dataset tailored for mask-guided image editing, using the proposed data synthesis pipeline. It contains paired source images, target images, prompts, and masks of arbitrary shapes from natural scenes and challenging infographic images for training and evaluating regional editing methods.
- Extensive qualitative and quantitative experiments demonstrate that the proposed approach performs the requested edits correctly while improving localization accuracy, background consistency, and boundary alignment.

2 Related Work

Diffusion models (Ho et al., 2020; Rombach et al., 2022) and flow-matching models (Lipman et al., 2022; Liu et al., 2022), together with scalable transformer backbones (Peebles & Xie, 2023; Ma et al., 2024), provide the generative foundation for regional edit models. A model must accurately localize, keep the unmasked area unchanged, and integrate the generated foreground with its surrounding context. Existing work mainly addresses these requirements through mask-reference-based editing and refinement at region boundaries.

Mask-reference-based regional editing. Masks provide explicit spatial information that is difficult to express through language alone. Early diffusion inpainting methods either blend generated and source latents during denoising or condition the model on masked images, object shapes, and high resolution source content (Avrahami et al., 2023; Wang et al., 2023; Xie et al., 2023). PowerPaint (Zhuang et al., 2024) further uses task prompts to unify filling, insertion, removal, and shape guided inpainting. Recent generalist editors (Mao et al., 2025; Li et al., 2026) encode masks within long context multimodal inputs or combine semantic target recognition with dedicated inpainting branches. Task specific methods (Jiang et al., 2025; Wasserman et al., 2025) improve removal or insertion through guidance from masked regions and purpose built training pairs. RegionE (Chen et al., 2025) and SpotEdit (Qin et al., 2026) instead infer editable or stable tokens from intermediate features and selectively reuse background features. Despite their different implementations, these approaches generally treat the mask or estimated region as a conditioning signal rather than a constraint on the underlying probability path. In contrast, **MaskFlow** incorporates the user mask into both the probability path and the flow-matching objective. This design directly constrains edits with masks of arbitrary shapes while maintaining the source trajectory outside the mask.

Blending and refinement at boundaries. Classical Poisson editing (Pérez et al., 2003) and its modified formulation (Tanaka et al., 2012) reconstruct a pasted region by matching source gradients under destination boundary conditions. Deep Image Blending (Zhang et al., 2020) augments this principle with learned content and style objectives. These methods can suppress abrupt pixel discontinuities, but they operate on a completed composite and may introduce color propagation or fail to repair semantic and structural incompatibility created during generation. Recent approaches move closer to generative models. Long et al. (2025) uses trajectory divergence and feature injection to localize shape changes, whereas RefineAnything (Zhou et al., 2026) and PixPerfect (Zheng et al., 2026) learn refiners that operate at boundaries or in pixel space. However, automatically inferred

regions do not provide exact user control. Auxiliary losses and subsequent refinement also cannot directly correct intermediate generative decisions that produce an incompatible foreground. The proposed **MaskFlow** instead applies soft-Poisson de-seaming at every sampling step to refine the predicted vector field, allowing boundary consistency to influence the entire generation trajectory.

3 PRELIMINARIES

Flow-matching. Let the Gaussian noise distribution be $p_0 = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and the target data distribution be $p_1 = p_{\text{data}}$. Flow-matching models learn a vector field $\boldsymbol{v_\theta}$ that transports samples from p_0 to p_1 through a multistep ODE solver (Lipman et al., 2022). Let $\epsilon \sim p_0$ and $\mathbf{x}_1 \sim p_1$, and consider time-dependent schedules $\alpha(t), \beta(t) \in [0, 1]$ for $t \in [0, 1]$. The probability path is defined as $\mathbf{x}(t) = \alpha(t)\mathbf{x}_1 + \beta(t)\epsilon$, where $\alpha(0) = 0$, $\beta(0) = 1$, $\alpha(1) = 1$, and $\beta(1) = 0$. A commonly used schedule is $\alpha(t) = 1 - \sigma(t)$ and $\beta(t) = \sigma(t)$, where $\sigma(t) \in [0, 1]$ is a monotonically decreasing noise schedule with $\sigma(0) = 1$ and $\sigma(1) = 0$ (Liu et al., 2022). The target vector field is $\boldsymbol{u}(t) = \dot{\alpha}(t)\mathbf{x}_1 + \dot{\beta}(t)\epsilon = \dot{\sigma}(t)(\epsilon - \mathbf{x}_1)$, with $\mathbf{x}(0) = \epsilon$ and $\mathbf{x}(1) = \mathbf{x}_1$. Flow-matching trains a model to predict $\boldsymbol{v_\theta}(\mathbf{x}(t), \sigma(t))$ from the sample $\mathbf{x}(t)$ at time t using the objective

$$\mathcal{L}_{\text{CFM}}(\boldsymbol{\theta}) = \mathbb{E}\left[\|\boldsymbol{v_\theta}(\mathbf{x}(t), \sigma(t)) - \dot{\sigma}(t)(\epsilon - \mathbf{x}_1)\|_2^2\right]. \tag{1}$$

At time t, the clean target can be estimated from the predicted vector field as

$$\widehat{\mathbf{x}}_1(t) = \mathbf{x}(t) - \frac{\sigma(t)}{\dot{\sigma}(t)}\boldsymbol{v_\theta}(\mathbf{x}(t), \sigma(t)). \tag{2}$$

During sampling, we use $\boldsymbol{x}_i$ to denote the realized latent state at the i-th discretization point. Starting from $\boldsymbol{x}_0 \sim p_0$, the generated image is obtained by integrating the ODE forward for $T \in \mathbb{N}^+$ steps $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \frac{\sigma_{i+1} - \sigma_i}{\dot{\sigma}_i}\boldsymbol{v_\theta}(\boldsymbol{x}_i, \sigma_i)$, where $\boldsymbol{x}_i$ is a realization of $\mathbf{x}(i/T)$ and $\sigma_i = \sigma(i/T)$. The index i increases sequentially from 0 to $T - 1$, and $\boldsymbol{x}_T$ is the generated sample.

Poisson image editing. Poisson image editing (Pérez et al., 2003) seamlessly clones a foreground region into a background image by aligning image gradients. Let $\mathbb{S} \subset \mathbb{R}^2$ be the spatial domain of one image channel, and let $\Omega \subset \mathbb{S}$ denote a closed region with boundary $\partial\Omega$. The foreground image f is defined over Ω, and the background image g is defined over $\mathbb{S}\backslash\Omega$. Poisson image editing seeks a scalar field $h^\star$ over Ω that satisfies

$$h^\star = \arg\min_h \int_\Omega \|\nabla_{\boldsymbol{p}} h(\boldsymbol{p}) - \nabla_{\boldsymbol{p}} f(\boldsymbol{p})\|_2^2 \, \mathrm{d}\boldsymbol{p} \quad \text{s.t.} \quad h|_{\partial\Omega} = g|_{\partial\Omega}. \tag{3}$$

Applying the Euler–Lagrange equation yields the Poisson equation with Dirichlet boundary conditions

$$\Delta_{\boldsymbol{p}} h^\star(\boldsymbol{p}) = \Delta_{\boldsymbol{p}} f(\boldsymbol{p}) \quad \text{s.t.} \quad h^\star|_{\partial\Omega} = g|_{\partial\Omega}, \tag{4}$$

where $\Delta_{\boldsymbol{p}} = \nabla_{\boldsymbol{p}} \cdot \nabla_{\boldsymbol{p}}$ is the Laplacian operator over the spatial coordinate $\boldsymbol{p}$. Equation 4 aligns the synthesized image $h^\star$ with the foreground image f over Ω while maintaining color continuity along the boundary. It can be solved efficiently using Jacobi iteration.

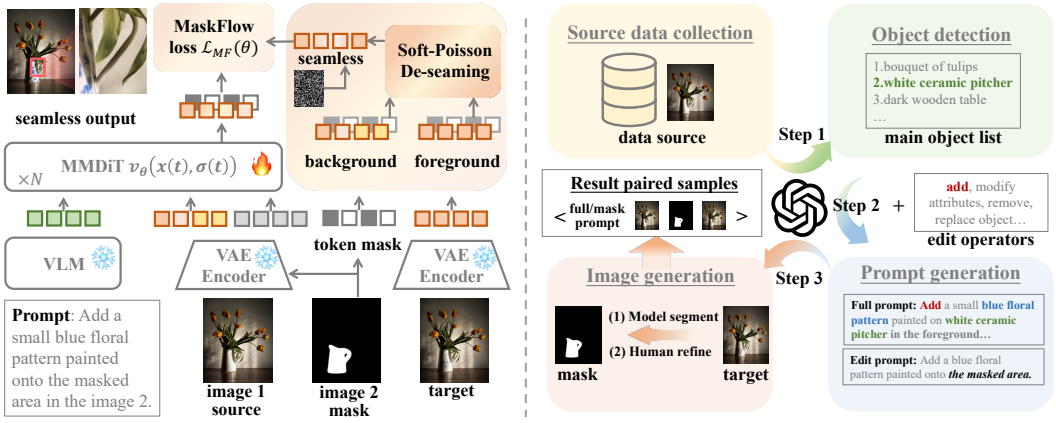
4 METHODOLOGY

We propose **MaskFlow** for regional image editing with precise localization, consistent background preservation and seamless boundary transitions. Section 4.1 introduces the training framework for precise localization and consistent background preservation. Section 4.2 presents the refinement procedure for seamless boundary transitions. Section 4.3 describes the data synthesis pipeline.

4.1 MASKFLOW

MaskFlow incorporates the mask into the probability path. Let $\boldsymbol{x}_S, \boldsymbol{x}_1 \in \mathbb{R}^d$ be vectorized source and target latents, where $d = CHW$. Let $\boldsymbol{M} \in \{0, 1\}^{H \times W}$ be a binary mask whose entry $M_{i,j} = 1$ denotes an editable pixel. We use $\boldsymbol{m} \in \{0, 1\}^d$ to denote the channel-wise broadcast and vectorization of $\boldsymbol{M}$. The proposed probability path is

$$\boldsymbol{x}(t) = \boldsymbol{m} \odot (\alpha(t)\boldsymbol{x}_1 + \beta(t)\epsilon) + (\mathbf{1} - \boldsymbol{m}) \odot \widetilde{\boldsymbol{x}}(t), \tag{5}$$



(a) The regional editing training pipeline of MaskFlow. **(b) The paired regional editing data synthesis pipeline.**

Figure 2: Overview of the proposed **MaskFlow** framework and data synthesis pipeline. **(a)** During training, the source image and mask are provided to the MMDiT for vector field prediction. Soft-Poisson de-seaming integrates the generated foreground with the source background. **(b)** The three-stage data synthesis pipeline uses a vision language model and an image generation model to produce tuples in the form <prompt,source,mask,target>.

where the first term generates content inside the masked region, whereas the unmasked component $\widetilde{\boldsymbol{x}}(t)$ preserves the background. In practice, we use $\widetilde{\boldsymbol{x}}(t) = \alpha(t)\boldsymbol{x}_S + \beta(t)\epsilon$, so that the unmasked region follows the source image rather than the target image.

The training objective must account for masks with different shapes and sizes. We define the masked area ratio as $a(\boldsymbol{M}) = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W} M_{i,j}$. We then introduce an adaptive mask weight into Equation 1 to balance supervision from regions of different sizes

$$\mathcal{L}_{\mathrm{MF}}(\boldsymbol{\theta}) = \mathbb{E}\left[\left\|\boldsymbol{m} \odot \frac{\omega_{\mathrm{mask}}}{a(\boldsymbol{M})}\left(\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}(t), \sigma(t) \mid \boldsymbol{x}_S, \boldsymbol{M}) - \dot{\boldsymbol{x}}(t)\right)\right\|_2^2\right], \tag{6}$$

where $\omega_{\mathrm{mask}} > 0$ is the loss weight. We assume that each training mask contains at least one editable pixel, so $a(\boldsymbol{M}) > 0$. This objective balances the contributions of masks that cover different areas, and is utilized as the supervision for model training.

4.2 Soft-Poisson De-seaming

We propose **Soft-Poisson de-seaming**, which refines the predicted vector field at every sampling step through a unified gradient-domain objective. Given a predicted vector field $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}(t), \sigma(t))$, we first estimate the clean target $\widehat{\mathbf{x}}_1$ using Equation 2 and then reshape the source and estimated target latents into feature maps. Let $\boldsymbol{x}_S(\boldsymbol{p}), \widehat{\boldsymbol{x}}_1(\boldsymbol{p}) \in \mathbb{R}^C$ denote their feature vectors at spatial position $\boldsymbol{p}$, and let $\boldsymbol{z}(\boldsymbol{p}) \in \mathbb{R}^C$ be the refined spatial field. All differential operators below act on $\boldsymbol{p}$ and are applied componentwise.

Starting from the binary mask $\boldsymbol{M}$, we use a finite-support Gaussian kernel to obtain transition weights while retaining a value of 1 inside the original editable region. The resulting soft mask $\widetilde{\boldsymbol{M}}$ defines an expanded region $\widetilde{\Omega} = \left\{\boldsymbol{p} = (i,j) \in \{1,\dots,H\} \times \{1,\dots,W\} \;\middle|\; \widetilde{M}_{i,j} > 0\right\}$. Let $\widetilde{m}(\boldsymbol{p}) = \widetilde{M}_{i,j}$ for $\boldsymbol{p} = (i,j)$. Larger values of $\widetilde{m}(\boldsymbol{p})$ identify positions that should remain close to the generated edit, whereas smaller values gradually increase the influence of the source image toward the exterior of $\widetilde{\Omega}$.

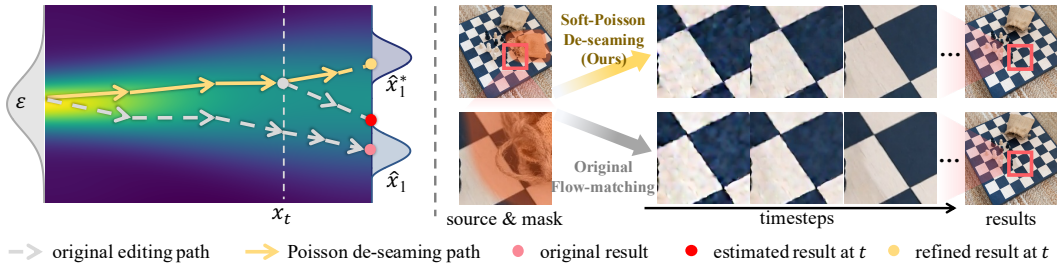


Figure 3: Probability paths for Soft-Poisson de-seaming in yellow and conventional flow-matching in gray. **Left.** The sampling paths move from noise ϵ toward an estimate with seams $\widehat{\boldsymbol{x}}_1$ or a seamless estimate $\widehat{\boldsymbol{x}}_1^\star$. Soft-Poisson de-seaming corrects the vector field at time t and moves the sample toward the seamless distribution. **Right.** A comparison of the two processes, where the upper result from our method contains fewer boundary artifacts.

We obtain the refined field by solving

$$\begin{aligned}\boldsymbol{z}^\star = \arg\min_{\boldsymbol{z}} \; & \int_{\widetilde{\Omega}} \|\nabla_{\boldsymbol{p}} \boldsymbol{z}(\boldsymbol{p}) - \nabla_{\boldsymbol{p}} \widehat{\boldsymbol{x}}_1(\boldsymbol{p})\|_F^2 \, \mathrm{d}\boldsymbol{p} \\ & + \lambda_{\mathrm{e}} \int_{\widetilde{\Omega}} \widetilde{m}(\boldsymbol{p}) \, \|\boldsymbol{z}(\boldsymbol{p}) - \widehat{\boldsymbol{x}}_1(\boldsymbol{p})\|_2^2 \, \mathrm{d}\boldsymbol{p} \qquad \text{s.t.} \quad \boldsymbol{z}|_{\partial\widetilde{\Omega}} = \boldsymbol{x}_S|_{\partial\widetilde{\Omega}}, \\ & + \lambda_{\mathrm{s}} \int_{\widetilde{\Omega}} (1 - \widetilde{m}(\boldsymbol{p})) \, \|\boldsymbol{z}(\boldsymbol{p}) - \boldsymbol{x}_S(\boldsymbol{p})\|_2^2 \, \mathrm{d}\boldsymbol{p}\end{aligned} \tag{7}$$

where $\lambda_{\mathrm{e}}, \lambda_{\mathrm{s}} > 0$. The first term transfers the spatial gradients of the estimated target to preserve the generated structure. The second term anchors the refined field to the estimated edit in regions with large soft-mask values. The third term progressively restores the source feature as the soft-mask value decreases. Together with the boundary condition, these terms form a continuous transition from the generated foreground to the preserved background within a single objective.

Applying the Euler–Lagrange equation to Equation 7 yields

$$\Delta_{\boldsymbol{p}} \boldsymbol{z}(\boldsymbol{p}) - \Delta_{\boldsymbol{p}} \widehat{\boldsymbol{x}}_1(\boldsymbol{p}) = \lambda_{\mathrm{e}} \widetilde{m}(\boldsymbol{p}) \, (\boldsymbol{z}(\boldsymbol{p}) - \widehat{\boldsymbol{x}}_1(\boldsymbol{p})) + \lambda_{\mathrm{s}} \, (1 - \widetilde{m}(\boldsymbol{p})) \, (\boldsymbol{z}(\boldsymbol{p}) - \boldsymbol{x}_S(\boldsymbol{p})) \,, \tag{8}$$

which holds for every $\boldsymbol{p} \in \widetilde{\Omega}$ under the boundary condition in Equation 7. Here, $\Delta_{\boldsymbol{p}} = \nabla_{\boldsymbol{p}} \cdot \nabla_{\boldsymbol{p}}$ is the spatial Laplacian, and solving this Poisson equation we can get the refined spatial field $\boldsymbol{z}^\star$.

Combining $\boldsymbol{z}^\star$ inside $\widetilde{\Omega}$ with $\boldsymbol{x}_S$ outside it forms a complete refined feature map $\widehat{\mathbf{X}}_1^\star(t) \in \mathbb{R}^{C \times H \times W}$. Vectorizing this feature map gives the seamless target estimate $\widehat{\boldsymbol{x}}_1^\star(t) = \mathrm{vec}(\widehat{\mathbf{X}}_1^\star(t)) \in \mathbb{R}^{CHW}$. We use this estimate to redirect the current sampling trajectory by replacing the original vector field with $\boldsymbol{v}_{\boldsymbol{\theta}}^\star(\boldsymbol{x}(t), \sigma(t)) = \frac{\dot{\sigma}(t)}{\sigma(t)} \, (\boldsymbol{x}(t) - \widehat{\boldsymbol{x}}_1^\star(t))$. The next latent state is computed as $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \frac{\sigma_{i+1} - \sigma_i}{\dot{\sigma}_i} \boldsymbol{v}_{\boldsymbol{\theta}}^\star(\boldsymbol{x}_i, \sigma_i)$. For $i = 0, \ldots, T-1$, repeating this refinement before each sampling update continuously steers the denoising path toward a target that is compatible with both the generated edit and the source context.

In practice, we discretize Equation 8 for efficient numerical computation. Let $\mathrm{N}(\boldsymbol{p})$ be the set of neighboring pixels of $\boldsymbol{p} \in \widetilde{\Omega}$, and let $\widetilde{m}_{\boldsymbol{p}}$ denote the corresponding soft-mask value. Define $J_{\boldsymbol{p}} = |\mathrm{N}(\boldsymbol{p})| + \lambda_{\mathrm{e}} \widetilde{m}_{\boldsymbol{p}} + \lambda_{\mathrm{s}} \, (1 - \widetilde{m}_{\boldsymbol{p}})$. The discrete equation using Jacobi iteration is

$$\begin{aligned}\boldsymbol{z}_{\boldsymbol{p}}^{(k+1)} = \frac{1}{J_{\boldsymbol{p}}} \Bigg[& \sum_{\boldsymbol{q} \in \mathrm{N}(\boldsymbol{p}) \cap \widetilde{\Omega}} \boldsymbol{z}_{\boldsymbol{q}}^{(k)} + \sum_{\boldsymbol{q} \in \mathrm{N}(\boldsymbol{p}) \setminus \widetilde{\Omega}} \boldsymbol{x}_{S,\boldsymbol{q}} + \sum_{\boldsymbol{q} \in \mathrm{N}(\boldsymbol{p})} (\widehat{\boldsymbol{x}}_{1,\boldsymbol{p}} - \widehat{\boldsymbol{x}}_{1,\boldsymbol{q}}) \\ & + \lambda_{\mathrm{e}} \widetilde{m}_{\boldsymbol{p}} \widehat{\boldsymbol{x}}_{1,\boldsymbol{p}} + \lambda_{\mathrm{s}} \, (1 - \widetilde{m}_{\boldsymbol{p}}) \, \boldsymbol{x}_{S,\boldsymbol{p}} \Bigg].\end{aligned} \tag{9}$$

The complete derivation is provided in supplementary Section A. After K iterations, the refined clean feature is approximated by $\widehat{\boldsymbol{x}}_{1,\boldsymbol{p}}^\star \approx \boldsymbol{z}_{\boldsymbol{p}}^{(K)}$ for every $\boldsymbol{p} \in \widetilde{\Omega}$. Experiments and visualizations demonstrate smoother boundary transitions.

Table 1: Editing accuracy and background preservation on MEData. Bold and underlined values indicate the best and second-ranked results, respectively.

	Global					Background	
Method	CLIP ↑	DINO ↑	FID ↓	PSNR ↑	SSIM ↑	MSE ↓	LPIPS ↓
Gemini 3 Flash Image	0.9724	**0.9537**	<u>20.38</u>	<u>22.00</u>	0.7450	0.0023	0.0505
GPT Image 2	<u>0.9734</u>	0.9367	24.12	16.79	0.5069	0.0181	0.2064
BAGEL-7B-MoT	0.9464	0.9055	27.74	20.80	0.7560	0.0030	0.0648
FLUX.2-dev	0.9555	0.9138	29.99	19.48	0.6644	0.0063	0.0781
HiDream-O1-Image	0.9146	0.8435	45.34	13.17	0.3613	0.0450	0.4313
QwenImage-Edit-2511	0.9492	0.9107	29.85	19.11	0.6412	0.0088	0.1269
RefineAnything	0.9621	0.9057	26.41	21.29	<u>0.7740</u>	<u>0.0000</u>	0.0007
RegionE	0.9672	0.9313	25.83	21.03	0.7399	0.0031	0.0751
SpotEdit	0.9332	0.8924	56.49	10.89	0.3001	0.0844	0.5083
MaskFlow	**0.9782**	<u>0.9532</u>	**19.90**	**22.60**	**0.7846**	<u>0.0000</u>	**0.0000**

4.3 Data pipeline

To improve localization precision using the specified masks, we design a simple and efficient data synthesis pipeline that generates paired source images, masks, prompts, and target images. The resulting **MEData** dataset contains approximately 10K pairs from natural scenes and infographics.

As shown in Figure 2(b), the pipeline contains object detection, prompt generation, and image generation. Given source images containing multiple concepts and potentially editable objects, **object detection** identifies the salient objects in each image. As there may be many visually insignificant objects whose editing provides limited value for training, we prompt a vision language model to identify the primary objects in each source image. The **prompt generation** stage produces two instructions for regional editing. The first is a complete instruction that specifies the editing operation, target position, and desired result. The second replaces the explicit position description with a demonstrative expression, encouraging the model to obtain localization information from the mask during training. The final **image generation** stage synthesizes target images and constructs their masks. We use the complete instructions with detailed positions to generate the target images. SAM (Kirillov et al., 2023) then provides an initial segmentation of each target region and then the human annotators refine these regions and create masks of arbitrary shapes that resemble practical user inputs.

5 Experiments

5.1 Experimental settings

Implementation details. We use the QwenImage-Edit-2511 model (Wu et al., 2025) and train attention LoRA modules (Hu et al., 2022; Vaswani et al., 2017). We train the model for 5K steps with rank 256 using the Prodigy optimizer (Mishchenko & Defazio, 2023). Equations 5 and 6 provide regional control during training. We solve the seamless refinement in Equation 9 using 50 Jacobi iterations by default. During inference, we use 50 sampling steps and a classifier-free guidance scale of 4.0 (Ho & Salimans, 2022). The model supports images with varying resolutions and aspect ratios.

Evaluation metrics. We evaluate overall editing quality and background preservation on the MEData benchmark. CLIP similarity (Radford et al., 2021) measures semantic alignment between the edited results and reference images, while DINO similarity (Oquab et al., 2023) measures structural consistency. We also report FID (Heusel et al., 2017), PSNR (Korhonen & You, 2012), and SSIM (Wang et al., 2004) to assess image fidelity. For the ablation studies, we additionally report global LPIPS (Zhang et al., 2018) and VGG (Simonyan & Zisserman, 2014) perceptual distance.

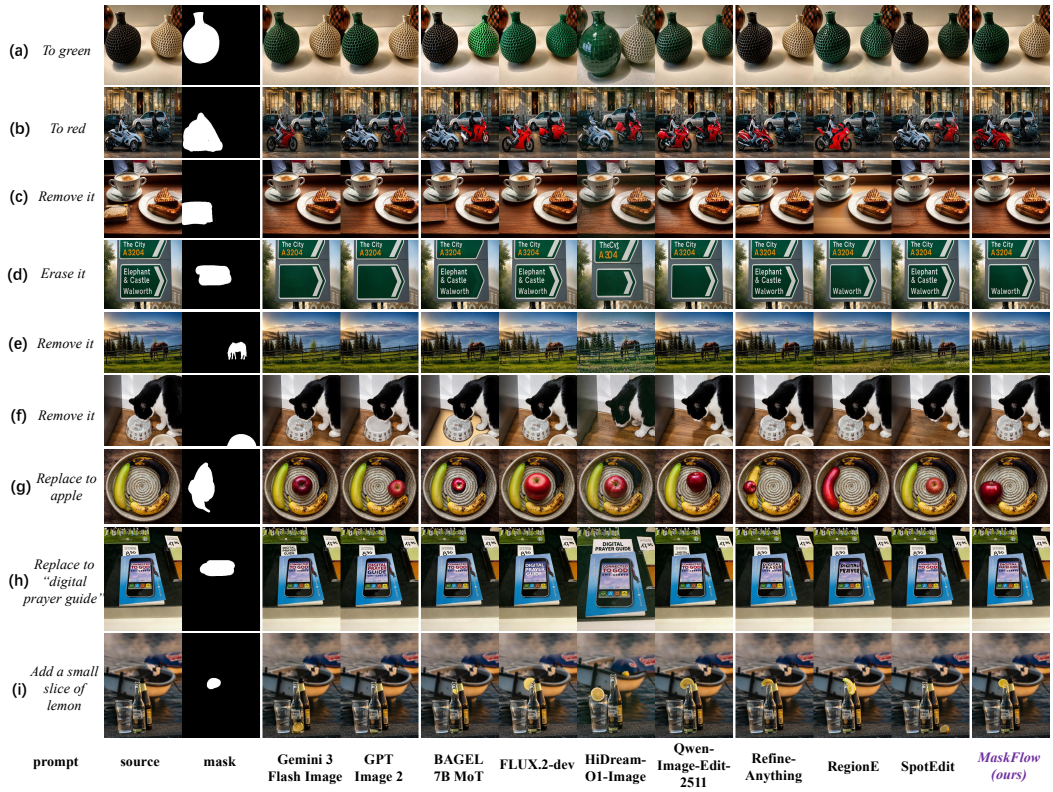


Figure 4: Qualitative comparison with commercial models, open sourced general image editing models, and mask-reference-based methods. Several baselines edit content outside the specified masks, as shown in **(a), (b), (g), (h), and (i)**, alter unrelated background content, as shown in **(a), (d), (e), and (f)**, or introduce visible boundary seams, as shown in **(c), (d), and (f)**. **MaskFlow** improves localization, background preservation, and boundary transitions. Please zoom in for details.

Background preservation is evaluated using mean squared error and LPIPS computed only over the unmasked regions.

5.2 Image Editing Comparisons

We compare MaskFlow with commercial models (Achiam et al., 2023; Team et al., 2023), open sourced general image editing models (Deng et al., 2025; Labs, 2025; Cai et al., 2026; Wu et al., 2025), and methods designed for regional editing (Zhou et al., 2026; Chen et al., 2025; Qin et al., 2026) through qualitative and quantitative experiments.

Qualitative analysis. Figure 4 presents the qualitative results. In Figure 4(b), several baselines fail to localize the edit or preserve the target identity despite receiving an explicit mask. In contrast, MaskFlow follows both the mask and the editing instruction. The example in Figure 4(h) further reveals background changes in competing methods, where text outside the target region is altered. MaskFlow replaces only the text inside the specified regions while preserving the remaining content. In the object removal example in Figure 4(f), BAGEL (Deng et al., 2025) does not follow the instruction, while QwenImage (Wu et al., 2025) edits the wrong location. Gemini (Team et al., 2023) and RefineAnything (Zhou et al., 2026) localize the removal correctly but leave visible seams between the edited foreground and preserved background. MaskFlow accurately localizes the removal and produces a smoother boundary transition. Overall, these qualitative comparisons show that MaskFlow performs the requested edits within the target regions, preserves surrounding content, and integrates the edited foreground more naturally with the background.

Table 2: Ablation study of the proposed modules. "MF" denotes MaskFlow, and "SPD" denotes soft-Poisson de-seaming.

MF	SPD	FID ↓	CLIP ↑	DINO ↑	LPIPS ↓	PSNR ↑	SSIM ↑	VGG ↓
-	-	29.85	0.9492	0.9106	0.2070	19.11	0.6412	7.269
✓	-	20.51	0.9761	0.9505	0.1074	22.38	0.7828	3.991
✓	✓	**19.90**	**0.9782**	**0.9532**	**0.1047**	**22.60**	**0.7846**	**3.952**

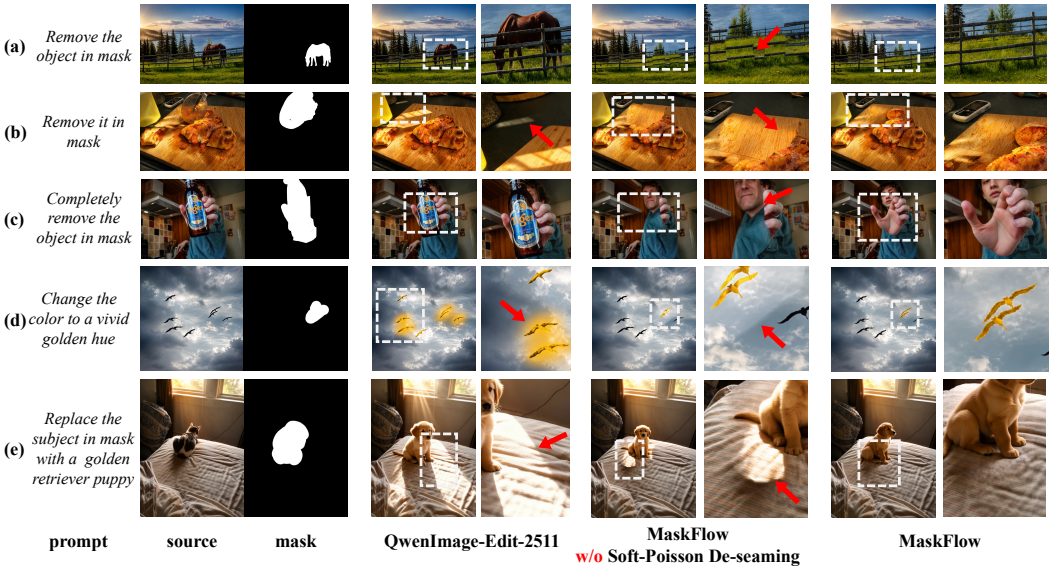


Figure 5: Qualitative ablation of the proposed modules. In each group, the left image shows the full result and the right image shows an enlarged view. The red arrows point out the wrong regions. Zoom in for better visualization.

Quantitative analysis. Table 1 reports the quantitative comparisons between our method and baselines. MaskFlow achieves the best CLIP, FID, PSNR, and SSIM scores for global image evaluations and the best background LPIPS. These results support the qualitative findings and show that MaskFlow improves editing accuracy without sacrificing the unedited background.

5.3 Ablation Studies

Effectiveness of the proposed modules. We analyze the contribution of each proposed module. As illustrated in Figure 5, the base model exhibits poor localization and background consistency given regional masks and editing instructions. MaskFlow without de-seaming refinement constrains the editable regions and preserves background details. However, visible seams may remain because the edited foreground and preserved background follow different trajectories. Adding soft-Poisson de-seaming improves visual continuity and produces smoother transitions near the boundary. The quantitative results in Table 2 support these observations. MaskFlow substantially improves all reported metrics over the base model. Soft-Poisson de-seaming provides consistent additional gains, reducing FID from 20.51 to 19.90 and improving PSNR from 22.38 to 22.60. This result confirms that the two components are complementary, with MaskFlow providing regional control and soft-Poisson de-seaming improving boundary integration.

Localization precision. This experiment studies how *additional* position descriptions in the prompts affect localization. In both settings, the model receives a prompt and a mask. By default, the proposed method is **not** trained with additional textual position descriptions, only extracting location features from the masks. To evaluate this design choice, we train another model using additional position descriptions. Table 3 presents the results. When an explicit mask is available, omitting

Table 3: Effect of *additional* textual position descriptions on regional image editing given the masks.

Additional Position	FID ↓	CLIP ↑	DINO ↑	LPIPS ↓	PSNR ↑	SSIM ↑	VGG ↓
✓	29.49	0.9472	0.8750	0.1227	21.15	0.7726	4.128
✗ (Ours)	**17.21**	**0.9795**	**0.9563**	**0.1036**	**22.74**	**0.7852**	**3.937**



Figure 6: Application to infographic editing compared with the base model and a standard fine-tuning baseline. The red arrows point out the wrong edit regions. Please zoom in for details.

redundant position descriptions encourages the model to rely more strongly on the spatial information in the mask. This strategy consistently improves the performance, reducing FID from 29.49 to 17.21. The results show that removing redundant textual position cues during training improves mask-based localization precision.

5.4 Infographic Application

Infographics combine visual and textual elements for applications such as commercial promotion and poster design. They often contain complex layouts, making it difficult to specify precise locations using language alone. Figure 6 shows that both the base model and the standard fine-tuning baseline fail to localize some editable regions and substantially alter background text. In contrast, MaskFlow uses masks of arbitrary shapes to constrain the editable regions and produces reliable visual and textual edits while preserving surrounding content. These results demonstrate the effectiveness and practical potential of MaskFlow for infographic editing.

6 CONCLUSION

This paper presents **MaskFlow**, a training framework for regional image editing with precise localization, consistent background preservation, and seamless boundary transitions. MaskFlow incorporates the mask into the flow-matching probability path and uses a mask-aware objective to improve localization accuracy and preserve unmasked content. Training on the prompts without additional position descriptions pushes the model to learn localization from masks, further strengthening the attentions to editable regions. The proposed Soft-Poisson de-seaming module refines the vector field during sampling to produce smooth transitions between edited and preserved regions. Moreover, the constructed MEData provides a new regional editing dataset for further study. Quantitative and qualitative experiments demonstrate the effectiveness of MaskFlow on both natural scenes and infographic images, showing promise for practical application.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM transactions on graphics (TOG)*, 42(4):1–11, 2023.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

Qi Cai, Jingwen Chen, Chengmin Gao, Zijian Gong, Yehao Li, Yingwei Pan, Yi Peng, Zhaofan Qiu, Kai Yu, Yiheng Zhang, et al. Hidream-o1-image: A natively unified image generative foundation model with pixel-level unified transformer. *arXiv preprint arXiv:2605.11061*, 2026.

Pengtao Chen, Xianfang Zeng, Maosen Zhao, Mingzhu Shen, Peng Ye, Bangyin Xiang, Zhibo Wang, Wei Cheng, Gang Yu, and Tao Chen. Regione: Adaptive region-aware generation for efficient image editing. *arXiv preprint arXiv:2510.25590*, 2025.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8362–8371, 2024.

Longtao Jiang, Zhendong Wang, Jianmin Bao, Wengang Zhou, Dongdong Chen, Lei Shi, Dong Chen, and Houqiang Li. Smarteraser: Remove anything from images using masked-region guidance. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24452–24462, 2025.

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 *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Jing Yu Koh, Daniel Fried, and Russ R Salakhutdinov. Generating images with multimodal language models. *Advances in Neural Information Processing Systems*, 36:21487–21506, 2023.

Jari Korhonen and Junyong You. Peak signal-to-noise ratio revisited: Is simple beautiful? In *2012 Fourth International Workshop on Quality of Multimedia Experience*, pp. 37–38, 2012. doi: 10.1109/QoMEX.2012.6263880.

Black Forest Labs. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Yaowei Li, Yuxuan Bian, Xuan Ju, Zhaoyang Zhang, Junhao Zhuang, Ying Shan, Yuexian Zou, and Qiang Xu. Brushedit: All-in-one image inpainting and editing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Zeqian Long, Mingzhe Zheng, Kunyu Feng, Xinhua Zhang, Hongyu Liu, Harry Yang, Linfeng Zhang, Qifeng Chen, and Yue Ma. Follow-your-shape: Shape-aware image editing via trajectory-guided region control. *arXiv preprint arXiv:2508.08134*, 2025.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11451–11461. IEEE, 2022.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Chaojie Mao, Jingfeng Zhang, Yulin Pan, Zeyinzi Jiang, Zhen Han, Yu Liu, and Jingren Zhou. Ace++: Instruction-based image creation and editing via context-aware content filling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1958–1966, 2025.

Konstantin Mishchenko and Aaron Defazio. Prodigy: An expeditiously adaptive parameter-free learner. *arXiv preprint arXiv:2306.06101*, 2023.

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. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 4296–4304, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Yuyang Peng, Shishi Xiao, Keming Wu, Qisheng Liao, Bohan Chen, Kevin Lin, Danqing Huang, Ji Li, and Yuhui Yuan. Bizgen: Advancing article-level visual text rendering for infographics generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23615–23624, June 2025.

Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22 (3):313–318, July 2003. ISSN 0730-0301.

Zhibin Qin, Zhenxiong Tan, Zeqing Wang, Songhua Liu, and Xinchao Wang. Spotedit: Selective region editing in diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18683–18692, 2026.

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 *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Masayuki Tanaka, Ryo Kamio, and Masatoshi Okutomi. Seamless image cloning by a closed form solution of a modified poisson problem. In *SIGGRAPH Asia 2012 Posters*, pp. 1–1. 2012.

Yinghao Tang, Xueding Liu, Boyuan Zhang, Tingfeng Lan, Yupeng Xie, Jiale Lao, Yiyao Wang, Haoxuan Li, Tingting Gao, Bo Pan, et al. Igenbench: Benchmarking the reliability of text-to-infographic generation. *arXiv preprint arXiv:2601.04498*, 2026.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J. Fleet, Radu Soricut, Jason Baldridge, Mohammad Norouzi, Peter Anderson, and William Chan. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18359–18369, June 2023.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Navve Wasserman, Noam Rotstein, Roy Ganz, and Ron Kimmel. Paint by inpaint: Learning to add image objects by removing them first. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18313–18324, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22428–22437, 2023.

Yingjing Xu, Jie Kong, Jiazhi Wang, Xiao Pan, Bo Lin, and Qiang Liu. Insightedit: Towards better instruction following for image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2694–2703, 2025.

Lingzhi Zhang, Tarmily Wen, and Jianbo Shi. Deep image blending. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 231–240, 2020.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. Enabling instructional image editing with in-context generation in large scale diffusion transformer. *Advances in Neural Information Processing Systems*, 38:139195–139227, 2026.

Haitian Zheng, Yuan Yao, Yuqian Zhou, Jiebo Luo, Zhe Lin, et al. Pixperfect: Seamless latent diffusion local editing with discriminative pixel-space refinement. *Advances in Neural Information Processing Systems*, 38:128266–128292, 2026.

Dewei Zhou, You Li, Zongxin Yang, and Yi Yang. Refineanything: Multimodal region-specific refinement for perfect local details. *arXiv preprint arXiv:2604.06870*, 2026.

Jun Zhou, Jiahao Li, Zunnan Xu, Hanhui Li, Yiji Cheng, Fa-Ting Hong, Qin Lin, Qinglin Lu, and Xiaodan Liang. Fireedit: Fine-grained instruction-based image editing via region-aware vision language model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13093–13103, 2025.

Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *European Conference on Computer Vision*, pp. 195–211. Springer, 2024.

A SOFT-POISSON DE-SEAMING

A.1 DETAILED FORMULA DERIVATION

Let the VAE-encoded source and target feature maps $\mathbf{X}_S, \mathbf{X}_1$ belong to $\mathbb{R}^{C\times H\times W}$, and let $\boldsymbol{x}_S, \boldsymbol{x}_1 \in \mathbb{R}^{CHW}$ denote their vectorized representations. At time t, the model predicts a vector field $\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}(t), \sigma(t))$. For a spatial coordinate $\boldsymbol{p}$, we use $\boldsymbol{x}_S(\boldsymbol{p}), \widehat{\boldsymbol{x}}_1(\boldsymbol{p}) \in \mathbb{R}^C$ to denote the source feature and estimated target feature at that position.

Let Ω denote the editable region and define the binary mask function as

$$m(\boldsymbol{p}) = \mathbf{1}_{\boldsymbol{p}\in\Omega}. \tag{10}$$

To support seamless editing, we dilate the region to $\widetilde{\Omega} \supset \Omega$. Let $b(\boldsymbol{p}) \in (0, 1)$ be the normalized value produced by a finite-support Gaussian blur in the transition region. The soft mask is

$$\widetilde{m}(\boldsymbol{p}) = \begin{cases} 1, & \boldsymbol{p} \in \Omega, \\ b(\boldsymbol{p}), & \boldsymbol{p} \in \widetilde{\Omega}\backslash\Omega, \\ 0, & \boldsymbol{p} \in \mathbb{S}\backslash\widetilde{\Omega}. \end{cases} \tag{11}$$

The goal of Soft-Poisson de-seaming is to refine the vector field and reduce seams between the masked and unmasked regions.

The core idea is to obtain a corrected vector field $\boldsymbol{v}^{\star}_{\boldsymbol{\theta}}(\boldsymbol{x}(t), \sigma(t))$ from the refined target estimate $\widehat{\boldsymbol{x}}^{\star}_1(t)$. At time t, the target is estimated as

$$\widehat{\boldsymbol{x}}_1(t) = \boldsymbol{x}(t) - \frac{\sigma(t)}{\dot{\sigma}(t)}\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}(t), \sigma(t)). \tag{12}$$

We reshape this estimate into a spatial feature field $\widehat{\boldsymbol{x}}_1(\boldsymbol{p}, t) \in \mathbb{R}^C$. Inspired by Poisson image editing in Equation 4, we optimize the following objective at time t under the boundary constraint $\boldsymbol{z}|_{\partial\widetilde{\Omega}} = \boldsymbol{x}_S|_{\partial\widetilde{\Omega}}$

$$\begin{aligned} \boldsymbol{z}^{\star} = \arg\min_{\boldsymbol{z}} \; & \int_{\widetilde{\Omega}} \|\nabla_{\boldsymbol{p}}\boldsymbol{z}(\boldsymbol{p}) - \nabla_{\boldsymbol{p}}\widehat{\boldsymbol{x}}_1(\boldsymbol{p})\|_F^2 \, \mathrm{d}\boldsymbol{p} \\ & + \lambda_{\mathrm{e}} \int_{\widetilde{\Omega}} \widetilde{m}(\boldsymbol{p}) \, \|\boldsymbol{z}(\boldsymbol{p}) - \widehat{\boldsymbol{x}}_1(\boldsymbol{p})\|_2^2 \, \mathrm{d}\boldsymbol{p} \\ & + \lambda_{\mathrm{s}} \int_{\widetilde{\Omega}} (1 - \widetilde{m}(\boldsymbol{p})) \, \|\boldsymbol{z}(\boldsymbol{p}) - \boldsymbol{x}_S(\boldsymbol{p})\|_2^2 \, \mathrm{d}\boldsymbol{p}, \end{aligned} \tag{13}$$

where $\lambda_{\mathrm{e}}, \lambda_{\mathrm{s}} > 0$. We omit t from the spatial fields for clarity because the optimization has the same form at every time step.

Let $\mathcal{J}(\boldsymbol{z})$ denote the objective function

$$\begin{aligned} \mathcal{J}(\boldsymbol{z}) = \int_{\widetilde{\Omega}} \Big(& \|\nabla_{\boldsymbol{p}}\boldsymbol{z}(\boldsymbol{p}) - \nabla_{\boldsymbol{p}}\widehat{\boldsymbol{x}}_1(\boldsymbol{p})\|_F^2 \\ & + \lambda_{\mathrm{e}}\widetilde{m}(\boldsymbol{p}) \, \|\boldsymbol{z}(\boldsymbol{p}) - \widehat{\boldsymbol{x}}_1(\boldsymbol{p})\|_2^2 \\ & + \lambda_{\mathrm{s}} \, (1 - \widetilde{m}(\boldsymbol{p})) \, \|\boldsymbol{z}(\boldsymbol{p}) - \boldsymbol{x}_S(\boldsymbol{p})\|_2^2 \Big) \, \mathrm{d}\boldsymbol{p}. \end{aligned} \tag{14}$$

We solve this problem using a variational method. Consider a small perturbation $\epsilon\boldsymbol{\eta}(\boldsymbol{p})$, where $\boldsymbol{\eta}(\boldsymbol{p}) \in \mathbb{R}^C$ is an arbitrary differentiable vector field satisfying $\boldsymbol{\eta}|_{\partial\widetilde{\Omega}} = \mathbf{0}$. At an extremum, the first variation must satisfy

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{J}(\boldsymbol{z} + \epsilon\boldsymbol{\eta})\Big|_{\epsilon=0} = 0. \tag{15}$$

Term 1. Define the gradient-alignment term as

$$\mathcal{J}_{\mathrm{g}}(\boldsymbol{z} + \epsilon\boldsymbol{\eta}) = \int_{\widetilde{\Omega}} \|\nabla_{\boldsymbol{p}}(\boldsymbol{z} + \epsilon\boldsymbol{\eta}) - \nabla_{\boldsymbol{p}}\widehat{\boldsymbol{x}}_1\|_F^2 \, \mathrm{d}\boldsymbol{p}. \tag{16}$$

Its derivative at $\epsilon = 0$ is

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{J}_{\mathrm{g}}(\boldsymbol{z}+\epsilon\boldsymbol{\eta})\Big|_{\epsilon=0} = 2\int_{\widetilde{\Omega}} \langle \nabla_{\boldsymbol{p}}\boldsymbol{z} - \nabla_{\boldsymbol{p}}\widehat{\boldsymbol{x}}_1, \nabla_{\boldsymbol{p}}\boldsymbol{\eta}\rangle_F \,\mathrm{d}\boldsymbol{p} \tag{17}$$

$$= 2\int_{\partial\widetilde{\Omega}} \boldsymbol{\eta}^\top (\nabla_{\boldsymbol{p}}\boldsymbol{z} - \nabla_{\boldsymbol{p}}\widehat{\boldsymbol{x}}_1)\,\boldsymbol{n}\,\mathrm{d}s - 2\int_{\widetilde{\Omega}} \boldsymbol{\eta}^\top (\Delta_{\boldsymbol{p}}\boldsymbol{z} - \Delta_{\boldsymbol{p}}\widehat{\boldsymbol{x}}_1)\,\mathrm{d}\boldsymbol{p} \tag{18}$$

$$= -2\int_{\widetilde{\Omega}} \boldsymbol{\eta}^\top (\Delta_{\boldsymbol{p}}\boldsymbol{z} - \Delta_{\boldsymbol{p}}\widehat{\boldsymbol{x}}_1)\,\mathrm{d}\boldsymbol{p}, \tag{19}$$

where the second equality follows from integration by parts, $\boldsymbol{n}$ is the outward unit normal vector, and the boundary integral vanishes because $\boldsymbol{\eta}|_{\partial\widetilde{\Omega}} = \mathbf{0}$.

Term 2. Define the edited-content alignment term as

$$\mathcal{J}_{\mathrm{e}}(\boldsymbol{z}+\epsilon\boldsymbol{\eta}) = \lambda_{\mathrm{e}}\int_{\widetilde{\Omega}} \widetilde{m}(\boldsymbol{p})\,\|(\boldsymbol{z}+\epsilon\boldsymbol{\eta}) - \widehat{\boldsymbol{x}}_1\|_2^2\,\mathrm{d}\boldsymbol{p}. \tag{20}$$

Its derivative at $\epsilon = 0$ is

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{J}_{\mathrm{e}}(\boldsymbol{z}+\epsilon\boldsymbol{\eta})\Big|_{\epsilon=0} = 2\lambda_{\mathrm{e}}\int_{\widetilde{\Omega}} \widetilde{m}(\boldsymbol{p})\,(\boldsymbol{z} - \widehat{\boldsymbol{x}}_1)^\top\,\boldsymbol{\eta}\,\mathrm{d}\boldsymbol{p}. \tag{21}$$

Term 3. Define the source-alignment term as

$$\mathcal{J}_{\mathrm{s}}(\boldsymbol{z}+\epsilon\boldsymbol{\eta}) = \lambda_{\mathrm{s}}\int_{\widetilde{\Omega}} (1-\widetilde{m}(\boldsymbol{p}))\,\|(\boldsymbol{z}+\epsilon\boldsymbol{\eta}) - \boldsymbol{x}_S\|_2^2\,\mathrm{d}\boldsymbol{p}. \tag{22}$$

Its derivative at $\epsilon = 0$ is

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{J}_{\mathrm{s}}(\boldsymbol{z}+\epsilon\boldsymbol{\eta})\Big|_{\epsilon=0} = 2\lambda_{\mathrm{s}}\int_{\widetilde{\Omega}} (1-\widetilde{m}(\boldsymbol{p}))\,(\boldsymbol{z} - \boldsymbol{x}_S)^\top\,\boldsymbol{\eta}\,\mathrm{d}\boldsymbol{p}. \tag{23}$$

Combining the three terms gives

$$0 = 2\int_{\widetilde{\Omega}} \Big[-(\Delta_{\boldsymbol{p}}\boldsymbol{z} - \Delta_{\boldsymbol{p}}\widehat{\boldsymbol{x}}_1) + \lambda_{\mathrm{e}}\widetilde{m}(\boldsymbol{p})\,(\boldsymbol{z} - \widehat{\boldsymbol{x}}_1) + \lambda_{\mathrm{s}}\,(1-\widetilde{m}(\boldsymbol{p}))\,(\boldsymbol{z} - \boldsymbol{x}_S)\Big]^\top \boldsymbol{\eta}\,\mathrm{d}\boldsymbol{p}. \tag{24}$$

Because this equality holds for every differentiable $\boldsymbol{\eta}$ satisfying the homogeneous boundary condition, the vector-valued integrand must vanish. Rearranging the resulting equation gives

$$\Big(-\Delta_{\boldsymbol{p}} + \lambda_{\mathrm{e}}\widetilde{m}(\boldsymbol{p}) + \lambda_{\mathrm{s}}(1-\widetilde{m}(\boldsymbol{p}))\Big)\boldsymbol{z}(\boldsymbol{p}) = \quad -\Delta_{\boldsymbol{p}}\widehat{\boldsymbol{x}}_1(\boldsymbol{p}) + \lambda_{\mathrm{e}}\widetilde{m}(\boldsymbol{p})\widehat{\boldsymbol{x}}_1(\boldsymbol{p}) + \lambda_{\mathrm{s}}(1-\widetilde{m}(\boldsymbol{p}))\boldsymbol{x}_S(\boldsymbol{p}), \tag{25}$$

where $\boldsymbol{z}|_{\partial\widetilde{\Omega}} = \boldsymbol{x}_S|_{\partial\widetilde{\Omega}}$.

A.2 Discretization

To solve Equation 25 in the discrete latent feature space, we discretize the spatial differential operators. Let $\boldsymbol{p} = (i, j)$ be a pixel in $\widetilde{\Omega}$, let $\mathrm{N}(\boldsymbol{p})$ denote its neighboring pixels, and let $\boldsymbol{z}_{\boldsymbol{p}} \in \mathbb{R}^C$ be the feature vector at $\boldsymbol{p}$. The discrete negative Laplacian $\boldsymbol{L}$ is

$$(\boldsymbol{L}\boldsymbol{z})_{\boldsymbol{p}} = \sum_{\boldsymbol{q}\in\mathrm{N}(\boldsymbol{p})} (\boldsymbol{z}_{\boldsymbol{p}} - \boldsymbol{z}_{\boldsymbol{q}}) = |\mathrm{N}(\boldsymbol{p})|\,\boldsymbol{z}_{\boldsymbol{p}} - \sum_{\boldsymbol{q}\in\mathrm{N}(\boldsymbol{p})} \boldsymbol{z}_{\boldsymbol{q}}. \tag{26}$$

The boundary condition $\boldsymbol{z}|_{\partial\widetilde{\Omega}} = \boldsymbol{x}_S|_{\partial\widetilde{\Omega}}$ remains in effect. For a neighbor $\boldsymbol{q} \in \mathrm{N}(\boldsymbol{p}) \cap \widetilde{\Omega}$, the value $\boldsymbol{z}_{\boldsymbol{q}}$ is unknown and must be solved. For a neighbor $\boldsymbol{q} \in \mathrm{N}(\boldsymbol{p})\backslash\widetilde{\Omega}$, the boundary condition provides the known value

$$\boldsymbol{z}_{\boldsymbol{q}} = \boldsymbol{x}_{S,\boldsymbol{q}}. \tag{27}$$

Therefore,

$$(\boldsymbol{L}\boldsymbol{z})_{\boldsymbol{p}} = |\mathrm{N}(\boldsymbol{p})|\, \boldsymbol{z}_{\boldsymbol{p}} - \sum_{\boldsymbol{q}\in\mathrm{N}(\boldsymbol{p})\cap\widetilde{\Omega}} \boldsymbol{z}_{\boldsymbol{q}} - \sum_{\boldsymbol{q}\in\mathrm{N}(\boldsymbol{p})\setminus\widetilde{\Omega}} \boldsymbol{x}_{S,\boldsymbol{q}}. \tag{28}$$

Using Equation 28, we discretize Equation 25. Let $\widetilde{m}_{\boldsymbol{p}}$ be the soft-mask value at $\boldsymbol{p}$ and define

$$J_{\boldsymbol{p}} = |\mathrm{N}(\boldsymbol{p})| + \lambda_{\mathrm{e}}\widetilde{m}_{\boldsymbol{p}} + \lambda_{\mathrm{s}}(1-\widetilde{m}_{\boldsymbol{p}}). \tag{29}$$

The discrete system is

$$\begin{aligned} J_{\boldsymbol{p}}\boldsymbol{z}_{\boldsymbol{p}} - \sum_{\boldsymbol{q}\in\mathrm{N}(\boldsymbol{p})\cap\widetilde{\Omega}} \boldsymbol{z}_{\boldsymbol{q}} = {} & \sum_{\boldsymbol{q}\in\mathrm{N}(\boldsymbol{p})\setminus\widetilde{\Omega}} \boldsymbol{x}_{S,\boldsymbol{q}} + \sum_{\boldsymbol{q}\in\mathrm{N}(\boldsymbol{p})} (\widehat{\boldsymbol{x}}_{1,\boldsymbol{p}} - \widehat{\boldsymbol{x}}_{1,\boldsymbol{q}}) \\ & + \lambda_{\mathrm{e}}\widetilde{m}_{\boldsymbol{p}}\widehat{\boldsymbol{x}}_{1,\boldsymbol{p}} + \lambda_{\mathrm{s}}(1-\widetilde{m}_{\boldsymbol{p}})\boldsymbol{x}_{S,\boldsymbol{p}}. \end{aligned} \tag{30}$$

In practice, we use the four-neighbor stencil

$$\mathrm{N}(\boldsymbol{p}) = \{(i+1,j),(i-1,j),(i,j+1),(i,j-1)\}, \qquad |\mathrm{N}(\boldsymbol{p})| = 4. \tag{31}$$

The neighborhood sum can be implemented efficiently using convolution.

A.3 JACOBI ITERATION

Soft-Poisson de-seaming obtains the corrected vector field $\boldsymbol{v}^{\star}_{\boldsymbol{\theta}}(\boldsymbol{x}(t),\sigma(t))$ from the refined target $\widehat{\boldsymbol{x}}^{\star}_1(t)$, which is approximated by solving Equation 30. We use Jacobi iteration because of its simplicity. The update is

$$\begin{aligned} \boldsymbol{z}_{\boldsymbol{p}}^{(k+1)} = \frac{1}{J_{\boldsymbol{p}}}\Bigg[& \sum_{\boldsymbol{q}\in\mathrm{N}(\boldsymbol{p})\cap\widetilde{\Omega}} \boldsymbol{z}_{\boldsymbol{q}}^{(k)} + \sum_{\boldsymbol{q}\in\mathrm{N}(\boldsymbol{p})\setminus\widetilde{\Omega}} \boldsymbol{x}_{S,\boldsymbol{q}} + \sum_{\boldsymbol{q}\in\mathrm{N}(\boldsymbol{p})} (\widehat{\boldsymbol{x}}_{1,\boldsymbol{p}} - \widehat{\boldsymbol{x}}_{1,\boldsymbol{q}}) \\ & + \lambda_{\mathrm{e}}\widetilde{m}_{\boldsymbol{p}}\widehat{\boldsymbol{x}}_{1,\boldsymbol{p}} + \lambda_{\mathrm{s}}(1-\widetilde{m}_{\boldsymbol{p}})\boldsymbol{x}_{S,\boldsymbol{p}}\Bigg]. \end{aligned} \tag{32}$$

After $K \in \mathbb{N}^{+}$ iterations, we use $\boldsymbol{z}_{\boldsymbol{p}}^{(K)}$ as the approximate solution $\boldsymbol{z}^{\star}_{\boldsymbol{p}}$ for every $\boldsymbol{p}\in\widetilde{\Omega}$.

A.4 VECTOR FIELD REFINEMENT WITH POISSON EDITING

Solving Equation 32 gives the local refined solution $\boldsymbol{z}^{\star}_{\boldsymbol{p}}(t) \approx \boldsymbol{z}_{\boldsymbol{p}}^{(K)}$ for $\boldsymbol{p}\in\widetilde{\Omega}$ at time t. Combining this solution with $\boldsymbol{x}_S(\boldsymbol{p})$ for $\boldsymbol{p}\notin\widetilde{\Omega}$ forms a complete refined feature map $\widehat{\mathbf{X}}^{\star}_1(t)\in\mathbb{R}^{C\times H\times W}$. We then obtain $\widehat{\boldsymbol{x}}^{\star}_1(t) = \mathrm{vec}(\widehat{\mathbf{X}}^{\star}_1(t)) \in \mathbb{R}^{CHW}$ and calculate the corrected vector field as

$$\boldsymbol{v}^{\star}_{\boldsymbol{\theta}}(\boldsymbol{x}(t),\sigma(t)) = \frac{\dot{\sigma}(t)}{\sigma(t)}\left(\boldsymbol{x}(t) - \widehat{\boldsymbol{x}}^{\star}_1(t)\right). \tag{33}$$

The next latent state is then obtained using

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \frac{\sigma_{i+1}-\sigma_i}{\dot{\sigma}_i}\boldsymbol{v}^{\star}_{\boldsymbol{\theta}}(\boldsymbol{x}_i,\sigma_i). \tag{34}$$

For $i = 0,\ldots,T-1$, we repeat the Poisson refinement before each sampling update until obtaining the final edited image $\boldsymbol{x}_T$. No refinement is required at the terminal state, where $\sigma_T = 0$.