Inline Critic Steers Image Editing

Weitai Kang[1] **Xiaohang Zhan**[2] **Yizhou Wang**[2] **Mang Tik Chiu**[2] **Jason Kuen**[2]

Kangning Liu[2] **Yan Yan**[1]

[1]University of Illinois Chicago [2]Adobe

Abstract

Instruction-based image editing exhibits heterogeneous difficulty not only across cases but also across regions of an image, motivating refinement approaches that allocate correction to where the model struggles. Existing refinement signals arrive late, after a fully generated image or a completed denoising step. We ask whether such a signal can act *within* an ongoing forward pass. To investigate this, we probe a frozen image-editing model and find that although generation capability emerges only in the last few layers, the error pattern is already set in early layers (rank correlation $\rho = 0.83$ with the final-layer error map). Based on this, we introduce **Inline Critic**, a learnable token that critiques a frozen model's predictions at its intermediate layers and steers its hidden states to refine generation during the forward pass. A three-stage recipe is proposed to stabilize the training from learning how to critique to steering generation. As a result, we achieve state of the art on GEdit-Bench (7.89), a $+9.4$ gain on RISEBench over the same backbone, and the strongest open-source result on KRIS-Bench (81.92, surpassing GPT-4o). We further provide analyses showing that the critic genuinely shapes the model's attention and prediction updates at subsequent layers.

1 Introduction

Instruction-based image editing aims to modify an input image according to a natural-language instruction. Despite recent progress, the task remains challenging because its difficulty is highly heterogeneous, both within a single image and across cases. Some demand major yet precise modification while others only need small changes or remain untouched [39, 47, 46, 52, 27]. An editor should therefore allocate different amounts of effort across regions and cases.

Recent work addresses this issue by introducing *refinement* signals that identify weak generations and allocate additional correction accordingly. One line of work computes this signal from a fully generated image: an external vision-language model or reward model critiques the output, which drives reranking, regeneration, or reinforcement learning [23, 29, 49, 25, 53, 24]. Another line obtains the signal earlier in the generation process, from each completed denoising step: confidence, uncertainty, or difficulty estimates then adjust the current step or steer subsequent iterations [39, 21, 9, 19, 1, 20, 12, 6, 22, 5]. Pushing this further, we are curious whether *the refinement signal can act even earlier, within the ongoing forward pass*, to adjust the prediction before a step completes.

To investigate this possibility, we analyze the intermediate hidden states of a strong pretrained image editor [45]. We freeze the model and train lightweight probe heads at multiple intermediate layers to assess their encoded generation ability. On 400 random samples drawn from different datasets, for each probe head's output we compute a prediction error map against the ground truth. We pool and rank-normalize the error maps, and compare them against the map of the final layer. As shown in Fig. 1, although intermediate-layer generation becomes competitive only near the last few layers, the

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

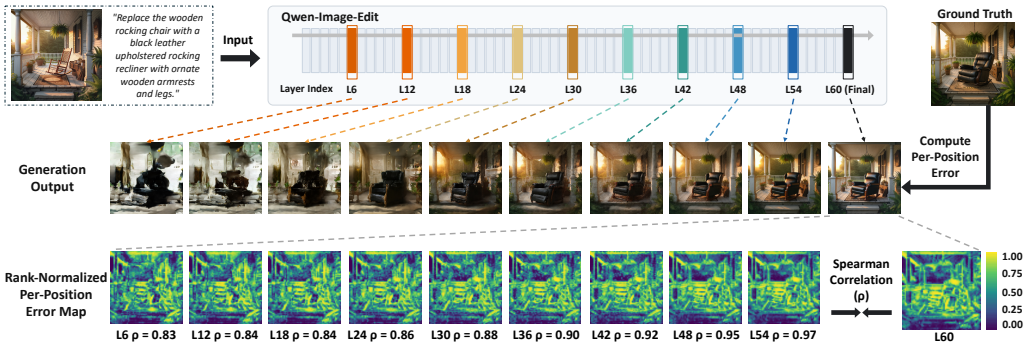

Figure 1: Motivation for inline critic. **Top:** On a frozen model, we add lightweight heads at intermediate blocks (L6–L54) to probe the generation outputs, with the original final output at L60. **Middle:** Each head runs its own multi-step denoising to generate an image. **Bottom:** Based on the final output, we compute a MSE map against the ground truth in each block. After 8x pooling and rank-normalization, we measure the Spearman correlation between each intermediate map and the final map. **Results:** Although generation capability only emerges in the last few blocks, the spatial error pattern is set far earlier. Even at L6, agreement with the final-layer map already reaches $\rho = 0.83$, and grows monotonically with depth. We therefore ask whether this prior can be exploited within the forward pass to let the model recognize and correct its weak regions in time.

relative error pattern is already stable within early layers, with a Spearman's $\rho = 0.83$ against the final layer's map averaged by steps and even higher agreement in later layers. In other words, *where the model performs well or poorly emerges long before the forward pass completes.*

Motivated by this, we aim to inline a critique into the forward pass, so that the model can learn to judge its emerged difficulty pattern. This critique is coupled with the generation objective, forcing it to be a useful refinement signal. Therefore, we introduce **Inline Critic**, a learnable token added to a frozen model that critiques its prediction at multiple intermediate layers and steers its hidden states to refine generation. We stabilize training through a three-stage recipe. We first probe the generation behavior of intermediate layers, the error of which the critic then learns to predict and later steers the model's prediction at each layer. At inference, the model does not need to compute the probe heads or critic predictions, with just an additional critic token in the input sequence.

Our contributions are as follows: (i) **Analysis**: by probing the intermediate layers, we find that the spatial pattern of *where* the model struggles is already stable in early layers, long before the forward pass completes; (ii) **Method**: motivated by this, we propose **Inline Critic**, a learnable token that critiques the model's prediction at intermediate layers and steers their hidden states to refine generation, together with a three-stage curriculum that first establishes the critique signal, then activates generation steering; (iii) **Results**: our method reaches state of the art on benchmarks (e.g. 7.89 on GEdit-Bench). We further provide analyses showing that the critic influence where our model re-allocates attention at the next layer and where it updates its prediction in subsequent layers.

2 Related Work

2.1 Refinement Mechanisms for Generation

We organize refinement methods by when the refinement signal becomes available. *Post-generation* methods derive feedback from completed images [23, 29, 49, 25, 53, 24], typically using an external vision-language model or reward model. *Post-step* methods derive feedback after a completed denoising forward pass and use it to adjust the current step or steer subsequent iterations [39, 21, 9, 19, 1, 20, 12, 6, 22, 5]. Our method instead produces a spatial critique within the ongoing forward pass and lets later layers consume it before the denoising step completes.

Post-Generation Refinement. Post-generation methods improve generation by evaluating completed generation outputs and using the verdict to drive reranking, regeneration, or reinforcement-learning fine-tuning. Reflect-DiT [24] feeds a vision-language model's textual critique back as

in-context guidance for another generation attempt. ReflectionFlow [53] scales inference-time optimization for text-to-image diffusion using VLM-derived reflection data. ReasonEdit [49] frames editing as a multimodal-LLM thinking–editing–reflection loop. ThinkRL-Edit [23], UniWorld-V2 [25], and EditScore [29] optimize image editors with reinforcement learning signals from explicit reward models or implicit MLLM feedback. These methods are powerful for semantic assessment, but their feedback is only available after the image has been generated.

Post-Step Refinement. Post-step methods use signals extracted after a denoising forward pass to adjust the current step or subsequent iterations. Diffusion Self-Guidance [12] extracts internal features and backpropagates a property-specific objective to steer sampling. Attend-and-Excite [6] similarly extracts attention activations and backpropagates an objective on them to update the latent. Perturbed-Attention Guidance [1] and Autoguidance [19] construct a degraded or weak prediction at each step and use the difference as guidance, while Feedback Guidance [21] formulates a state-dependent per-step feedback signal. Difficulty- and uncertainty-aware samplers provide finer spatial signals: Patch Forcing [39] estimates per-patch difficulty to allocate more denoising effort to hard regions, and BayesDiff [20] and De Vita and Belagiannis [9] estimate pixel-wise uncertainty for diffusion sampling. In discrete masked generation, Token-Critic [22] scores generated visual tokens for remasking, and MaskGIT [5] uses token confidence to schedule iterative decoding. These methods show that local difficulty and uncertainty are useful refinement signals, but the signal is produced only after the forward pass completes, rather than emerging within the ongoing forward pass.

2.2 Learnable Tokens for Generation

Learnable tokens provide a parameter-efficient way to adapt frozen generators. Textual Inversion [14] represents a new visual concept as a learned token embedding inserted into the prompt sequence. DreamArtist [11] learns positive and negative prompt embeddings, while ReVersion [16] learns tokens that encode relations between subjects. Several works make the learned representation layer- or time-dependent: NeTI [2] parameterizes the token as a function of layer index and denoising timestep, P+ [42] uses different embeddings across cross-attention layers, and ProSpect [50] assigns learnable tokens to different denoising stages. These tokens primarily act as conditioning codes that inject new concepts, relations, or attributes. Our critic token serves a different role: it is trained to critique the model by predicting spatial generation error from intermediate hidden states, and to expose this critique back to later layers to steer the generation behavior.

3 Method

We first provide a brief review of the image editing backbone (Sec. 3.1), then introduce our proposed Inline Critic (Sec. 3.2) and its three-stage training recipe (Sec. 3.2.1–3.2.3), illustrated in Fig. 2.

3.1 Preliminary: Qwen-Image-Edit

Our backbone is the open-source instruction-based image editor Qwen-Image-Edit [45]. A Qwen2.5-VL [3] encodes the editing instruction together with the source image into conditioning tokens $\mathbf{c}$, forming the *text stream*. The *image stream* consists of S *noise tokens* that the model denoises, concatenated with source image tokens encoded by a Wan2.1-VAE [41]. A double-stream diffusion transformer of $L = 60$ MM-DiT blocks [13] operates jointly on the two streams: at each block, the text-stream and image-stream tokens are concatenated along the sequence dimension and mixed by a joint self-attention, and modality-specific feed-forward and adaptive-LayerNorm layers update each stream. After block L, the noise-token outputs form the per-position velocity prediction.

Training follows the rectified-flow objective [28, 26]. Let $\mathbf{x}_0 \in \mathbb{R}^{S \times d_z}$ be the VAE-encoded target latent and $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ a noise sample. Given a time $t \in [0, 1]$, noise tokens are set to $\mathbf{x}_t = (1 - t)\,\mathbf{x}_0 + t\,\mathbf{x}_1$, and the network's per-position velocity prediction $f_\theta(\mathbf{x}_t, t, \mathbf{c})$ is regressed against the target velocity $\mathbf{v}^\star = \mathbf{x}_1 - \mathbf{x}_0$ with the mean-squared-error (MSE) loss

$$\mathcal{L}_{\text{gen}}(\theta) \;=\; \mathbb{E}_{t,\,\mathbf{x}_0,\,\mathbf{x}_1,\,\mathbf{c}}\Big[\big\| f_\theta(\mathbf{x}_t, t, \mathbf{c}) - \mathbf{v}^\star \big\|^2\Big]. \tag{1}$$

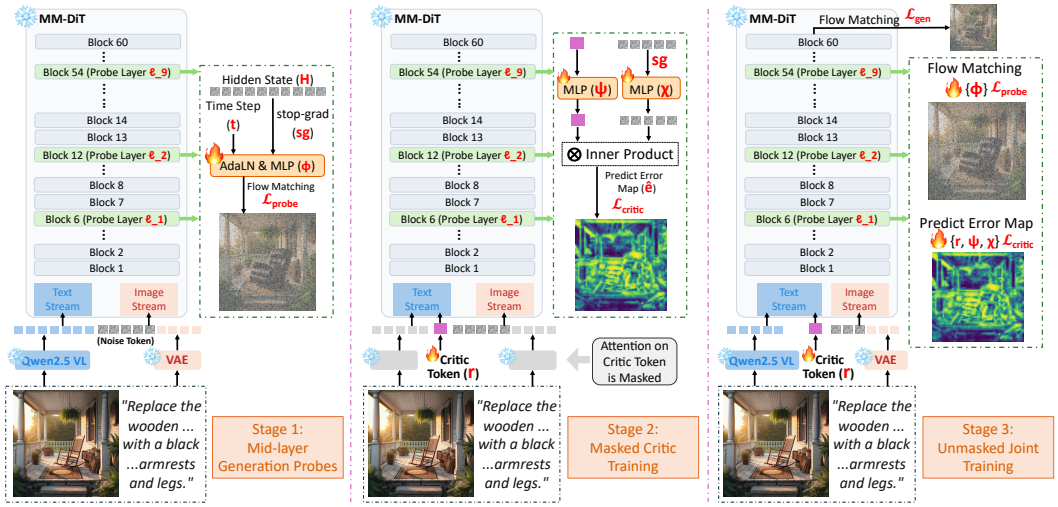

Figure 2: **Three-stage training of Inline Critic. Left**: Stage 1 trains probe heads at several intermediate layers to predict the generation target from frozen hidden states. **Middle**: Stage 2 adds a critic token that is trained, at every probed block, to predict the error of Stage-1's probes; an attention mask hides it from other tokens so the model and the probes are unaffected. **Right**: Stage 3 removes the mask and trains the critic jointly with the generation loss to steer generation.

3.2 Inline Critic

Inline Critic adds two lightweight modules to the frozen editor. First, we pick K intermediate blocks $\mathcal{L}^* = \{\ell_1, \ldots, \ell_K\} \subseteq \{1, \ldots, L\}$ uniformly across the network. At each $\ell \in \mathcal{L}^*$, a probe head $g_\phi^{(\ell)}$ takes the detached noise token hidden states $\mathbf{H}^{(\ell)} \in \mathbb{R}^{S \times d}$ as input and is supervised by the same rectified-flow target $\mathbf{v}^\star$ as the final layer; its per-position prediction error against $\mathbf{v}^\star$ forms the error map that the critic later learns to predict. Second, we append a learnable critic token $\mathbf{r} \in \mathbb{R}^d$ to the text stream, giving conditioning $[\mathbf{c}; \mathbf{r}]$. At every probed layer ℓ, small projection layers turn its hidden state $\mathbf{r}^{(\ell)}$ into a per-position estimate of this error map, and the critic token is later trained to influence the model's prediction through attention. The probes and projection layers participate only in training; at inference, they are dropped and only the critic token remains active in the forward pass.

Co-training all components from initialization would make the critic learn against a target that changes as the critic begins to influence the backbone, an instability analogous to the moving-target issue addressed by target networks in deep Q-learning [30]. We therefore stage the training so that the critic learns against a stable target before being released to influence the model. We detail each stage in the following subsections.

3.2.1 Stage 1: Mid-layer Generation Probes

As illustrated in Fig. 2 (left), we instantiate each probe $g_\phi^{(\ell)}$ as

$$g_\phi^{(\ell)}(\mathbf{H}^{(\ell)}, t) \;=\; \mathrm{MLP}_\phi^{(\ell)}\Big(\mathrm{AdaLN}_\phi^{(\ell)}\big(\mathrm{sg}(\mathbf{H}^{(\ell)}),\, \tau(t)\big)\Big), \tag{2}$$

where $\mathrm{sg}(\cdot)$ is the stop-gradient operator, $\tau(t)$ is the time embedding shared with the backbone, $\mathrm{AdaLN}_\phi^{(\ell)}$ is a continuous adaptive LayerNorm conditioned on $\tau(t)$ (the same module type used inside the MM-DiT blocks), and $\mathrm{MLP}_\phi^{(\ell)}$ is a 3-layer feed-forward head whose output matches the per-position velocity. Only ϕ is trainable in this stage. The probes are trained with the original rectified-flow loss of Eq. 1 applied at every probed layer:

$$\mathcal{L}_{\text{probe}}(\phi) \;=\; \frac{1}{K} \sum_{\ell \in \mathcal{L}^*} \mathbb{E}_{t,\, \mathbf{x}_0,\, \mathbf{x}_1,\, \mathbf{c}} \Big[\big\| g_\phi^{(\ell)}(\mathbf{H}^{(\ell)}, t) - \mathbf{v}^\star \big\|^2 \Big]. \tag{3}$$

3.2.2 Stage 2: Masked Critic Training

As illustrated in Fig. 2 (middle), we now introduce the critic token $\mathbf{r}$ together with K pairs of small projection heads $\{\psi^{(\ell)}, \chi^{(\ell)}\}_{\ell \in \mathcal{L}^*}$ (each a 3-layer MLP). At every probed layer ℓ and noise token position i, the critic prediction is the inner product

$$\hat{e}_i^{(\ell)} \;=\; \big\langle\, \psi^{(\ell)}\big(\mathbf{r}^{(\ell)}\big),\; \chi^{(\ell)}\big(\mathrm{sg}(\mathbf{H}_i^{(\ell)})\big)\,\big\rangle, \qquad i = 1, \ldots, S, \tag{4}$$

where $\mathbf{H}_i^{(\ell)} \in \mathbb{R}^d$ is the hidden state of the i-th noise token at block ℓ. The per-layer prediction error map is $\hat{\mathbf{e}}^{(\ell)} = (\hat{e}_1^{(\ell)}, \ldots, \hat{e}_S^{(\ell)})$.

To train the critic without changing the hidden states, we hide the critic token from the other tokens' attention at every block. Otherwise, the attention on the critic token would shift the noise tokens' hidden states and the probe error along with them, leaving the critic chasing a moving target. Letting r denote the critic's index in the joint sequence, we add a mask M to the pre-softmax scores with

$$M_{ij} \;=\; \begin{cases} -\infty & \text{if } i \neq r \text{ and } j = r, \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where i (query) and j (key) range over all tokens in the joint sequence. The joint self-attention output for every non-critic token is then identical to a run with no critic token, so the model and the per-position generation error stay fixed. The critic itself attends to the other tokens normally.

The critic target is the frozen Stage-1 probes' per-position generation error. Since this error spans several orders of magnitude across noise tokens and time steps, we rescale it with $\log(1 + \cdot)$ to keep the learning from being dominated by a few high-value positions:

$$e_i^{(\ell)} \;=\; \log\Big(1 + \big\| g_\phi^{(\ell)}(\mathbf{H}^{(\ell)}, t)_i - \mathbf{v}_i^\star \big\|^2\Big), \tag{6}$$

where $g_\phi^{(\ell)}(\mathbf{H}^{(\ell)}, t)_i$ and $\mathbf{v}_i^\star$ are the probe output and the target velocity, and the gradient is stopped on $e_i^{(\ell)}$. The critic is trained by MSE, averaged over the K probed layers and S noise-token positions:

$$\mathcal{L}_{\text{critic}}(\mathbf{r}, \psi, \chi) \;=\; \frac{1}{K\,S} \sum_{\ell \in \mathcal{L}^*} \sum_{i=1}^{S} \Big(\hat{e}_i^{(\ell)} \;-\; \mathrm{sg}\big(e_i^{(\ell)}\big)\Big)^2. \tag{7}$$

Only $\{\mathbf{r}, \psi^{(\ell)}, \chi^{(\ell)}\}$ are trainable, while θ and ϕ remain frozen. In Stage 2, the critic predicts the per-position generation error at every probed layer without affecting the model or its own target.

3.2.3 Stage 3: Unmasked Joint Training

As illustrated in Fig. 2 (right), in Stage 3 we unmask the critic and jointly optimize the generation loss with the critic loss. This unmasking relaxes the strict target stability of Stage 2 in exchange for joint adaptation, which is acceptable since the critic has been pre-calibrated in Stage 2. The joint loss is

$$\mathcal{L}_{\text{stage 3}} \;=\; \mathcal{L}_{\text{gen}} \;+\; \lambda_c\, \mathcal{L}_{\text{critic}} \;+\; \lambda_p\, \mathcal{L}_{\text{probe}}, \tag{8}$$

where $\mathcal{L}_{\text{gen}}, \mathcal{L}_{\text{critic}}, \mathcal{L}_{\text{probe}}$ are Eqs. 1, 7, 3, and $\lambda_c, \lambda_p > 0$ are loss weights. The trainable parameters are $\{\phi, \mathbf{r}, \psi^{(\ell)}, \chi^{(\ell)}\}$, while θ remains frozen. Because the unmasked attention and $\mathcal{L}_{\text{gen}}$ now shift the noise token hidden states at intermediate blocks and thereby change the encoded generation behavior that the probes were trained against, we unfreeze ϕ and keep $\mathcal{L}_{\text{probe}}$ in the loss so that the probes can track this change. With the backbone frozen, unmasking enables the critic token $\mathbf{r}$ to steer intermediate hidden states through attention. Gradients from $\mathcal{L}_{\text{gen}}$ then flow into $\mathbf{r}$ via these attention connections, driving it to refine the model's generation behavior, while $\mathcal{L}_{\text{critic}}$ keeps the critic anchored to the updated probed generation error. In effect, this turns the critic token into an inline verifier that self-evaluates the intermediate generation at every probed layer and emits corrective feedback through attention, without any additional forward pass or external reward model. At inference, the critic token stays unmasked in the forward pass, and we no longer compute the probe outputs or the critic prediction. The per-step cost is therefore essentially identical to the original editor backbone.

Table 1: **Main results on GEdit-Bench [27] (EN split).** The best Overall score is in bold.

Model	Semantic Consistency	Perceptual Quality	Overall
BAGEL [10]	7.36	6.83	6.52
FLUX.1-Kontext [Pro] [4]	7.02	7.60	6.56
Step1X-Edit [27]	7.66	7.35	6.97
Qwen-Image-Edit-2509 [45]	8.15	7.86	7.54
UniWorld-Qwen-Image-Edit-2509 [25]	8.36	7.87	7.76
ReasonEdit-Q [49]	8.34	7.97	7.77
UniWorld-V2 [25]	8.39	8.02	7.83
Critic-v1	8.33	7.98	7.80
Critic-v2	8.48	8.02	**7.89**

Table 2: **Main results on KRIS-Bench [46].** Closed-source proprietary models are marked with †. The best Overall score is in bold.

Model	Attribute Perception	Spatial Perception	Temporal Prediction	Factual	Social Science	Natural Science	Conceptual	Logical Reasoning	Instruction Decomposition	Procedural	Overall
BAGEL-Think [10]	67.42	68.33	58.67	66.18	63.55	61.40	61.92	48.12	50.22	49.02	60.18
Doubao† [43]	70.92	59.17	40.58	63.30	65.50	61.19	62.23	47.75	60.58	54.17	60.70
Gemini 2.0† [18]	66.33	63.33	63.92	65.26	68.19	56.94	59.65	54.13	71.67	62.90	62.41
Uni-CoT [36]	72.76	72.87	67.10	71.85	70.81	66.00	67.16	53.43	73.93	63.68	68.00
Critic-v1	84.06	86.42	55.97	79.96	83.35	82.57	82.76	67.91	74.39	71.15	79.02
GPT-4o† [31]	83.17	79.08	68.25	79.80	85.50	80.06	81.37	71.56	85.08	78.32	80.09
Critic-v2	85.76	89.50	58.44	82.09	84.40	84.91	84.79	77.46	75.94	76.70	**81.92**

4 Experiments

4.1 Implementation

We train Inline Critic on two Qwen-Image-Edit [45] backbones, Qwen-Image-Edit-2509 and Qwen-Image-Edit-2511, resulting in Critic-v1 and Critic-v2, respectively. The model uses $K = 9$ probe layers. Training is conducted on a 2.2M-pair mixture of public image-editing datasets [44, 51, 17, 7, 8, 15, 48, 35]. The instruction is replaced by an Qwen3.5-27B [37] rewrite using the ThinkRL-Edit [23] prompt with probability 0.5. For the main Critic-v1 and Critic-v2 models, each stage is trained for 4 epochs with AdamW on 32 A100-80G GPUs. We evaluate on RISEBench [52], KRIS-Bench [46], GEdit-Bench [27], and ImgEdit [47], with the same test-time rewrite applied to Critic-v1 and Critic-v2 on RISEBench and KRIS-Bench. GEdit-Bench and ImgEdit use raw instructions. We use 40 denoising steps at guidance scale 4.0. Details of dataset, augmentation, optimization, and training hyperparameters are provided in Appendix A. Qualitative results and limitations are put in Appendix B and C.

4.2 Main Results

GEdit-Bench. We start with general editing instructions on GEdit-Bench, scored by GPT-4.1. As shown in Tab. 1, our Critic-v2 achieves state-of-the-art performance with an $\mathbf{7.89}$ Overall score, also obtaining the highest Semantic Consistency (8.48) and tying for the best Perceptual Quality (8.02). Our Critic-v1 reaches 7.80, outperforming UniWorld-Qwen-Image-Edit-2509 [25] (7.76) under the same backbone. Both Semantic Consistency and Perceptual Quality improve over the backbone.

KRIS-Bench. We next evaluate on reasoning-heavy benchmarks, where the difficulty arises primarily from language reasoning and world knowledge. This shifts the bottleneck onto the understanding encoder (Qwen2.5-VL [3]) of our backbone, so we first rewrite the original instruction with the stronger Qwen3.5-27B to provide a clearer, more direct prompt. We provide ablation on this rewrite in Sec. 4.3. Table 2 reports per-sub-axis scores on KRIS-Bench, scored by GPT-4o with the official prompts. Our Critic-v2 ranks first overall at $\mathbf{81.92}$, surpassing GPT-4o [31] (80.09) by $+1.83$ and the strongest open-source baseline Uni-CoT [36] (68.00) by $+13.92$.

RISEBench. RISEBench (Tab. 3, scored by GPT-4.1 with the official prompts) targets the same reasoning-heavy regime, with categories that probe temporal, causal, spatial, and logical reasoning. Our Critic-v2 attains an Overall of $\mathbf{37.8}$ with rewritten instructions, compared with 19.4 for its

Table 3: **Main results on RISEBench [52] (%).** Closed-source proprietary models are marked with †. The best Overall score among open-source editors is in bold.

Method	Instruction Reasoning	Appearance Consistency	Visual Plausibility	Temporal Reasoning	Causal Reasoning	Spatial Reasoning	Logical Reasoning	Overall
Closed-source proprietary models								
GPT-Image-1† [32]	62.8	80.2	94.9	36.5	34.4	37.0	10.6	30.0
Gemini-2.5-Flash-Image† [40]	61.2	86.0	91.3	29.4	48.9	37.0	18.8	33.9
Gemini-3-pro-image† [38]	77.0	85.5	94.4	43.5	63.3	48.0	37.6	48.3
GPT-Image-2† [34]	73.8	89.3	94.9	45.9	66.7	50.0	34.1	49.4
GPT-Image-1.5† [33]	69.7	92.5	94.9	57.6	62.2	62.0	21.2	51.4
Open-source editors								
FLUX.1-Kontext-Dev [4]	26.0	71.6	85.2	2.3	5.5	13.0	1.2	5.8
Qwen-Image-Edit-2509 [45]	37.2	66.4	86.9	4.7	11.1	17.0	2.4	9.2
BAGEL (w/ CoT) [10]	45.9	73.8	80.1	5.9	17.8	21.0	1.2	11.9
Qwen-Image-Edit-2511 [45]	49.9	71.0	91.5	21.2	18.9	31.0	4.7	19.4
Critic-v1	59.1	85.4	92.2	37.6	46.7	42.0	8.2	34.2
Critic-v2	63.7	88.5	90.7	35.3	47.8	50.0	15.3	**37.8**

Table 4: **Main results on ImgEdit [47].** Recent strong baselines are re-evaluated under a consistent protocol. The best Overall score is in bold.

Method	action	add	adjust	bkgd	compose	extract	remove	replace	style	Overall
Qwen-Image-Edit-2509 [45]	4.769	4.407	4.323	4.255	3.652	3.427	4.512	4.605	4.850	4.311
UniWorld-Qwen-Image-Edit-2509 [25]	4.806	4.407	4.330	4.255	3.449	3.766	4.535	4.609	4.883	4.338
Qwen-Image-Edit-2511 [45]	4.639	4.425	4.287	4.191	3.855	3.877	4.519	4.587	4.860	4.360
Critic-v1	4.824	4.533	4.351	4.316	3.739	3.578	4.469	4.609	4.837	4.362
Critic-v2	4.806	4.481	4.394	4.248	3.899	3.889	4.539	4.558	4.863	**4.408**

backbone with raw instructions. The $+18.4$ gap therefore compares different prompt conditions. Table 6 provides comparisons under matched prompt conditions on the 2509 backbone. Critic-v2 establishes a new state of the art among open-source editors and also outperforms several proprietary models, including Gemini-2.5-Flash-Image† (33.9) and GPT-Image-1† (30.0).

ImgEdit. We score ImgEdit by GPT-4.1 with official prompts. We find that run-to-run variance of this benchmark can exceed typical method-to-method gaps. Compounding this issue, most prior works do not release their exact scoring configurations (e.g., GPT temperature). We therefore re-evaluate recent strong baselines under the consistent protocol in Tab. 4. Our Critic-v2 obtains the best Overall score at $\mathbf{4.408}$, and Inline Critic consistently improves over the corresponding backbone on most categories (7 of 9 for Critic-v1, 8 of 9 for Critic-v2). On the same 2509 backbone, our Critic-v1 (4.362) further outperforms the UniWorld-Qwen-Image-Edit-2509 RL-finetune [25] (4.338).

Heterogeneity results. Fig. 3 reports the performance in cases with different difficulty on ImgEdit with Qwen-Image-Edit-2509 as the baseline. The left panel groups samples into bins according to the baseline's per-sample scores, while the right panel groups samples by edit type. Both panels show that the gain of Critic-v1 concentrates on the hard cases, with a bit of variance on the easy cases. Specifically, we bring $+\mathbf{0.897}$ on the $[1, 2)$ score bin and $+\mathbf{0.151}$ on the hardest edit type `extract`.

4.3 Ablation Studies

For the ImgEdit ablations in Tab. 5, we use 288K samples from the training set on Qwen-Image-Edit-2509. Each stage of the three-stage curriculum runs for one epoch, with the generation loss used only in Stage 3. LoRA, the learnable token, and the single-stage critic each receive one epoch of generation-loss training. For reproducibility, we choose Qwen3.5-27B as the judge at temperature 0.

As shown in Tab. 5, simply training with additional data by LoRA brings only a very limited improvement. For Inline Critic, directly using only the stage 3 of Sec. 3.2 is suboptimal. In that case, the critic target keeps moving before the critique training stabilizes, and the generation loss might pull the critic token toward acting as a generic learnable conditioning token. Consequently, single-stage Inline Critic reaches 4.213, essentially matching the learnable token baseline at 4.214. Instead, introducing our three-stage training recipe can improve Overall to 4.252. We also find that the choice of K is not sensitive: increasing from $K{=}4$ to $K{=}9$ gives the same Overall score (both at 4.252). We adopt $K{=}9$ in the final model mainly to add a small amount of trainable capacity and to better expose the behavior of different model layers. Finally, rewrite augmentation provides a further improvement, raising Overall to 4.270. On the reasoning-heavy RISEBench, Tab. 6 shows that Inline

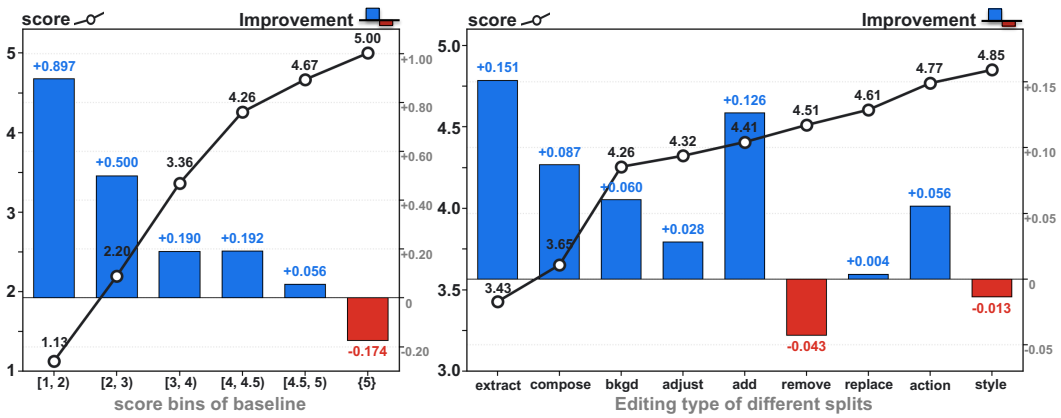

Figure 3: **Heterogeneity results on ImgEdit.** Bins are formed by baseline performance (left) and edit type (right). Our method's improvement concentrates on the hard cases.

Table 5: **Ablation of design choices on ImgEdit.** "Learnable token" adds a single learnable conditioning token trained with only the generation loss. K denotes the number of probed layers. "3-stage" follows the training curriculum of Sec. 3.2; "single-stage" uses only stage 3 of Sec. 3.2. "rewrite aug." replaces the training instruction with a Qwen3.5-27B rewrite [23] with probability 0.5.

Method	action	add	adjust	bkgd	compose	extract	remove	replace	style	**Overall**
Qwen-Image-Edit-2509 (backbone)	4.750	4.179	4.174	3.961	3.725	3.365	4.318	4.268	4.797	**4.171**
+ LoRA	4.815	4.088	3.940	4.060	3.812	3.444	4.341	4.304	4.850	**4.184**
+ Learnable token	4.843	4.214	4.181	4.184	3.696	3.439	4.302	4.362	4.707	**4.214**
+ Inline Critic, single-stage (K=4)	4.796	4.109	4.145	4.028	4.000	3.470	4.380	4.199	4.790	**4.213**
+ Inline Critic, 3-stage (K=4)	4.861	4.168	4.241	4.085	3.884	3.544	4.399	4.294	4.793	**4.252**
+ Inline Critic, 3-stage (K=9)	4.852	4.249	4.153	4.234	3.725	3.521	4.450	4.286	4.797	**4.252**
+ Inline Critic, 3-stage (K=4) + rewrite aug.	4.935	4.197	4.145	4.128	3.826	3.715	4.395	4.297	4.790	**4.270**

Critic consistently improves the backbone under both instruction settings: from 9.2 to 18.6 (+9.4) with raw instructions, and from 28.9 to 34.2 (+5.3) with rewritten instructions.

4.4 Analysis

We analyze whether the critic token truly steers spatial changes inside the backbone. Our experiments show that three behaviors align with the critic prediction: A) the spatial redistribution of the model's attention, B) the spatial update of the model's generation behavior, and C) the difference in generation shift between runs with and without the critic token.

Setup. All analyses use $n = 600$ held-out validation samples. We empirically inspect layer $L36$ to $L42$ at denoising step $t = 38$, where the critic is better calibrated, thereby reducing the confound that the critic might be under-trained at the analyzed point. All Spearman correlations are computed in the same way as in Fig. 1. Let $E = \hat{e}^{(36)}$ be the critic's predicted error map. For each experiment, we define a spatial-change map Y on the noise tokens and measure its agreement with E by

$$\rho = \mathrm{Spearman}(Y, E). \tag{9}$$

Experiment A: attention redistribution. We first measure where the model reallocates attention between $L36$ and $L37$, and compare this spatial pattern against the critic's prediction E. For noise tokens acting as the key, $\alpha^{(\ell)}$ is a map about the sum of pre-softmax attention scores each token receives from all queries. We then use

$$Y_A = |\alpha^{(37)} - \alpha^{(36)}|. \tag{10}$$

As shown in Fig. 4 (left), the median ρ is +0.542, with $P(\rho > 0) = 0.90$ and $P(\rho > 0.2) = 0.80$. The two maps are positively aligned: regions flagged as harder (i.e., with higher error prediction) by the critic also exhibit the higher attention changes, indicating that the critic redirects attention.

Table 6: **Ablation of test-time instruction rewrite on RISEBench [52] (%).** We compare Qwen-Image-Edit-2509 and Critic-v1 under raw and rewritten test instructions. The Qwen3.5-27B rewriter follows the ThinkRL-Edit [23] template. The best Overall score is in bold.

Method	Instruction Reasoning	Appearance Consistency	Visual Plausibility	Temporal Reasoning	Causal Reasoning	Spatial Reasoning	Logical Reasoning	Overall
Raw instruction								
Qwen-Image-Edit-2509 [45]	37.2	66.4	86.9	4.7	11.1	17.0	2.4	9.2
Critic-v1	44.9	73.2	89.7	7.1	24.4	32.0	8.2	18.6
Rewritten instruction								
Qwen-Image-Edit-2509 [45]	57.1	85.5	91.5	36.5	32.2	41.0	3.5	28.9
Critic-v1	59.1	85.4	92.2	37.6	46.7	42.0	8.2	**34.2**

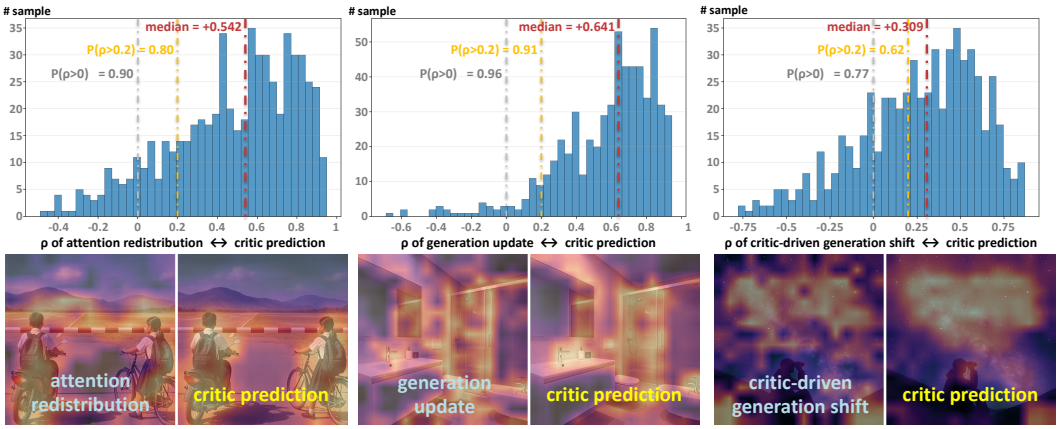

Figure 4: **Top**: per-sample histogram of ρ between the different model-side spatial map and the critic's prediction. **Bottom**: one sample showing the spatial map (left) and critic's prediction (right).

Experiment B: generation behavior update. We next measure where the model's encoded generation behavior changes between $L36$ and $L42$, and compare this spatial pattern against the critic's prediction E. For each probe layer, $m^{(\ell)}$ is a map about the error between the probe layer ℓ and the ground-truth. We then use

$$Y_B = |m^{(42)} - m^{(36)}|. \tag{11}$$

Fig. 4 (center) shows a strong agreement, with median $\rho = +0.641$, $P(\rho > 0) = 0.96$, and $P(\rho > 0.2) = 0.91$. The regions where the critic predicts higher error are essentially the regions where the flow matching representation moves the most between two probe layers, indicating that the critic's prediction shapes the actual generation update.

Experiment C: generation shift by critic's causal intervention. We finally test whether the alignment observed in Experiment B is genuinely causal. To this end, we run inference twice on the same input: once without masking the critic token (un), and once with the critic token masked out (ma). Let $m^{(\ell)}_{\text{un}}$ and $m^{(\ell)}_{\text{ma}}$ denote the generation error $m^{(\ell)}$ (as defined in Experiment B) under the unmasked and masked settings, respectively. We then define

$$Y_C = \left|(m^{(42)}_{\text{un}} - m^{(36)}_{\text{un}}) - (m^{(42)}_{\text{ma}} - m^{(36)}_{\text{ma}})\right|, \tag{12}$$

which is used to measure, between the two layers, how much of the $L36 \to L42$ shift in generation error can be attributed to access to the critic. As shown in Fig. 4 (right), we observe a clear positive correlation, with median $\rho = +0.309$, $P(\rho > 0) = 0.77$, and $P(\rho > 0.2) = 0.62$. This suggests that the critic plays a causal role in steering the generation process.

5 Conclusion

We introduced **Inline Critic**, a learnable token that critiques a frozen editor at intermediate layers and steers its hidden states to refine generation within the forward pass, trained by a three-stage

curriculum. Empirically, Critic-v2 achieves 7.89 Overall on GEdit-Bench, 81.92 on KRIS-Bench, 37.8 on RISEBench with rewritten instructions ($+18.4$ relative to the backbone with raw instructions), showing consistent improvements across general editing and reasoning-heavy settings. Our analysis further shows that the attention and generation updates align with the critic's prediction, supporting Inline Critic as an effective refinement mechanism.

Acknowledgements

This research was partially supported by NSF grants IIS-2525840, CNS-2432534, and ECCS-2514574, as well as an unrestricted gift from Adobe Research. The views and conclusions expressed in this article are solely those of the authors and do not necessarily reflect those of the funding agencies or supporting organizations.

References

[1] Donghoon Ahn, Hyoungwon Cho, Jaewon Min, Wooseok Jang, Jungwoo Kim, SeonHwa Kim, Hyun Hee Park, Kyong Hwan Jin, and Seungryong Kim. Self-rectifying diffusion sampling with perturbed-attention guidance. In *Computer Vision – ECCV 2024*, pages 1–17, 2024. doi: 10.1007/978-3-031-73464-9_1. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/09184.pdf.

[2] Yuval Alaluf, Elad Richardson, Gal Metzer, and Daniel Cohen-Or. A neural space-time representation for text-to-image personalization. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 42(6):243:1–243:10, 2023. doi: 10.1145/3618322. URL https://doi.org/10.1145/3618322.

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. URL https://arxiv.org/abs/2502.13923.

[4] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space. Black Forest Labs technical report (arXiv:2506.15742), 2025. URL https://arxiv.org/abs/2506.15742.

[5] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11315–11325, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Chang_MaskGIT_Masked_Generative_Image_Transformer_CVPR_2022_paper.html.

[6] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (SIGGRAPH)*, 42(4):1–10, 2023. doi: 10.1145/3592116. URL https://doi.org/10.1145/3592116.

[7] Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. ShareGPT-4o-Image: Aligning multimodal models with GPT-4o-level image generation. *arXiv preprint arXiv:2506.18095*, 2025. URL https://arxiv.org/abs/2506.18095.

[8] Zhihong Chen, Xuehai Bai, Yang Shi, Chaoyou Fu, Huanyu Zhang, Haotian Wang, Xiaoyan Sun, Zhang Zhang, Liang Wang, Yuanxing Zhang, Pengfei Wan, and Yi-Fan Zhang. OpenGPT-4o-Image: A comprehensive dataset for advanced image generation and editing. *arXiv preprint arXiv:2509.24900*, 2025. URL https://arxiv.org/abs/2509.24900.

[9] Michele De Vita and Vasileios Belagiannis. Diffusion model guided sampling with pixel-wise aleatoric uncertainty estimation. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 3844–3854, February 2025. URL https://openaccess.thecvf.com/content/WACV2025/html/De_Vita_Diffusion_Model_Guided_Sampling_with_Pixel-Wise_Aleatoric_Uncertainty_Estimation_WACV_2025_paper.html.

[10] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. URL https://arxiv.org/abs/2505.14683.

[11] Ziyi Dong, Pengxu Wei, and Liang Lin. DreamArtist++: Controllable one-shot text-to-image generation via positive-negative adapter. *arXiv preprint arXiv:2211.11337*, 2022. URL https://arxiv.org/abs/2211.11337.

[12] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 16222–16239. Curran Associates, Inc., 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/3469b211b829b39d2b0cfd3b880a869c-Abstract-Conference.html.

[13] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 12606–12633. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

[14] 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 *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=NAQvF08TcyG.

[15] Feng Han, Yibin Wang, Chenglin Li, Zheming Liang, Dianyi Wang, Yang Jiao, Zhipeng Wei, Chao Gong, Cheng Jin, Jingjing Chen, and Jiaqi Wang. UniREditBench: A unified reasoning-based image editing benchmark. *arXiv preprint arXiv:2511.01295*, 2025. URL https://arxiv.org/abs/2511.01295.

[16] Ziqi Huang, Tianxing Wu, Yuming Jiang, Kelvin C. K. Chan, and Ziwei Liu. ReVersion: Diffusion-based relation inversion from images. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. doi: 10.1145/3680528.3687658. URL https://ziqihuangg.github.io/projects/reversion.

[17] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Cihang Xie, and Yuyin Zhou. HQ-Edit: A high-quality dataset for instruction-based image editing. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=mZptYYttFj.

[18] Kat Kampf and Nicole Brichtova. Experiment with Gemini 2.0 Flash native image generation. Google Developers Blog, 2025. URL https://developers.googleblog.com/en/experiment-with-gemini-20-flash-native-image-generation.

[19] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 52996–53021. Curran Associates, Inc., 2024. doi: 10.52202/079017-1679. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5ee7ed60a7e8169012224dec5fe0d27f-Abstract-Conference.html.

[20] Siqi Kou, Lei Gan, Dequan Wang, Chongxuan Li, and Zhijie Deng. BayesDiff: Estimating pixel-wise uncertainty in diffusion via bayesian inference. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=YcM6ofShwY.

[21] Felix Koulischer, Florian Handke, Johannes Deleu, Thomas Demeester, and Luca Ambrogioni. Feedback guidance of diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=8ySOcf7UpM.

[22] José Lezama, Huiwen Chang, Lu Jiang, and Irfan Essa. Improved masked image generation with token-critic. In *Computer Vision – ECCV 2022*, pages 70–86, 2022. doi: 10.1007/978-3-031-20050-2_5. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/2901_ECCV_2022_paper.php.

[23] Hengjia Li, Liming Jiang, Qing Yan, Yizhi Song, Hao Kang, Zichuan Liu, Xin Lu, Boxi Wu, and Deng Cai. ThinkRL-Edit: Thinking in reinforcement learning for reasoning-centric image editing. *arXiv preprint arXiv:2601.03467*, 2026. URL https://arxiv.org/abs/2601.03467.

[24] Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Arsh Koneru, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Reflect-DiT: Inference-time scaling for text-to-image diffusion transformers via in-context reflection. *arXiv preprint arXiv:2503.12271*, 2025. URL https://arxiv.org/abs/2503.12271.

[25] Zongjian Li, Zheyuan Liu, Qihui Zhang, Bin Lin, Feize Wu, Shenghai Yuan, Zhiyuan Yan, Yang Ye, Wangbo Yu, Yuwei Niu, Shaodong Wang, Xinhua Cheng, and Li Yuan. Uniworld-V2: Reinforce image editing with diffusion negative-aware finetuning and MLLM implicit feedback. *arXiv preprint arXiv:2510.16888*, 2025. URL https://arxiv.org/abs/2510.16888.

[26] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

[27] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai, Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, Yibo Zhu, Binxing Jiao, Xiangyu Zhang, Gang Yu, and Daxin Jiang. Step1X-Edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025. URL https://arxiv.org/abs/2504.17761.

[28] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

[29] Xin Luo, Jiahao Wang, Chenyuan Wu, Shitao Xiao, Xiyan Jiang, Defu Lian, Jiajun Zhang, Dong Liu, and Zheng Liu. EditScore: Unlocking online RL for image editing via high-fidelity reward modeling. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=E7YpL4L4Xh.

[30] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. doi: 10.1038/nature14236. URL https://doi.org/10.1038/nature14236.

[31] OpenAI. Introducing 4o image generation. OpenAI product release, 2025. URL https://openai.com/index/introducing-4o-image-generation/.

[32] OpenAI. GPT Image 1. OpenAI API model card, 2025. URL https://platform.openai.com/docs/models/gpt-image-1.

[33] OpenAI. GPT Image 1.5. OpenAI API model card, 2025. URL https://platform.openai.com/docs/models/gpt-image-1.5.

[34] OpenAI. GPT Image 2. OpenAI API model card, 2026. URL https://platform.openai.com/docs/models/gpt-image-2.

[35] Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. Pico-Banana-400K: A large-scale dataset for text-guided image editing. *arXiv preprint arXiv:2510.19808*, 2025. URL https://arxiv.org/abs/2510.19808.

[36] Luozheng Qin, Jia Gong, Yuqing Sun, Tianjiao Li, Haoyu Pan, Mengping Yang, Xiaomeng Yang, Chao Qu, Zhiyu Tan, and Hao Li. Uni-CoT: Towards unified chain-of-thought reasoning across text and vision. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=5nevWRoNjn.

[37] Qwen Team. Qwen3.5-27B. Hugging Face model card, 2026. URL https://huggingface.co/Qwen/Qwen3.5-27B.

[38] Naina Raisinghani. Introducing nano banana pro. Google Blog; introduces Gemini 3 Pro Image, 2025. URL https://blog.google/innovation-and-ai/products/nano-banana-pro.

[39] Johannes Schusterbauer, Ming Gui, Yusong Li, Pingchuan Ma, Felix Krause, and Björn Ommer. Denoising, fast and slow: Difficulty-aware adaptive sampling for image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2604.19141.

[40] David Sharon and Nicole Brichtova. Image editing in Gemini just got a major upgrade. Google Blog; introduces Gemini 2.5 Flash Image / Nano Banana, 2025. URL https://blog.google/products/gemini/updated-image-editing-model.

[41] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

[42] Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. P+: Extended textual conditioning in text-to-image generation. *arXiv preprint arXiv:2303.09522*, 2023. URL https://arxiv.org/abs/2303.09522.

[43] Peng Wang, Yichun Shi, Xiaochen Lian, Zhonghua Zhai, Xin Xia, Xuefeng Xiao, Weilin Huang, and Jianchao Yang. SeedEdit 3.0: Fast and high-quality generative image editing. *arXiv preprint arXiv:2506.05083*, 2025. URL https://arxiv.org/abs/2506.05083.

[44] Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhu Chen. OmniEdit: Building image editing generalist models through specialist supervision. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=Hlm0cga0sv.

[45] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025. URL https://arxiv.org/abs/2508.02324.

[46] Yongliang Wu, Zonghui Li, Xinting Hu, Xinyu Ye, Xianfang Zeng, Gang Yu, Wenbo Zhu, Bernt Schiele, Ming-Hsuan Yang, and Xu Yang. KRIS-Bench: Benchmarking next-level intelligent image editing models. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=aWSh1Ec64T.

[47] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. ImgEdit: A unified image editing dataset and benchmark. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=uUCSrMlfD3.

[48] Yejy53. Nano-consistent-150k. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/Yejy53/Nano-consistent-150k.

[49] Fukun Yin, Shiyu Liu, Yucheng Han, Zhibo Wang, Peng Xing, Rui Wang, Wei Cheng, Yingming Wang, Aojie Li, Zixin Yin, Pengtao Chen, Xiangyu Zhang, Daxin Jiang, Xianfang Zeng, and Gang Yu. ReasonEdit: Towards reasoning-enhanced image editing models. *arXiv preprint arXiv:2511.22625*, 2025. URL https://arxiv.org/abs/2511.22625.

[50] Yuxin Zhang, Weiming Dong, Fan Tang, Nisha Huang, Haibin Huang, Chongyang Ma, Tong-Yee Lee, Oliver Deussen, and Changsheng Xu. ProSpect: Prompt spectrum for attribute-aware personalization of diffusion models. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 42(6):244:1–244:14, 2023. doi: 10.1145/3618342. URL https://doi.org/10.1145/3618342.

[51] Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. UltraEdit: Instruction-based fine-grained image editing at scale. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=9ZDdlgH6O8.

[52] Xiangyu Zhao, Peiyuan Zhang, Kexian Tang, Xiaorong Zhu, Hao Li, Wenhao Chai, Zicheng Zhang, Renqiu Xia, Guangtao Zhai, Junchi Yan, Hua Yang, Xue Yang, and Haodong Duan. Envisioning beyond the pixels: Benchmarking reasoning-informed visual editing. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=T3UksaPK64.

[53] Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15329–15339, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhuo_From_Reflection_to_Perfection_Scaling_Inference-Time_Optimization_for_Text-to-Image_Diffusion_ICCV_2025_paper.html.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Sec. 1 state our three contributions (the probing analysis, the Inline Critic with three-stage training, and the empirical results), each of which is supported in Sec. 3 and Sec. 4. Quantitative claims (e.g., GEdit-Bench 7.89, KRIS-Bench 81.92, +9.4 on RISEBench) match the corresponding tables in Sec. 4.2.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix C contains a Limitations section: the critic's training target is per-position MSE of the probe head, the training data is noisy, and MSE itself is not always a faithful measure of generation quality, which leaves the critic's learning signal noisy. We point to a feature-space critic target as a natural next step.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not contain theorems or formal theoretical results; the equations in Sec. 3 are definitions of the model architecture and training objective, not formal theoretical claims requiring proof.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sec. 3 fully specifies the architecture, the three-stage training procedure, all loss functions, the attention-mask formulation, and the training/inference behavior. Sec. 4.1 lists the backbones, the probed layers $\mathcal{L}^*$, the projection dimensions, the 2.2M-pair training mixture (with citations to all source datasets), the instruction-rewrite probability, the optimizer (AdamW with explicit lr/β/weight decay), the schedule, the per-device batch size and gradient accumulation, the hardware (4×8 A100-80G), the loss weights λ_c, λ_p, and the inference settings (resolution, 40 steps, guidance scale 4.0). Both backbones (Qwen-Image-Edit-2509 and 2511) are publicly released, as are all training datasets.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code release is not guaranteed at this time due to internal constraints. To support reproducibility we instead provide a complete written specification: Sec. 3 fully describes the architecture and three-stage training procedure, and Sec. 4.1 lists the exact training data mixture (all from public sources, with citations), backbone checkpoints (both publicly released), optimizer, schedule, batch sizes, hardware, and inference settings. All datasets and base models used are publicly accessible.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 4.1 specifies, for both the main and ablation runs, the training-data mixture and size, the resolution, optimizer (AdamW with explicit lr, β, weight decay), learning-rate schedule and warmup, per-device batch size, gradient accumulation, number of epochs per stage, loss weights (λ_c, λ_p), classifier-free guidance dropout probability, and instruction-rewrite probability, as well as the test-time settings (number of denoising steps, guidance scale, input resolution).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Main-table benchmark scores are reported as single deterministic numbers (LLM judges at temperature 0), since training a single 2.2M-pair, three-stage run on 32 A100-80G GPUs is too expensive to repeat across seeds. To address judge-side variance, we re-evaluate every strong baseline under one consistent protocol on ImgEdit (Sec. 4.2), and the analysis study (Sec. 4.4) reports per-sample distributions (median and tail probabilities of ρ) over n=600 held-out samples.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Sec. 4.1 specifies the hardware (4 nodes of 8 A100-80G GPUs, i.e., 32 A100-80G in total) and the per-stage schedule (4 epochs each with batch size 4 and gradient accumulation 4), along with the data scale ($\sim$2.2M pairs at 512$\times$512). Ablations use a smaller setup (288K pairs, 1 epoch) on the same hardware.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics. The work uses publicly released image-editing benchmarks and datasets and a publicly released backbone, conducts no human-subject research, and does not collect new personal data. Anonymity is preserved in the submission.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: Our work proposes a training recipe for instruction-based image editing on top of a publicly released backbone (Qwen-Image-Edit). It does not introduce a new capability, dataset, or deployable system beyond what the backbone already provides; the broader societal implications are inherited from the underlying generative-image literature and are not specific to this paper.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a new pretrained model, a new dataset, or any new high-risk asset. Inline Critic is a method that adapts an existing publicly released backbone

(Qwen-Image-Edit) on existing publicly released datasets, so no additional release-time safeguards apply.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All third-party assets are credited via citation in Sec. 3.1 and Sec. 4.1, including the backbone Qwen-Image-Edit [45], the encoder Qwen2.5-VL [3], the rewriter Qwen3.5-27B [37], and every training-data source (OmniEdit, UltraEdit, HQ-Edit, ShareGPT-4o-Image, OpenGPT-4o-Image, UniREdit-Data-100K, Nano-Consistent-150K, and the SFT split of Pico-Banana-400K) as well as every benchmark (RISEBench, KRIS-Bench, GEdit-Bench, ImgEdit). We use these assets in accordance with their public licenses for non-commercial research.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release new datasets, model checkpoints, or other reusable assets at this time.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All evaluations use automated LLM/VLM judges on public benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects research, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We use the publicly released LLM Qwen3.5-27B [37] as an instruction rewriter, both during training (with probability 0.5) and at test time on RISEBench and KRIS-Bench, following the ThinkRL-Edit [23] prompt template (Sec. 4.1). The ablation in Sec. 4.3 (Tab. 6) explicitly disentangles the contribution of this rewrite from the contribution of Inline Critic. LLM use is restricted to instruction rewriting and to LLM-as-judge evaluation; LLMs are not used for writing this paper.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

Appendix

A Details of Implementation

Training. We build Inline Critic on two public Qwen-Image-Edit [45] backbones, Qwen-Image-Edit-2509 and Qwen-Image-Edit-2511, yielding two variants that we refer to as Critic-v1 and Critic-v2. We use $K = 9$ probes at $\mathcal{L}^* = \{6, 12, 18, 24, 30, 36, 42, 48, 54\}$, with 256-dim outputs for $\psi^{(\ell)}$ and $\chi^{(\ell)}$. The training set is a 2.2M-pair mixture from OmniEdit [44], UltraEdit [51], HQ-Edit [17], ShareGPT-4o-Image [7], OpenGPT-4o-Image [8], UniREdit-Data-100K [15], Nano-Consistent-150K [48], and the SFT split of Pico-Banana-400K [35], processed at 512×512. For classifier-free guidance, the instruction is dropped with probability 0.05, and with probability 0.5 replaced by an offline Qwen3.5-27B [37] rewrite using the ThinkRL-Edit [23] prompt. For the main Critic-v1 and Critic-v2 models, each stage runs for 4 epochs with AdamW (lr 1×10^{-4}, $\beta = (0.9, 0.999)$, weight decay 0.01) and a cosine-with-restarts schedule (min-LR ratio 0.1, 500 warmup steps). The per-device batch is 4 with gradient accumulation 4, on 4 nodes of 8 A100-80G GPUs. Stage 3 uses $\lambda_c = \lambda_p = 1$ in Eq. 8.

Testing. We evaluate on RISEBench [52], KRIS-Bench [46], GEdit-Bench [27], and ImgEdit [47], with the same test-time Qwen3.5-27B rewrite applied to Critic-v1 and Critic-v2 on RISEBench and KRIS-Bench. GEdit-Bench and ImgEdit use raw instructions. We resize inputs to a $\sim 1024^2$-pixel budget at the original aspect ratio, and use 40 denoising steps at guidance scale 4.0.

B Qualitative Results

We provide qualitative results in Fig. 5. Starting from the early backbone layers, the probe-layer error maps highlight broad regions that are likely to require modification. As the depth increases from $L6$ to $L54$, these regions gradually become more focused, indicating that the critic refines the model's internal behavior throughout the forward pass. The final outputs demonstrate that this progressive refinement leads to edits that are both instruction-faithful and visually coherent.

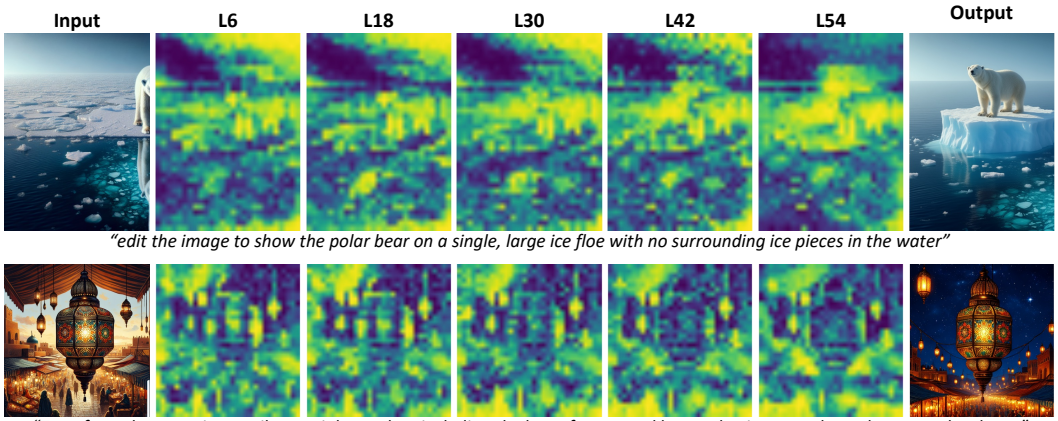

Figure 5: **Qualitative results.** We visualize the probe-layer error maps across backbone layers ($L6 \rightarrow L54$), together with the input image (left) and the final edited result (right).

C Limitations

The critic is trained to predict the per-position MSE of the probe head, but the training data can be highly noisy. Therefore it is not always a faithful measure of generation quality, which leaves the critic's learning signal correspondingly noisy. A natural next step is to replace the MSE target with a feature-space criterion, so that the critic learns perceptually meaningful errors.