COPYING AS DEFENSE: PROTECTING IMAGES FROM EDITING BY ELICITING MODEL COPYING

Anonymous authors
Paper under double-blind review

ABSTRACT

Image editors built on diffusion transformers (DiTs), such as Flux Kontext, make it possible to turn a single photograph into a convincing manipulated image, including harmful and non-consensual edits. Proactive defenses aim to prevent such manipulations by adding an imperceptible perturbation before an image is shared. Existing defenses for DiT-based in-context editors take an intuitive approach: make the output attend *less* to the protected reference image. But this leaves the model free to follow the edit instruction using the text and its own generated tokens, often producing a new, weakly grounded image rather than preventing the edit altogether. We find that the opposite strategy is more effective. Instead of leading the model to ignore the reference, we encourage it to *copy it*. Through controlled attention interventions, we show that concentrating target-token attention on the reference image in only a few early blocks is sufficient to collapse the editing behavior, causing the model to reproduce its input, often as a lossy copy, rather than execute the requested transformation. Based on this insight, we introduce Copying-as-Defense (CaD), which maximizes per-token target-to-reference attention mass in early transformer blocks. Since the attention rows are normalized, increasing reference attention suppresses the attention allocated to the text and generated canvas. Restricting the objective to early blocks also lets us truncate backpropagation path, keeping optimization cheap. Across editors and benchmarks, CaD substantially suppresses instruction-following. It reduces average edit success from 81.4% on unprotected images to 24.0%, a 70.6% relative reduction. Compared with DeContext, the strongest baseline on average at 43.0% edit success, CaD further reduces edit success by 44.3%. CaD is also substantially cheaper to optimize, running roughly 1.8–$2.0\times$ faster per step than comparable optimization-based methods.

1 INTRODUCTION

A photograph posted online can now be transformed into a convincing manipulated image with a single natural-language instruction. The same capability that enables benign editing also lowers the barrier to harmful uses, including non-consensual sexualized imagery, impersonation, misinformation, fraud, and copyright misuse (Li et al., 2026; NBC News, 2026; Marchal et al., 2024). One response has been to protect an image before it is shared by adding a small, imperceptible perturbation designed to make downstream edits fail. This idea predates modern diffusion editors, with work disrupting GAN-based facial manipulation systems (Ruiz et al., 2020), and has since expanded to diffusion-based editing (Salman et al., 2023; Choi et al., 2025), unauthorized personalization Marchal et al. (2024), and artistic imitation (Liang et al., 2023; Shan et al., 2023; Hönig et al., 2025). As generative architectures evolve, however, the mechanism a defense must disrupt evolves with them.

That shift is especially important for the latest generation of diffusion transformers (DiTs) (Peebles & Xie, 2023; Chen et al., 2024; Ma et al., 2024). A DiT replaces the convolutional U-Net backbone with a transformer operating on latent patches (Peebles & Xie, 2023). Multimodal DiTs (MMDiT), introduced with Stable Diffusion 3, instead use separate weights for text and image tokens and let information flow in both directions between them (Esser et al., 2024). Models such as Flux Kontext and FIBO Edit mix such *double-stream* blocks, which keep modality-specific weights and mix tokens through attention over their concatenation, with *single-stream* blocks that subsequently process the concatenated sequence (Black Forest Labs, 2025; Gutflaish et al., 2025). Editors built on this design condition on a reference image by concatenating its tokens with the noisy output

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

canvas along the sequence dimension. For adversarial protection, this architectural change matters twice because the representation we want to disrupt has changed, and backpropagating through a multi-billion-parameter editor is expensive.

Recent work therefore attacks the mechanism through which an in-context editor reads its reference. DeContext (Shen et al., 2025) suppresses target-to-reference attention, aiming to *decontextualize* generation. The intuition is that if the output cannot attend to the protected image, then it cannot edit it. But should protection actually make the editor forget the reference?

We find that reference suppression is not only insufficient, but can be counterproductive. Each softmax attention row sums to one, so reducing attention to reference tokens reallocates probability mass to the text instruction and the evolving target canvas, which are precisely the pathways that support instruction following and generation. A successfully "decontextualized" editor can therefore still satisfy a malicious prompt by generating content that is less grounded in the protected input. Moreover, recent interpretability work on FLUX.2 shows that text tokens absorb reference content within the first few blocks and carry it to the output even when the reference is removed from the computation afterwards (Ge et al., 2026). Suppression applied after this point therefore cannot remove what has already been absorbed, so a conservative defense should avoid this residual leakage altogether rather than attenuate it downstream. This failure is particularly visible for edits the model can plausibly generate without relying on the image at all (*i.e.*, ones with strong generative prior) . It also fails on full-body photographs, where the person, and especially sensitive identity such as the face, occupies only a small fraction of the reference tokens.

We therefore invert the objective. Rather than optimizing a perturbation that causes the editor to use the protected image *less*, we ask what happens if it is forced to use the image *more*. Our controlled interventions on Flux Kontext reveal that forcing target queries to place their attention on reference keys in only a few early double-stream blocks causes the model to reproduce the reference rather than execute the edit. Conversely, explicitly removing reference attention does not prevent generation. Even with a black reference image, the same qualitative distinction exists, *i.e.*, forcing target⇒reference attention yields the black input itself, whereas suppressing that attention still lets the model generate a new image. An output we want is therefore not forgetting, but *copying*.

This observation motivates CaD. For every target query, CaD drives target-to-reference attention mass toward one in a small set of early double-stream blocks. The objective has three useful consequences. First, attention normalization makes the reference compete directly with the text and target tokens, suppressing instruction following without needing to predict the future malicious prompt. Second, the objective is per token, preventing small identity regions from being washed out by an image-level average. Third, early-layer targeting makes the protection computationally efficient by construction. Once the last protected block is reached, we can truncate the forward pass at the last selected block and backpropagate only through this shallow set of layers, rather than differentiating through the billion-scale model. This shortened backpropagation path substantially reduces the cost of every optimization step.

Our contributions are: **First**, we identify two failures of attention-suppression defenses for multimodal DiT editors: *mass redistribution*, which leaves the text and target pathways free to render instruction-adjacent outputs similar to text-to-image generation with weak or no image conditioning, and *token coverage*, which makes image-averaged suppression poorly suited to full-body images. **Second**, through causal attention interventions, we show that the opposite direction is far more effective, *i.e.*, forcing early target-to-reference attention makes the editor copy its reference rather than edit it. **Third**, we introduce CaD, a per-token copy objective applied to early double-stream blocks. This placement is central to the method as it both induces copying and lets us truncate the model after the selected blocks, dramatically shortening the forward/backward path for each protection step. **Lastly**, we show that CaD substantially improves protection over existing baselines while retaining comparable imperceptibility and reducing protection time.

2 Related Work

Different harms motivate different protections. There are many ways an unauthorized image edit can be harmful, and consequently many ways researchers have tried to protect images from such edits. When the concern is misuse of a person's identity or face, prior work has proposed protections against

facial manipulation and personalized generation (Le et al., 2023; Liu et al., 2024). When the concern is artistic style mimicry, methods such as Glaze (Shan et al., 2023), Mist (Liang & Wu, 2023; Zheng et al., 2025), and related approaches protect an artist's work from being used to reproduce their style. Other methods target more general image manipulation, including inpainting and instruction-based editing (Salman et al., 2023; Choi et al., 2025; Jeon et al., 2025). In other words, "protecting an image" is not a single objective because what should be disrupted depends on what an attacker is trying to preserve, modify, or generate. At the same time, a growing body of work has questioned how much protection these methods provide outside the particular threat models for which they were designed (Li et al., 2026; Hönig et al., 2025; Radiya-Dixit et al., 2021). Protections that are effective against one manipulation may offer little protection against another, making broad claims of image protection difficult.

Protection is a moving target. Two broader challenges make proactive protection particularly difficult. First, generative models continue to evolve. A protection designed around one model family may no longer target the representations or mechanisms used by its successors, requiring defenses to evolve alongside the models they are intended to protect. Second, adversarial perturbations are inherently vulnerable to transformations of the protected image. Even ordinary processing, such as JPEG compression, resizing, or added noise, can weaken a perturbation, while an adversary can intentionally apply more sophisticated purification procedures designed to remove it (Hönig et al., 2025). These limitations do not eliminate the value of proactive protection, but they make it important to be precise about both the model and the manipulation against which a defense is evaluated.

Protecting images from modern DiT editors. Recent image editors increasingly replace the UNet backbone used by earlier latent diffusion models with diffusion transformers (DiTs) and related multimodal transformer architectures (Peebles & Xie, 2023; Esser et al., 2024). Models such as FLUX.1 Kontext (Black Forest Labs, 2025), Step1X-Edit (Liu et al., 2025), and FIBO Edit (Gutflaish et al., 2025) perform instruction-based editing by jointly processing the reference image, the text instruction, and the image being generated. To the best of our knowledge, DeContext (Shen et al., 2025) is the only existing proactive protection method designed specifically around this new class of in-context DiT editors. This does not mean that earlier protection methods cannot be applied to newer editors. Rather, they were not designed specifically for their architecture.

We therefore evaluate representative approaches from several existing families after adapting them to FLUX.1 Kontext and FIBO Edit. PhotoGuard (Salman et al., 2023) perturbs the image so that its latent under the model's VAE matches that of a mid-gray image. DeContext (Shen et al., 2025) minimizes the attention that the generated image's tokens place on the reference image's tokens; for FIBO Edit, which takes structured JSON prompts, a VLM is called once per edit instruction on the clean image to produce them. Diff-PGD proposed as a baseline (Shen et al., 2025) maximizes the flow-matching error the model makes when reconstructing a clean edit from the perturbed reference image, where the clean edit for each instruction is generated once with the editor itself and cached; for FIBO Edit, each edit is generated from a VLM-produced JSON prompt, so this also requires one VLM call per instruction. Our method also operates on attention, but starts from a different observation about how reference information should be manipulated. Rather than suppressing the reference image, we ask whether the model can be made to rely on it *more*. This distinction motivates the analysis in Sections 3–5.

3 PRELIMINARIES

Threat model. Let an image owner hold the image $\boldsymbol{x} \in [0,1]^{H\times W\times 3}$ and release a protected image $\boldsymbol{x}_{\text{adv}} = \boldsymbol{x} + \delta$, where $\|\delta\|_\infty \le \epsilon$. An attacker later supplies $\boldsymbol{x}_{\text{adv}}$ together with an editing instruction c (and, when required by the editing pipeline, a mask) to an editor G. A protection succeeds when the resulting edit fails to realize the requested transformation on the depicted subject.

Optimization-based protection. We follow the standard per-image adversarial optimization template,

$$\delta^\star = \arg\max_{\|\delta\|_\infty \le \epsilon} \mathcal{L}_{\text{protect}}(\boldsymbol{x} + \delta), \tag{1}$$

optimized with projected gradient ascent. Existing methods differ primarily in $\mathcal{L}_{\text{protect}}$: encoder attacks alter the VAE representation, denoiser attacks disrupt the diffusion or flow prediction, and



(a) Face close-up: 92% subject tokens **(b)** Full body: 31% subject tokens

Figure 1: Reference suppression is less reliable when the protected subject occupies fewer image tokens. We overlay the Flux Kontext image-token grid on the protected inputs. Orange outlines mark majority-subject tokens and the blue box marks face tokens. Suppression disrupts the face-centric edits in (a), while both requested edits remain successful on the full-body subject in (b).

more recent white-box attacks directly manipulate the model's attention patterns. We provide the objectives and implementation details of the baselines used in our experiments in the appendix.

In-context editing with multimodal DiTs. We study two DiT-based in-context editors, Flux Kontext and FIBO. Despite architectural differences, both condition generation jointly on an editing instruction and a reference image. Their joint-attention computation operates over three token groups: text tokens C, noisy target-image tokens T, and reference-image tokens R. For a target query $q \in T$, head h in block b assigns attention

$$A^{(b)}_{h,q,k} = \mathrm{softmax}_k\left(\frac{(q^{(b)}_{h,q})^\top k^{(b)}_{h,k}}{\sqrt{d}}\right), \tag{2}$$

with its attention mass partitioned across the three groups:

$$\underbrace{\sum_{k\in C} A^{(b)}_{h,q,k}}_{\text{text}} + \underbrace{\sum_{k\in T} A^{(b)}_{h,q,k}}_{\text{target}} + \underbrace{\sum_{k\in R} A^{(b)}_{h,q,k}}_{\text{reference}} = 1. \tag{3}$$

This row-normalization constraint is central to our method: increasing the mass that target queries assign to reference tokens necessarily reduces the mass available to text and target tokens. We provide further architectural details for both editors in the appendix.

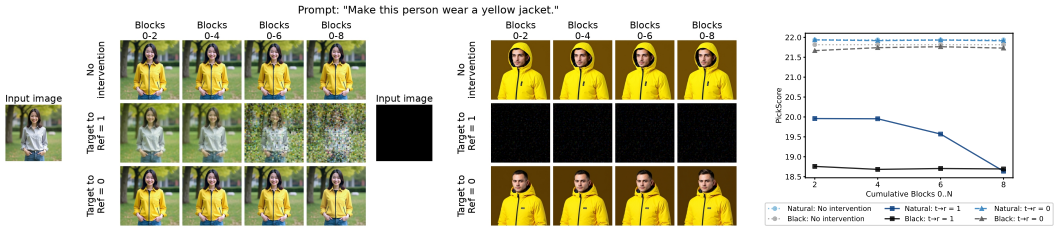
4 Rethinking Reference Attention

4.1 Why Suppressing Reference Attention Is Not Enough

DeContext suppresses the average attention mass that target queries place on reference keys. We find that its effectiveness depends strongly on image composition. As shown in Fig. 1, suppression can disrupt identity preservation for a tightly cropped face, yet still permit successful edits of a protected full-body subject.

A key reason is **token coverage**. In Flux Kontext, the VAE downsamples spatially by $8\times$ and its latents are packed into 2×2 patches, so a 512×512 image produces a 32×32 grid of 1024 image tokens, each corresponding to a 16×16 pixel region. In Fig. 1(a), 944 of 1024 tokens (92%) belong primarily to the subject, so an image-averaged suppression objective necessarily acts on subject information. In Fig. 1(b), only 322 tokens (31%) belong primarily to the person and just $7 \times 7 = 49$ tokens (4.8%) cover the face. This is representative of BodyBench more broadly: the face-and-body mask occupies only $28.45\% \pm 6.06\%$ of the image area on average. Thus, an objective averaged over all target–reference interactions can be driven down largely through background tokens while leaving comparatively few identity-critical interactions intact.

Two additional properties make reference suppression an incomplete objective. First, **attention mass is redistributed**: by Eq. 3, reducing reference attention necessarily reallocates mass to text and target keys. The editor can therefore remain strongly driven by the instruction and its evolving target representation even when grounding in the reference is weakened. Second, **reference information can move between token groups**. Recent analysis of FLUX.2 shows that text tokens absorb visual information from the reference within early blocks and can subsequently carry that information



(a) Qualitative intervention results. **(b)** PickScore on BodyBench.

Figure 2: Increasing target-to-reference attention induces copying, whereas suppressing it does not prevent editing. We intervene cumulatively over increasingly deep prefixes of Flux Kontext's early double-stream blocks. Target→Ref $= 0$ removes reference attention, while Target→Ref $= 1$ places all target-query attention on reference keys. **(a)** Suppression still permits instruction-consistent generation, whereas maximizing reference attention increasingly reproduces the natural or black reference. **(b)** On 10 BodyBench images, PickScore remains high and nearly flat for no intervention and Target→Ref $= 0$, but drops under Target→Ref $= 1$ as more blocks are intervened on.

toward the output (Ge et al., 2026). Removing a direct target-to-reference interaction therefore does not imply that reference information has been removed from the computation.

These observations suggest that the central question is not merely *how much* reference attention to manipulate, but in *which direction*. We perform a controlled intervention where everything is held fixed except whether target-to-reference attention is removed or concentrated onto the reference.

4.2 Manual intervention: minimizing versus maximizing reference attention

We intervene directly on target-query attention rows during inference. For target query i, let α_{ij} be its softmax attention over keys j, which are partitioned into text, target, and reference index sets $\mathcal{I}_{\text{txt}}$, $\mathcal{I}_{\text{tgt}}$, and $\mathcal{I}_{\text{ref}}$, and let (a_i, b_i, r_i) denote the total mass it assigns to each group, so that $a_i + b_i + r_i = 1$. A *minimize* intervention sets the reference mass to zero and renormalizes the remaining groups,

$$\tilde{\alpha}_{ij} = \begin{cases} \alpha_{ij}/(a_i + b_i), & j \notin \mathcal{I}_{\text{ref}}, \\ 0, & j \in \mathcal{I}_{\text{ref}}, \end{cases} \qquad (a_i, b_i, r_i) \mapsto \left(\tfrac{a_i}{a_i+b_i}, \tfrac{b_i}{a_i+b_i}, 0 \right). \tag{4}$$

A *maximize* intervention instead places all mass on the reference while preserving the within-reference distribution, *i.e.*, the reference block of the row is renormalized to sum to one,

$$\tilde{\alpha}_{ij} = \begin{cases} \alpha_{ij}/r_i, & j \in \mathcal{I}_{\text{ref}}, \\ 0, & j \notin \mathcal{I}_{\text{ref}}, \end{cases} \qquad (a_i, b_i, r_i) \mapsto (0, 0, 1). \tag{5}$$

In both cases the modified row is still a valid distribution (it sums to one). We apply each intervention cumulatively over increasingly deep prefixes of the double-stream transformer; for example, blocks 0–6 means that the intervention is applied in every double-stream block from 0 through 6.

Figure 2 makes the contrast between the two intervention directions visually clear. Removing target-to-reference attention does not stop the model from producing plausible edits that follow the instruction. Forcing target-to-reference attention, however, progressively changes the output from an edit into a reproduction of the reference as the intervention is applied across more early blocks. The principle holds even when the input image is a purely black image (the *black-reference control* in Fig. 2a): suppressing reference attention still allows the model to generate instruction-consistent content, whereas forcing reference attention causes the output to collapse toward the black reference.

We quantify the same behavior using 10 images from BodyBench in Fig. 2b. PickScore remains high and nearly flat with no intervention or with Target→Ref $= 0$, but drops substantially under Target→Ref $= 1$ as the intervention is extended through more blocks, confirming that stronger early reference attention increasingly suppresses successful editing.

5 Copying as Defense

The previous section suggests a reversal of reference suppression. Rather than minimize target-to-reference attention, we maximize it for every target query. Let $\mathcal{B}$ denote the selected early

double-stream blocks and let $A^{(b)}_{h,q,k}$ be the attention from target query $q \in T$ to reference key $k \in R$. We define the reference mass for each query and head as

$$m^{(b)}_{h,q} = \sum_{k \in R} A^{(b)}_{h,q,k}. \tag{6}$$

Our protection objective is

$$\mathcal{L}_{CaD}(\boldsymbol{x} + \delta) = -\frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \frac{1}{H|T|} \sum_{h=1}^{H} \sum_{q \in T} \left(1 - m^{(b)}_{h,q}\right)^2. \tag{7}$$

Maximizing Eq. 7 drives $m^{(b)}_{h,q} \to 1$ for each target query. Each design choice addresses a failure observed above. **Direction:** maximizing reference mass exploits softmax normalization to reduce the attention available to text and target keys, directly opposing the text and target pathways that would otherwise drive a pure text-to-image generation. **Per-token objective:** each target query is penalized separately, so a small set of identity-critical tokens cannot be hidden by a large, well-behaved background. **Early blocks:** the intervention study shows that early double-stream blocks are sufficient to induce copying. Restricting the loss to these blocks also changes the computational graph itself. Because no downstream activation contributes to the objective, we stop the forward pass immediately after $\max \mathcal{B}$ and backpropagate only through this prefix. Thus, unlike denoiser-level objectives or DeContext-style objectives that require substantially deeper execution, our gradient never traverses the remainder of the transformer. Early-layer targeting therefore provides both the mechanism that induces copying and the mechanism that makes protection fast. Additional details are specified in the experimental setup and appendix.

5.1 Experimental Setup

Benchmarks. We evaluate on InpaintGuardBench (Choi et al., 2025) and BodyBench (Li et al., 2026). InpaintGuardBench contains real, face-centric portraits in which the subject typically occupies most of the frame. We retain only images with a clearly visible identity and exclude unsuitable examples such as side-profile images, leaving 20 images. BodyBench contains synthetic full-body photographs, each paired with 15 editing prompts spanning diverse redressing and scene-resetting edits. We randomly sample 30 BodyBench images. This gives 50 protected images per experimental configuration and, with 15 edits per image, 750 edited outputs per configuration. We evaluate the protections in terms of both effectiveness, perceptibility, and computational efficiency.

Metrics. We evaluate protection effectiveness using Edit Success Rate (ESR), PickScore, CLIP directional similarity (CLIP_{dir}), and ArcFace similarity. ESR is the fraction of edited outputs judged successful by Qwen3-VL-30B-A3B-Instruct (Bai et al., 2025), which jointly considers the original image, edited image, edit instruction, and intended edited description. An edit is successful only if it realizes the requested transformation while preserving unrelated content and identity and avoiding major visual artifacts. PickScore (Kirstain et al., 2023) measures alignment between the edited image and a target caption describing the requested edit. CLIP_{dir} (Gal et al., 2022) measures whether the change from the original to the edited image follows the corresponding semantic change in text embedding space. ArcFace (Deng et al., 2019) measures identity similarity between the faces in the original and edited images. Lower values indicate stronger protection for all four metrics. We measure protection perceptibility between the original image and protected image using LPIPS (Zhang et al., 2018) and PSNR, where lower LPIPS and higher PSNR indicate a less perceptible perturbation. We additionally report computational efficiency using the median time per optimization step, T_S in seconds, and the resulting optimization time per protection, T_P in minutes.

Editors and baselines. We evaluate on two instruction-based DiT editors, FLUX.1 Kontext-dev (Black Forest Labs, 2025) and FIBO Edit (Gutflaish et al., 2025). We compare CaD against three representative proactive defenses: PhotoGuard (Salman et al., 2023), DeContext (Shen et al., 2025), and Diff-PGD (Shen et al., 2025). We adapt each baseline to the corresponding editor while preserving its original protection objective and use the editor-specific conditioning required by each model, including FIBO's structured JSON prompts. All methods are evaluated at $\epsilon \in \{4, 8, 12, 16\}/255$. Full objective definitions and model-specific implementation details are provided in the appendix.

Table 1: Protection performance across datasets and image editing models for $\epsilon = 12/255$. **Blue bold** indicates the best result and light blue indicates the second-best result. Runtime is measured per optimization step; total protection time uses 800 steps for the baselines and 1000 steps for ours.

			Edit Metrics			Identity	Image Metrics		Runtime	
Editor	**Dataset**	**Method**	**PickScore ↓**	**CLIP$_{\text{dir}}$ ↓**	**Edit Succ. ↓**	**ArcFace ↓**	**LPIPS ↓**	**PSNR ↑**	T_S ↓	T_P ↓
FIBO	InpaintGuard Bench	Unprotected	20.68	0.2428	0.800	0.9438	–	–	–	–
		PhotoGuard	20.58	0.2112	0.823	0.7870	**0.1497**	29.77	**0.057**	**0.76**
		DeContext	20.46	0.1971	0.597	0.7258	0.3599	29.48	1.385	18.47
		Diff-PGD	19.89	0.1751	0.527	0.7881	0.3240	**30.22**	1.531	20.41
		Ours	**19.56**	**0.1193**	**0.183**	**0.1627**	0.3070	28.76	0.754	12.57
	BodyBench	Unprotected	22.05	0.3408	0.818	0.8397	–	–	–	–
		PhotoGuard	21.80	0.3120	0.813	0.4561	**0.1534**	29.98	**0.057**	**0.76**
		DeContext	21.62	0.2784	0.664	0.4206	0.4343	29.42	1.385	18.47
		Diff-PGD	20.62	**0.2090**	**0.476**	0.4646	0.3894	**30.14**	1.531	20.41
		Ours	**20.51**	0.2115	0.564	**0.1891**	0.3815	28.73	0.754	12.57
FLUX Kontext	InpaintGuard Bench	Unprotected	21.05	0.2580	0.820	0.9118	–	–	–	–
		PhotoGuard	20.89	0.2467	0.780	0.8271	**0.1314**	29.02	**0.039**	**0.52**
		DeContext	21.84	0.2044	0.210	0.3478	0.3438	29.88	1.232	16.43
		Diff-PGD	20.48	0.1981	0.520	0.7524	0.3359	**30.15**	1.245	16.60
		Ours	**18.65**	**0.0513**	**0.160**	**0.3302**	0.3130	29.73	0.647	10.78
	BodyBench	Unprotected	22.42	0.3507	0.816	0.7817	–	–	–	–
		PhotoGuard	22.28	0.3378	0.818	0.5700	**0.1152**	29.34	**0.039**	**0.52**
		DeContext	22.29	0.2711	0.249	0.2823	0.4238	29.70	1.232	16.43
		Diff-PGD	21.38	0.2574	0.442	0.5204	0.4099	**30.14**	1.245	16.60
		Ours	**18.66**	**0.0359**	**0.051**	**0.2651**	0.3886	29.80	0.647	10.78

6 RESULTS

Table 1 and Fig. 3 summarize our main results at $\epsilon = 12/255$. Across both editors, CaD strongly disrupts the requested edit without requiring larger perturbations than the other optimization-based defenses. The clearest gains appear on Flux Kontext. On BodyBench, CaD reduces edit success to 5.1%, compared with 24.9% for DeContext and 44.2% for Diff-PGD. CLIP$_{\text{dir}}$ falls to 0.0359, far below the next-best protected result of 0.2574. We see the same pattern on InpaintGuardBench, where CaD reaches 16.0% edit success and a CLIP$_{\text{dir}}$ score of 0.0513. The qualitative examples in Fig. 3 match this behavior: while the baselines often still carry out the requested transformation, our protected outputs tend to remain close to the original input. We note the decontextualization behavior of DeContext in the third row of Fig. 3 where the identity of the original subject is not present in the edit, yet the unconditional generation (as if using a black reference) is still generated. Based on the intuition discussed in section 4, we would still see to avoid this behavior.

Our method also successfully protects images when optimized for FIBO Edit. On InpaintGuardBench, CaD reduces edit success from 80.0% on unprotected images to 18.3%, compared with 52.7% for Diff-PGD and 59.7% for DeContext. On BodyBench, the result is more mixed: CaD gives the lowest PickScore and ArcFace similarity, while Diff-PGD performs slightly better on CLIP$_{\text{dir}}$ and edit success. As Fig. 3 shows, failed FIBO edits do not always resemble clean copies of the input and can instead contain visible artifacts further supporting our lossy compression hypothesis.

Protection across perturbation budgets. We next vary the perturbation budget and compare edit success against LPIPS. Figure 4 shows that the advantage is not specific to $\epsilon = 12/255$. On both datasets, CaD reaches substantially lower edit success at comparable LPIPS. The gap is especially large at moderate budgets. On BodyBench, edit success is already near zero at $\epsilon = 8/255$, while DeContext and Diff-PGD remain above 50%.

Efficiency. Because our objective acts only on early double-stream blocks, the forward pass can stop after the last selected block and gradients do not traverse the remainder of the transformer. This substantially reduces both the time per optimization step (T_S) and the total time to construct a protection (T_P). On Flux Kontext, CaD requires 0.647 s per step and 10.78 min per protection, compared with 1.232 s/16.43 min for DeContext and 1.245 s/16.60 min for Diff-PGD. The same advantage holds on FIBO, where CaD takes 0.754 s per step and 12.57 min per protection. Figure 5 further shows that the full optimization budget is often unnecessary: by 300 steps, CaD already outperforms the final 800-step baselines across all four protection metrics, corresponding to only about 3.2 minutes of optimization on Flux Kontext.



Figure 3: Qualitative comparison across editors and datasets. We show FLUX Kontext on BodyBench (top) and FIBO Edit on InpaintGuardBench (bottom), with all protections using $\epsilon = 12/255$. Existing defenses fail to prevent the edit, while our method prevents the requested transformation.

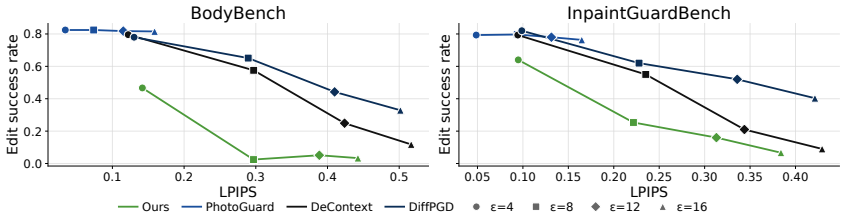


Figure 4: Protection remains strong across perturbation budgets. Edit success versus LPIPS for FLUX Kontext at $\epsilon \in \{4, 8, 12, 16\}/255$. Lower is better on both axes. Across budgets, our method achieves substantially lower edit success than the baselines at comparable LPIPS.

6.1 Ablations

Block selection. Our intervention study suggests that the copying behavior is established early in the network. We therefore vary how many consecutive double-stream blocks are included in the

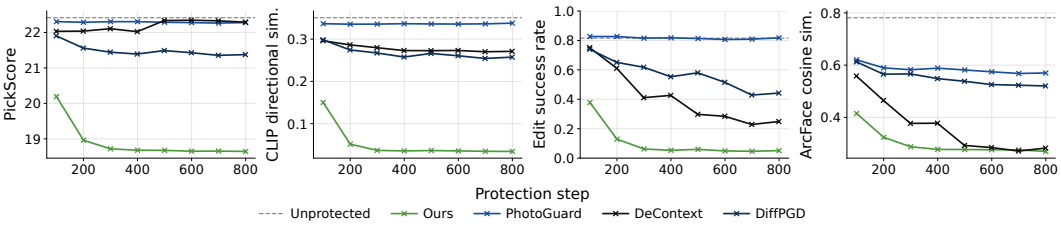


Figure 5: CaD reaches strong protection within the first few hundred optimization steps. Protection metrics on BodyBench with FLUX Kontext at $\epsilon = 12/255$. Lower is better for all metrics.

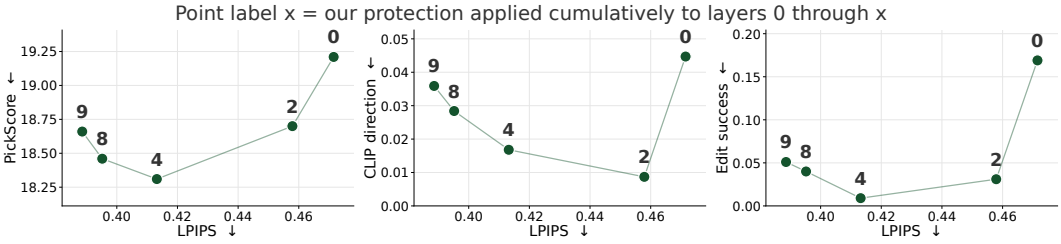


Figure 6: Early double-stream blocks are sufficient for strong protection. Each point x applies our objective cumulatively to blocks 0 through x on BodyBench. Lower is better on both axes.

protection objective. Figure 6 shows that optimizing only block 0 is weaker, while extending the objective through block 4 sharply decreases PickScore, CLIP_{dir}, and edit success. Using additional layers may not continue to improve protection. Instead, we trade a small amount of protection strength for lower LPIPS. This supports using these shallower prefix rather than using the full transformer.

Objective direction. We finally isolate the direction of the attention objective. Both variants use the same double blocks, perturbation budget, and number of PGD steps; the only difference is whether target-to-reference attention is minimized or maximized. Minimizing reference attention barely changes edit success, from 81.6% on the unprotected images to 80.7%. Maximizing the same attention instead reduces edit success to 5.1%, while also producing much larger drops in PickScore and CLIP_{dir}. This matches the intervention result from Section 4. The important choice is not simply to manipulate reference attention, but to increase it enough to push the model toward copying.

7 Conclusion

Modern in-context image editors make heavy use of the reference image, so existing defenses have naturally tried to protect that image by making the model attend to it less. We find that this is often the wrong direction. Suppressing reference attention can leave the model free to follow the instruction, particularly when the protected subject occupies only a small fraction of the image. Controlled interventions reveal a different behavior when reference attention is increased. Once target tokens are made to rely strongly on the reference in the early transformer blocks, the model begins to reproduce the input instead of editing it. We turn this observation into CaD, a protection objective that increases target-to-reference attention for each target token in a small set of early blocks. Across FLUX Kontext and FIBO Edit, CaD provides stronger protection than existing baselines while maintaining comparable perturbation quality, and its early-layer objective makes optimization considerably cheaper. Our results suggest that protecting images from modern editors need not mean hiding the reference from the model. Making the model rely on it more can itself be a defense.

Limitations. Our evaluation is limited to two DiT-based editors. Like other optimization-based proactive defenses, CaD also assumes white-box access to the protected editor during perturbation construction and requires per-image optimization, despite being substantially cheaper than deeper attention- or denoiser-based objectives. Finally, adversarial image protections can be weakened by common transformations such as compression, resizing, or noise, and by deliberate purification procedures; we do not evaluate robustness against such adaptive preprocessing or attacks here.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementation and debugging of methods and editing portions of the manuscript for clarity, structure, and presentation. In particular, AI tools were used to help implement model implementations and attention intervention mechanisms and develop visualization code. We did not use generative AI to generate the benchmark datasets used in our experiments or to formulate or prove mathematical theorems; other required-disclosure tasks were not applicable to this work. All AI-assisted research code was reviewed by at least one author. Experimental results reported in the paper were obtained from our implemented evaluation pipeline rather than generated by an AI system, and at least one author visually inspected the resulting outputs and analyses. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024.

June Suk Choi, Kyungmin Lee, Jongheon Jeong, Saining Xie, Jinwoo Shin, and Kimin Lee. DiffusionGuard: A robust defense against malicious diffusion-based image editing. In *International Conference on Learning Representations (ICLR)*, 2025.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

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, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

Rinon Gal, Or Patashnik, Haggai Maron, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. StyleGAN-NADA: CLIP-guided domain adaptation of image generators. *ACM Transactions on Graphics (SIGGRAPH)*, 2022.

Chris Ge, Rohit Gandikota, Antonio Torralba, and Tamar Rott Shaham. Vision-language binding in in-context image generation. In *Mechanistic Interpretability Workshop at ICML 2026*, 2026.

Eyal Gutflaish, Eliran Kachlon, Hezi Zisman, Tal Hacham, Nimrod Sarid, Alexander Visheratin, Saar Huberman, Gal Davidi, Guy Bukchin, Kfir Goldberg, et al. Generating an image from 1,000 words: Enhancing text-to-image with structured captions. *arXiv preprint arXiv:2511.06876*, 2025.

Robert Hönig, Javier Rando, Nicholas Carlini, and Florian Tramèr. Adversarial perturbations cannot reliably protect artists from generative ai. In *International Conference on Learning Representations*, volume 2025, pp. 70223–70263, 2025. URL https://openreview.net/forum?id=yfW1x7uBS5.

Joonsung Jeon, Woo Jae Kim, Suhyeon Ha, Sooel Son, and Sung-eui Yoon. AdvPaint: Protecting images from inpainting manipulation via adversarial attention disruption. In *International Conference on Learning Representations (ICLR)*, 2025.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Thanh Van Le, Hao Phung, Thuan Hoang Nguyen, Quan Dao, Ngoc N. Tran, and Anh Tran. Anti-DreamBooth: Protecting users from personalized text-to-image synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Qiwei Li, Salma Abdel Magid, Olga Russakovsky, Eric Gilbert, and Sarita Schoenebeck. Bodybench: Evaluating adversarial image defenses against ai nudification inpainting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2026.

Chumeng Liang and Xiaoyu Wu. Mist: Towards improved adversarial examples for diffusion models, 2023.

Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. In *International Conference on Machine Learning (ICML)*, 2023.

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.

Yixin Liu, Chenrui Fan, Yutong Dai, Xun Chen, Pan Zhou, and Lichao Sun. MetaCloak: Preventing unauthorized subject-driven text-to-image diffusion-based synthesis via meta-learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24219–24228, 2024.

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 (ECCV)*, 2024.

Nahema Marchal, Rachel Xu, Rasmi Elasmar, Iason Gabriel, Beth Goldberg, and William Isaac. Generative AI misuse: A taxonomy of tactics and insights from real-world data, 2024. URL https://arxiv.org/abs/2406.13843.

NBC News. Musk's AI chatbot Grok is still making sexual deepfakes. https://www.nbcnews.com/tech/tech-news/musks-ai-chatbot-grok-xai-making-sexual-deepfakes-imagine-rcna265855, April 2026. Accessed: 2026-09-24.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023.

Evani Radiya-Dixit, Sanghyun Hong, Nicholas Carlini, and Florian Tramer. Data poisoning won't save you from facial recognition. *arXiv preprint arXiv:2106.14851*, 2021.

Nataniel Ruiz, Sarah Adel Bargal, and Stan Sclaroff. Disrupting deepfakes: Adversarial attacks against conditional image translation networks and facial manipulation systems. In *European conference on computer vision*, pp. 236–251. Springer, 2020.

Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious AI-powered image editing. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 29894–29918, 2023.

Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y. Zhao. Glaze: Protecting artists from style mimicry by text-to-image models. In *USENIX Security Symposium*, 2023.

Linghui Shen, Mingyue Cui, and Xingyi Yang. DeContext as defense: Safe image editing in diffusion transformers. *arXiv preprint arXiv:2512.16625*, 2025.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Boyang Zheng, Chumeng Liang, and Xiaoyu Wu. Targeted attack improves protection against unauthorized diffusion customization. In *International Conference on Learning Representations (ICLR)*, 2025.

A Appendix

B Protection Methods and Their Instantiation on Each Editing Model

We compare four protection methods (PhotoGuard, DeContext, Diff-PGD, and ours) on two image-editing models, FLUX.1 Kontext-dev and FIBO Edit. All methods perturb the *context/reference* (to-be-protected) image x with the same projected gradient descent (PGD) procedure and differ only in the loss they optimize.

Common PGD setup. Images are resized and center-cropped to 512×512 and normalized to $[-1, 1]$. The perturbed image x' is kept in `float32` while the models run in `bfloat16`, so the perturbation is not lost to quantization. Each step performs a signed-gradient update followed by projection onto the ℓ_∞ ball of radius ε around x and onto the valid image range:

$$x' \leftarrow \mathrm{clip}_{[-1,1]}\Big(x + \mathrm{clip}_{[-\varepsilon,\varepsilon]}\left(x' + \alpha\ \mathrm{sign}(\nabla_{x'}\mathcal{L}) - x\right)\Big), \tag{8}$$

with $\alpha = 0.005$ in the $[-1, 1]$ scale. We use 800 steps for PhotoGuard, DeContext and Diff-PGD, and 1000 for our method. The sign in equation 8 is $+$ when we ascend a loss and $-$ when we descend one; each method below states which.

Notation. E is the model's VAE encoder (posterior mode/mean), F_θ its diffusion transformer, p an edit instruction from a pool $\mathcal{P}$ of 60 instructions directly from Shen et al. (2025) (expressions, accessories, poses, style and scene changes, and combinations), and t a timestep drawn uniformly from $\{980, \ldots, 999\}$, *i.e.* nearly pure noise. Both models are flow-matching models with noisy latent $z_t = (1 - \sigma_t)z + \sigma_t n$, $n \sim \mathcal{N}(0, I)$. In both models the transformer processes a joint sequence of tokens $[\,T \mid Z \mid C\,]$: text tokens T, target (to-be-generated) image tokens Z, and context image tokens $C = \mathrm{tokens}(E(x'))$. At 512×512 each image stream has $N = 1024$ tokens.

B.1 Model-specific components

FLUX.1 Kontext-dev. The VAE is a 16-channel KL autoencoder with $8\times$ spatial downsampling; latents are shifted and scaled by scalar factors and packed into 2×2 patches. Text conditioning is a pooled CLIP embedding plus 512 T5 tokens.

FIBO Edit. The VAE is the Wan 2.2 video VAE (48 latent channels, $16\times$ downsampling, no patch packing), applied to a single frame; latents are normalized per channel, $z = (z_{\text{raw}} - \mu)/\sigma$. Text conditioning uses a SmolLM3 encoder, whose hidden states and a per-block text layer are injected into the transformer, together with a joint additive attention mask over $[T \mid Z \mid C]$; the text length varies with the prompt and there is no guidance embedding. FIBO Edit expects a structured JSON prompt, so for each (image, instruction) pair we obtain a JSON description from the FIBO prompt-to-JSON vision-language model (VLM) applied to the *clean* image, and cache it.

B.2 PhotoGuard (encoder attack)

PhotoGuard attacks only the VAE encoder and does not involve the transformer or any prompt. It starts from a random point $x'_0 = \mathrm{clip}(x + u)$, $u \sim \mathcal{U}(-\varepsilon, \varepsilon)$, and *descends*

$$\mathcal{L}_{\text{PG}}(x') = \left\|E(x') - E(x_{\text{gray}})\right\|_2, \tag{9}$$

which pushes the encoding of x' towards that of a mid-gray image $x_{\text{gray}} = 0$. The step size decays linearly from α to $\alpha/100$ over the run. *FLUX:* E is the FLUX KL-VAE. *FIBO:* E is the Wan VAE.

B.3 DeContext (context-attention suppression)

DeContext (Shen et al., 2025) lowers the attention that generated tokens pay to the context image. At each step we sample $p \in \mathcal{P}$ and t, feed the transformer the packed sequence of pure noise for the target stream and the tokens of $E(x')$ for the context stream, and hook the query/key tensors after the QK-normalization of the first 26 single-stream blocks $\mathcal{S}$. After applying rotary embeddings, the attention of target queries over all keys is

$$A^{(l,h)} = \mathrm{softmax}\Big(\frac{Q_Z^{(l,h)} K^{(l,h)\top}}{\sqrt{d}} [+ M]\Big), \tag{10}$$

and the context proportion of layer l is $m_C^{(l)} = \mathrm{mean}_{h,q} \sum_{j \in C} A_{q,j}^{(l,h)}$. We *ascend*

$$\mathcal{L}_{\mathrm{DC}}(x') = 1 - \frac{1}{|\mathcal{S}|} \sum_{l \in \mathcal{S}} m_C^{(l)}, \tag{11}$$

i.e. we minimize the mean context attention proportion. *FLUX:* the sequence is $512 + 1024 + 1024 = 2560$ tokens (T5 length fixed at 512). *FIBO:* the same objective is used, but the text length is read from the current prompt instead of being fixed, and the additive attention mask M (masking padded text tokens) is added to the scores before the softmax so that the softmax matches the model's own.

B.4 Diff-PGD (edit-reconstruction loss)

Diff-PGD, proposed by (Shen et al., 2025), as a sensible baseline maximizes the flow-matching error of the model when it tries to produce a clean edit from the protected context. For each (x, p) we first generate a clean edited target $y_p = G(x, p)$ with the editing model itself, always from the original x, and encode it, $z_p^\star = E(y_p)$. Targets are generated lazily the first time a prompt is sampled and cached on disk and in memory, so each prompt costs one generation per image. At each step we sample p, t and noise n, form $z_t = (1 - \sigma_t) z_p^\star + \sigma_t n$, and *ascend*

$$\mathcal{L}_{\mathrm{DP}}(x') = \Big\| \big[F_\theta\big([\, z_t \mid \mathrm{tokens}(E(x'))\,],\, p,\, t \big) \big]_Z - (n - z_p^\star) \Big\|_2^2, \tag{12}$$

where $[\cdot]_Z$ keeps the outputs at the target-stream tokens and $n - z_p^\star$ is the flow-matching velocity target. *FLUX:* G is FLUX Kontext run for 28 steps with guidance scale 3.5, using the precomputed prompt embeddings and a separate scheduler instance so the attack's scheduler is unaffected. *FIBO:* G is the FIBO Edit pipeline run for 30 steps with classifier-free guidance scale 5.0 The JSON prompt (VLM-generated by default) is used both to generate the target and in the loss, and VLM-JSON targets are cached separately from those made with the minimal prompt.

B.5 Ours (context-attention maximization)

Our method also uses the attention of the target stream but acts on the double-stream (joint) blocks and drives the target queries' attention mass *onto* the reference (context) tokens. No prompt pool is needed: the pool contains a single empty prompt, so the attack is unconditional. For each selected double block $l \in \mathcal{D}$ we hook the normalized queries and keys of both streams (image stream $[Z \mid C]$ and text stream T), rebuild the joint sequence $[T \mid Z \mid C]$, apply the joint rotary embedding, and compute the ordinary softmax attention of the target queries. With $m_C^{(l)}$ as above (here called the reference mass, computed *per query* before averaging), we *ascend*

$$\mathcal{J}(x') = -\frac{1}{|\mathcal{D}|} \sum_{l \in \mathcal{D}} \mathrm{mean}_{h,q} \left(1 - \textstyle\sum_{j \in C} A_{q,j}^{(l,h)}\right)^2, \tag{13}$$

which pushes the reference mass of every head and target query towards 1. *FLUX:* $\mathcal{D}$ is double blocks 0–9, and the empty prompt is encoded by CLIP and T5 (512 tokens) with guidance 3.5. *FIBO:* $\mathcal{D}$ is double blocks 0–3 (half of the eight FIBO Edit's double blocks); the additive attention mask is added to the scores as in the model's own attention; and the empty instruction is turned into a single VLM-generated JSON per image, so the VLM is called once per image.