YOUR LATENT MASK IS WRONG: DECODER-EQUIVALENT COMPOSITING FOR DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Linearly interpolating between VAE latents using a downsampled mask field remains a common heuristic for diffusion inpainting. However, this approach systematically violates a key principle: latent compositing must respect decoder equivalence; decoding after compositing should approximate compositing after decoding. Because VAE latents capture global context rather than pixel-local structure, linear interpolation fails this requirement, producing seams, color shifts, and halos that diffusion subsequently amplifies into larger artifacts.

We propose Decoder-Equivalent Latent Compositing (DELC) and instantiate it with DecFormer, a 14M-parameter transformer that predicts per-channel blend weights and a nonlinear residual to realize mask-consistent latent fusion. DecFormer is trained so that decoding after fusion matches pixel-space alpha compositing, is plug-compatible with existing diffusion pipelines, and requires no backbone finetuning. It adds only 0.07% of FLUX.1-Dev's parameters and 9.26% of the VAE.

On the FLUX.1 family, DecFormer restores global color consistency, sharp boundaries, and high-fidelity masks, reducing halo metrics by up to 53% over broadcast-mask interpolation. Used as an inpainting prior, a lightweight LoRA on FLUX.1-Dev with DecFormer achieves fidelity comparable to FLUX.1-Fill, a fully finetuned inpainting model. While we demonstrate inpainting, DELC is a general recipe for decoder-equivalent latent editing (e.g., overlays, tone/relighting, warps).

1 INTRODUCTION

Latent diffusion models (LDMs) (Rombach et al., 2022; Peebles & Xie, 2022) dominate modern image generation, yet localized editing remains brittle. In practice, inpainting is commonly implemented by interpolating VAE latents under a downsampled mask. The heuristic is simple, but VAE decoders are nonlinear and spatially entangled, so mixing latents does not mix pixels. The result is off-manifold seams, color shifts, and halos that diffusion then amplifies across denoising steps.

We propose a simple principle: latent compositing should be *decoder-equivalent* (DE). For a frozen decoder D and any pixel-space operator F, a latent operator C_F should satisfy

$$D(C_F(z)) \;\approx\; F(D(z))\,.$$

That is, applying F after decoding should match applying C_F before decoding. This covers masking, overlays, tone/relighting, and warps/flow. As a concrete case, inpainting uses

$$F(x_A, x_B, m) = (1-m)\odot x_A + m\odot x_B, \quad \text{so} \quad D(C_F(z_A, z_B, m)) \approx (1-m)\odot D(z_A) + m\odot D(z_B).$$

Linear latent blending would satisfy decoder-consistency only if D were locally linear and channel-separable, assumptions that we show provably fail in modern VAEs (see Prop. 1).

Let $z = (1-\alpha)z_A + \alpha z_B$ and $\Delta z = z_B - z_A$. A first-order expansion at z_A gives $D(z) \approx D(z_A) + J_D(z_A)\,\alpha\Delta z$, while the pixel target is $D(z_A) + m\odot\Delta x$ with $\Delta x = D(z_B) - D(z_A)$. Matching these terms for generic (z_A, z_B, m) would require $J_D(z_A)$ to be diagonal, spatially local, and constant so that $J_D(z_A)\,\alpha\Delta z = m \odot \Delta x$. Modern VAEs violate these conditions (nonlinearity, cross-channel mixing, wide ERFs), hence linear latent blending is not decoder-equivalent. See Prop. 1 in §2.

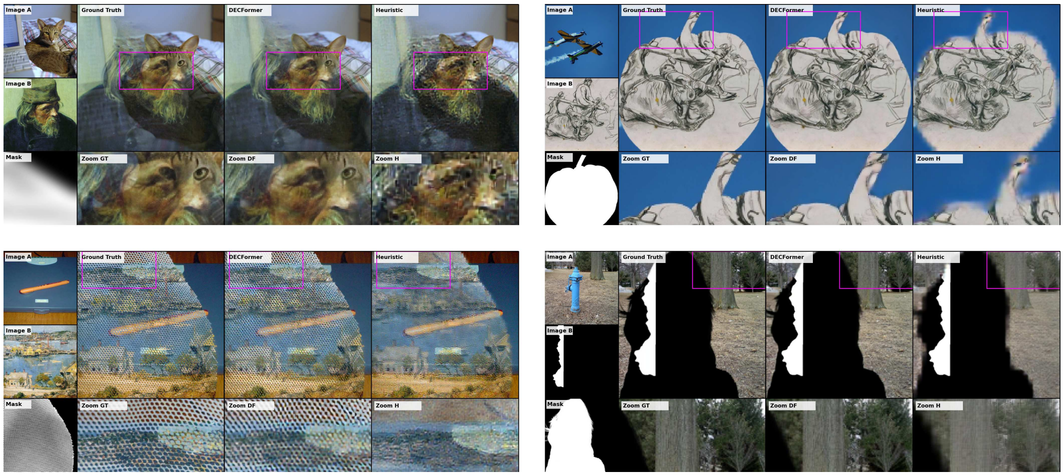


Figure 1: Each quadrant compares ground-truth pixel composites, our DECformer predictions, and heuristic latent interpolation (naïve mask broadcast). Across soft, binary, and structured masks, DECformer restores sharp edges and high-frequency detail, whereas the heuristic exhibits smearing on soft blends, halos and discoloration at boundaries, and blocky low-fidelity masks. Notably, in the bottom-right example, global background degradation occurs far from the masked region, reflecting how latent entanglement corrupts off-mask content; this effect is eliminated by DECformer.

Modern VAE latents couple wide spatial context and heterogeneous channels; broadcasting a single, downsampled mask and linearly mixing latents introduces boundary leakage and global color drift. Diffusion magnifies these inconsistencies, degrading both local fidelity and global coherence. Figure 1 shows the effect: heuristic blending yields visible halos and boundary mismatch, while a decoder-equivalent compositor restores sharp edges and global color.

We introduce **DELC** (Decoder-Equivalent Latent Compositing), a model-agnostic methodology for learning latent operators C_θ that are decode-equivalent with target pixel operators using only a frozen encoder–decoder and synthetic supervision from pixel composites. We interpret ”compositing” broadly to encompass not only traditional multi-image operations like inpainting and overlays, but also the composition of a single latent with a parametric adjustment, such as the color corrections demonstrated in this work. Importantly, even with anti-aliased mask pooling and per-channel masks, decoder-equivalence fails in practice due to nonlinearity and long-range coupling in modern VAEs. As an inpainting instantiation, we propose **DECformer**, a lightweight 14M-parameter transformer that predicts per-channel blend weights together with a nonlinear residual correction, supporting *genuinely soft masks* (α values outside $[0, 1]$ plus residual s) rather than merely feathered 0–1 mixing. This adds $<0.1\%$ overhead to a 12B diffusion backbone.

Contributions.

- **Principle & recipe.** We formalize decoder-consistency as a general criterion for latent compositing and present DELC, a simple training recipe to realize it from pixel-space supervision.
- **Theory.** We prove (Prop. 1) that linear latent interpolation cannot meet decoder-consistency except under unrealistic decoder assumptions.
- **Instantiation.** We design DECformer, a 14M-parameter compositor that restores mask fidelity and supports genuinely soft masks with negligible overhead.
- **Empirics & scope.** On FLUX-family VAEs, DELC reduces halo metrics by up to 60% and, with a lightweight LoRA, matches the fidelity of a dedicated inpainting model (FLUX-Fill). Our method repairs latent fusion but does not replace fully mask-aware backbones when joint prompt–mask reasoning is required.
- **Generalization.** Although we demonstrate inpainting, DELC applies to any pixel-space operator F (screen, overlay, relighting, warps/flow), providing a path to principled latent-space editing without re-decoding at every step.

2 Background and Problem Analysis

2.1 LDMs and VAE latents

Instead of denoising in pixel space, which is prohibitively expensive, modern diffusion models operate in the latent space a pretrained variational autoencoder (VAE). Given an image $x \in \mathbb{R}^{H \times W \times 3}$, the VAE encoder E produces a latent tensor

$$z = E(x) \in \mathbb{R}^{h \times w \times C}, \quad h, w \ll H, W,$$

on which diffusion unfolds over several steps. The decoder D then reconstructs pixels, reducing training and sampling cost by orders of magnitude while retaining perceptual fidelity.

A notable feature of these latents is that they resemble images. Channel-wise visualizations, show downsampled content aligned to the spatial (h, w) grid. This is not by design but an artifact of convolutional encoders: each latent voxel $z[i, j, :]$ is aggregated by strided convolutions over a receptive field centered at approximately

$$\Big(\lfloor i \cdot H/h \rfloor,\ \lfloor j \cdot W/w \rfloor\Big).$$

Because convolutions are translation-equivariant, neighboring voxels in (h, w) space correspond roughly to neighboring regions in pixel space, and so the (h, w)-grid of latents can be mistaken for a *miniature image*.

2.2 Heuristic latent masking

This image-like structure motivated the heuristic of applying masks directly in the latent space for inpainting, which is now common practice in commercial use, academic literature and mainstream pipelines. Concretely a pixel mask, often binary but common with softened edges [1], $m \in \{0, 1\}^{H \times W \times 1}$, is interpolated to match the latent resolution of z with a fixed resizer $\mathcal{S} : \{0, 1\}^{H \times W} \to [0, 1]^{h \times w}$ and the result latent mask is broadcast to match the channel dimensions of z.

$$\alpha \ = \ \mathcal{S}(m) \in [0, 1]^{h \times w}, \qquad \tilde{\alpha}[i, j, c] \ = \ \alpha[i, j] \quad \forall c \in \{1, \ldots, C\}.$$

At each denoising step, the latent update $\hat{z}_{t \to t-1}$ is convex blended with the original latent z_{t-1}^{orig} using latent mask α.

$$z_{t-1} \ = \ \tilde{\alpha} \odot \hat{z}_{t \to t-1} \ + \ (1 - \tilde{\alpha}) \odot z_{t-1}^{\text{orig}}.$$

Throughout our paper we refer to this approach as heuristic blending This approach of blending latents like images worked *well enough* in early VAEs such as Sta/ble Diffusion (SD) 1.x/2.x/XL, whose modest 4-channel latents were relatively image-like and whose decoder receptive fields were relatively narrow. But the justification is mathematically unsound. The autoencoder does *not* satisfy

$$E(x_A \oplus_m x_B) \ = \ m \cdot E(x_A) + (1 - m) \cdot E(x_B),$$

so convex mixing in latent space is not guaranteed to correspond to masked mixing in pixel space.

Proposition 1 (Impossibility of decoder-equivalence by linear blending)**.** *For a nonlinear VAE decoder D, there exist z_A, z_B and masks m such that*

$$D((1 - \alpha) z_A + \alpha z_B) \neq (1 - m) \odot D(z_A) + m \odot D(z_B)$$

for all $\alpha \in [0, 1]^{C \times H \times W}$. That is, no convex latent interpolation is decoder-equivalent in general.

2.3 Heuristic masking pathology

With the advent of diffusion transformers (DiTs) and modern VAEs, the flaws of this heuristic have become amplified:

[1] Diffusers inpainting guide and pipeline expliclity include mask processors that blur edges

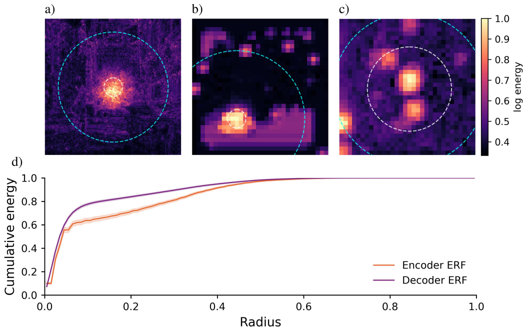


Figure 2: **Effective Receptive Field (ERF) analysis** of the Flux VAE on 256×256 images; panels (a–c) are randomly chosen near the pooled median. Statistics pool 128×3 probes with 95% bootstrap CIs; radii r_{50}, r_{90} are cumulative-energy radii (fraction of image/latent diagonal). **(a) Decoder FD ERF:** perturb one latent site by ε in all channels and plot $\|D(z + \varepsilon e) - D(z)\|_2$ per pixel (log). Visualization uses adaptive ε ($\sim$0.5% decoded RMS; clamp $[2\times10^{-3}, 10^{-2}]$); *metrics* use fixed ε=10^{-3}. **Radii:** $r_{50} \approx 0.044 \pm 0.024$**,** $r_{90} \approx 0.291 \pm 0.099$. The bright core plus global low-amplitude 'blanket' highlights decoder non-locality, seems to echo high contrast structures in source image. **(b) Encoder FD ERF:** inject a 1px impulse (δ=0.05) at pixel (i, j), compute Δz=$E(x + \delta e_{i,j})$−$E(x)$, and plot $\|\Delta z_{\cdot,u,v}\|_2$ per latent site (log). **Radii:** $r_{50} \approx 0.091 \pm 0.082$**,** $r_{90} \approx 0.356 \pm 0.099$. **(c) Gradient ERF:** for y=$D(z)$ and s=$\sum_{5\times5} y^2$, backpropagate to $\partial s/\partial z$ and show channelwise ℓ_2 per latent site. Central peak and multiple secondary latent clusters relied on by the same pixel patch, exposing repeated structure. **(d) Energy curves:** Shaded region shows 95% bootstrap confidence; both Encoder and Decoder shows a sharp core with long, low-amplitude tails, showing large ERFs; evidence that heuristic latent masking is inconsistent with the VAE and motivates DecFormer.

1. **High-dimensional embeddings:** Modern encoders that support transformers inflate the channel dimension ($C = 16$ in SD3 and Flux versus $C = 4$ in SD 1.x/2.x/XL) which stabilizes the projection into the high-dimensional token space. These channel are not ”mini-pixels” but heterogenous non-linear features. At a fixed latent spatial coordinate (i, j) channel $c = 1$ may encode edge structure aligned with that pixel region, while channel $c = 12$ may encode a texture statistic or semantic concept aggregated over a much wider or even global field. Broadcasting a scalar mask across channels is geometrically unsound and the error of which is amplified by the increased channel dimensionality.

2. **Boundary leakage:** The VAE decoder has a large receptive field, typically global due to attention presense, so seams in latent space cannot be localized. Mask boundaries bleed across hundreds of pixels in the reconstructed image, producing visible halos. This effect is exacerbated by soft masks: intermediate values between 0 and 1 create “partially replaced” voxels that are not valid encodings of either source image, and their errors diffuse widely through the receptive field.

3. **Invalid interpolation:** Empirically, more than half of voxels require mixing coefficients α outside $[0, 1]$ to reproduce the ground-truth encoding. Restricting α to convex interpolation therefore forces systematic error. Even binary masking ($\alpha \in \{0, 1\}$) is only marginally consistent; continuous soft masking produces especially implausible states, as no point on the manifold corresponds to a fractional blend of two encoded latents. Accurate inpainting thus requires both extrapolation and explicit residual corrections beyond convex mixing.

4. **Mask downsampling:** Resizing the pixel mask to latent resolution discards fine structure. Thin strokes or line segments may vanish entirely, while curved boundaries become blocky when aligned to the coarse (h, w) grid. As a result, regions are incorrectly included or excluded, reducing mask fidelity and often leaking information to the diffusion model that prevents successful inpainting (for example removal of object, changing of color). In effect, the model applies a low-resolution mask to a high-resolution image, introducing geometric errors even before any latent blending occurs.

Despite these limitations, heuristic latent masking remains the default in many pipelines. In diffusion transformers the flaws are especially severe: masked regions are often interpreted as entirely separate scenes, borders mismatch in color and texture, and edits either leak beyond the mask or fail altogether. Coarse downsampling of pixel masks forces aggressive dilation to prevent disappearance, but this sacrifices fine structure and makes small or precise edits infeasible. While mask-aware conditioned models have existed since the SD 1.x era—primarily addressing consistency—they were treated as optional alternatives. In contrast, for modern DiTs the pathology is so severe that mask-aware networks such as FLUX-Fill are essential for any masking task. Yet these approaches require retraining a second full-scale diffusion model, and frequently come with a loss in quality, an impractical cost for most deployments.

2.4 Related Work

Guided diffusion editing usually enlarges or retrains the denoiser. ControlNet adapters (Zhang et al., 2023), Paint-by-Example (Yang et al., 2023), DiffEditor (Mou et al., 2024), BrushNet (Ju et al., 2024), and PowerPaint (Zhuang et al., 2024) each introduce hundreds of millions of new weights and require multi-GPU training just to teach the backbone mask awareness. In practice, foundation deployments still ship separate inpainting checkpoints—Stable Diffusion maintains SD-inpaint and SDXL-inpaint variants (RunwayML, 2022; sdx, 2023), while FLUX Fill is a second 12B rectified-flow model (Black Forest Labs, 2024). The added maintenance cost has spurred community forks that graft extra control nets on top in search of cleaner seams (Alimama Creative Team, 2024a;b; rep, 2024; dif, 2024). LatentPaint (Corneanu et al., 2024) trims the training burden but still hands seam quality to the denoiser via broadcast latent masks.

Trajectory-level methods edit sampling paths instead: SDEdit (Meng et al., 2022), DiffEdit (Couairon et al., 2023), Blended Diffusion (Avrahami et al., 2022), and Differential Diffusion (Levin & Fried, 2023) modulate the noise schedule or splice reference latents during denoising. These strategies improve controllability yet still stitch latents with a single-channel mask at each step, so VAE boundary coupling produces halos. Independent analyses of autoencoder interpolations (Oring et al., 2021) show that convex latent blends leave the data manifold, underscoring the need for an explicit compositor. Our contribution targets this missing piece: a lightweight latent operator that enforces decoder-equivalent blending, offloading mask geometry from giant finetunes while leaving semantic inpainting to those backbones.

3 Methodology

3.1 Formulation

Heurisic latent blending by broadcasting a downsampled pixel mask produces systematic errors: artifacts cluster at mask boundaries and leakage arises from globally entangled VAE channels (Appendix 6). Despite this, interpolation remains a strong inductive prior especially away from mask edges; we therefore instantiate DELC for inpainting as channel-adaptive—a strictly stronger formulation than broadcast masking—blending along the $z_A \leftrightarrow z_B$ line, augmented with an explicit residual correction.

Formulation. We predict

$$\hat{z} = (1-\alpha)z_A + \alpha z_B + s, \qquad \alpha \in [0,1]^{C\times H\times W}, \;\; s \in \mathbb{R}^{C\times H\times W}.$$

Constraining α to $[0,1]$ yields a stable blend axis; s absorbs orthogonal leakage and curvature. This separation is critical: if α is unconstrained it collapses both roles, destabilizing training. Appendix A gives a closed-form decomposition into (α^*, s^*), and ablations are shown in Table 1.

Why a transformer. Latent channels are globally entangled: the reconstruction of a masked pixel depends on structure far outside the local region. Convolutions alone cannot capture this; a small transformer with global attention is the minimal mechanism to propagate long-range cues. Since our compositor runs at *every* diffusion step, we must balance expressivity against strict latency and parameter budgets.

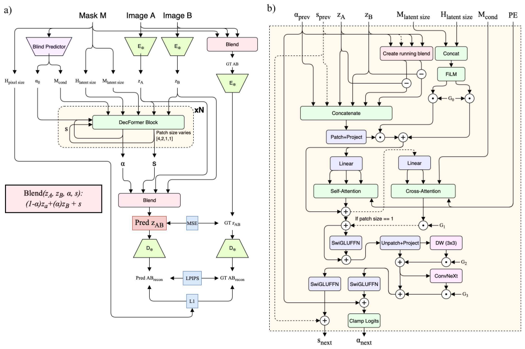


Figure 3: Overview of our training pipeline and DecFormer architecture. The left panel illustrates the overall flow: two input images and a pixel mask are encoded by a frozen VAE, the mask is processed by a lightweight CNN prior (architecture detailed in Appendix B), and DecFormer predicts channel-adaptive blend weights α and residual corrections s at latent resolution. The right panel zooms into a single DecFormer block, showing the feature stack, patching/unpatching, FiLM conditioning, attention and cross attention. For an expanded diagram on the DecFormer architecture see Appendix C.

Architecture. Each block receives a *rich feature stack*: latents (z_A, z_B), the running (α, s), and error cues $\|z^{(t)} - z_A\|, \|z^{(t)} - z_B\|$. Re-patching/unpatching permits injection of these per-voxel errors at each stage. Multi-scale blocks ($[4, 2, 1, 1]$ patch sizes) let coarse stages cheaply gather context, while patch $= 1$ refines pixel-level boundaries. Local convolutions after unpatching also suppress 1–2px halos. This design is both FLOP-efficient and effective at feeding both global and local signals. The full model has 27M parameters, orders of magnitude smaller than the 12B parameter diffusion model it augments.

3.2 Scheduler Modification

DecFormer is trained on clean latents, however naïve masking in diffusion pipelines operates directly on partially noised latents (x_t), pushing DecFormer far out-of-distribution (OOD) and causing failure. We consider two options: (i) train DecFormer on noised inputs, or (ii) reformulate the scheduler step to blend at the fully denoised x_0 then re-noise to x_{t-1}. We adopt (ii), as it preserves a clean supervision landscape, and does not degrade our primary pixel space losses with OOD inputs to the decoder.

In a velocity-form sampler such as flux, the network predicts

$$v_\theta(x_\sigma, \sigma), \qquad x_{\sigma'} \;\leftarrow\; x_\sigma + (\sigma' - \sigma)\, v_\theta(x_\sigma, \sigma).$$

Instead, we **(A)** decode to x_0, **(B)** blend there, then **(C)** re-target the velocity to land on the blended $x_0^\star$ and step once:

$$\begin{aligned}
\textbf{A:}\quad x_0^\theta \;&=\; x_\sigma - \sigma\, v_\theta(x_\sigma, \sigma)\\
\textbf{B:}\quad x_0^\star \;&=\; k \odot x_0^{\text{ref}} + (1-k) \odot x_0^\theta \quad (\text{or } x_0^\star = \mathcal{C}_\phi(x_0^\theta, x_0^{\text{ref}}, m))\\
\textbf{C:}\quad v^\star \;&=\; \frac{x_\sigma - x_0^\star}{\sigma}, \qquad x_{\sigma'} \;=\; x_\sigma + (\sigma' - \sigma)\, v^\star.
\end{aligned}$$

Here k is a keep-mask where 1 indicates regions to keep, and 0 to edit. At $\sigma \to 0$ we set $x_{\sigma'} = x_0^\star$.

Beyond our use case, this re-targeting appears to yield cleaner boundaries with naïve masks, indicating it is a sensible default for diffusion masking.

Figure 4: **Validation of DELC on a non-compositing task.** This experiment highlights the fragility of heurisic latent manipulation. **(Top)** Our DELC-trained model quantitatively bridges the massive error gap of the baseline. **(Bottom)** A qualitative example (Appendix H) shows how the naive method fails catastrophically with severe artifacts when faced with more aggressive color correction, while our model's output is nearly indistinguishable from the ground truth. This illustrates that VAE latents are not pixel-like and require a principled framework like DELC.

Metric	Baseline (heuristic)	DELC-trained model (Ours)
LPIPS ↓	0.4996 ± 0.0076	**0.0875** ± 0.0023
PSNR ↑	18.1630 ± 0.1968	**27.2835** ± 0.2301
SSIM ↑	0.4359 ± 0.0112	**0.8466** ± 0.0059

(a) Quantitative Metrics (Mean ± 95% CI) on 1024 randomly samples images from the COCO-2017 validation dataset

3.2.1 Model Design

We decompose the prediction as $\hat{z} = L(\alpha) + s$ with $L(\alpha) = (1-\alpha)z_B + \alpha z_A$ and $d = z_A - z_B$. The term $L(\alpha)$ models variation along the blend line $\operatorname{span}\{d\}$, while s captures residual structure off that line. We regularize s using a scale-aware sparsity prior

$$\mathcal{L}_{\text{shift}} = \lambda\, \mathbb{E}\big[w(d)\,|s|\big], \qquad w(d) = \tfrac{\|d\|}{\|d\|+\kappa},$$

which encourages the model to explain as much as possible along d (via α) when $\|d\|$ is large, while not penalizing s in regions where α is intrinsically uninformative ($\|d\|$ small). Note $\mathcal{L}_{\text{shift}}$ is isotropic in s and thus does not bias its direction toward d.

4 Experiments

Before evaluating our primary application of decoder-equivalent inpainting, we first conduct a controlled experiment to validate the core principle of DELC on a fundamentally different task.

4.1 Proof of Concept: Generality on a Parametric Transformation

Objective To validate the generality of the DELC framework beyond multi-image tasks, we test its ability to learn a complex parametric color transformation. This operator, defined as $F(x;\gamma,c,b) = (x^{1/\gamma} - 0.5)\cdot c + 0.5 + b$, combines gamma, contrast, and brightness adjustments. The task serves as an ideal proof of concept, as the pixel-space ground truth is exact, while the required latent-space manipulation is highly nonlinear and distinct from compositing.

Setup & Results We train a lightweight, FiLM-conditioned transformer to predict a latent residual, conditioned on the transformation parameters $(\log\gamma, \log c, b)$. The model is optimized via the DELC recipe to be decoder-equivalent, using a combined objective of latent MSE and pixel-space LPIPS. As demonstrated quantitatively and qualitatively in Figure 4, our DELC-trained model successfully learns this complex mapping, reproducing the target transformation with high fidelity, whereas naively operating on the latent causes the image to degrade catastrophically. This success validates that DELC is a general framework for learning decoder-equivalent latent operators, including single-image parametric adjustments. We report results n=1024 on the COCO-2017 validation set.

4.2 Network Design Ablations

We tested targeted ablations to isolate which components materially contribute to decoder-consistent compositing. Table 1 reports dataset-level means ±95% CIs over three seeds up to 80,000 steps.

Table 1: Ablation results. Removing halo-focused losses degrades boundary quality (↑Halo L1) even though global metrics are competitive. Removing the residual shift head (unconstrained α-only) substantially harms all metrics, confirming the need for both and residual s. Baseline balances both.

Experiment	Halo L1 (↓)	LPIPS (↓)	MSE (↓)
No Halo L1 Loss	0.0973 ± 0.0002	0.0299 ± 0.0003	0.0297 ± 0.0003
Baseline	0.0829 ± 0.0018	0.0303 ± 0.0015	0.0303 ± 0.0003
Unconstrained Alpha, No Shift	0.1079 ± 0.0012	0.0514 ± 0.0012	0.0331 ± 0.0003

Table 2: Comparison of DecFormer and the heuristic bilinear baseline at 1024px resolution (mean ± 95% CI, n=50).

Mask Type	**Method**	**SSIM ↑**	**PSNR (dB) ↑**	**LPIPS ↓**	**Halo L1 ↓**
Soft ($sigma$=21)	DecFormer	**0.985**$_{\pm.003}$	**41.3**$_{\pm.8}$	**0.027**$_{\pm.005}$	**0.018**$_{\pm.001}$
	Heuristic Bilinear	0.941$_{\pm.010}$	32.9$_{\pm1.1}$	0.088$_{\pm.016}$	0.050$_{\pm.005}$
Binary	DecFormer	**0.964**$_{\pm.017}$	**35.7**$_{\pm1.5}$	**0.045**$_{\pm.018}$	**0.060**$_{\pm.006}$
	Heuristic Bilinear	0.913$_{\pm.025}$	28.4$_{\pm1.3}$	0.110$_{\pm.029}$	0.141$_{\pm.008}$
Original	DecFormer	**0.968**$_{\pm.016}$	**38.6**$_{\pm1.5}$	**0.049**$_{\pm.018}$	**0.037**$_{\pm.005}$
	Heuristic Bilinear	0.918$_{\pm.024}$	31.1$_{\pm1.4}$	0.104$_{\pm.028}$	0.080$_{\pm.007}$
Thin	DecFormer	**0.967**$_{\pm.014}$	**34.7**$_{\pm1.5}$	**0.045**$_{\pm.017}$	**0.073**$_{\pm.005}$
	Heuristic Bilinear	0.920$_{\pm.030}$	27.3$_{\pm1.2}$	0.111$_{\pm.031}$	0.174$_{\pm.009}$

Ablation strategy. Because our claim concerns decoder-equivalence rather than architecture per se, we prioritized ablations that test the principles that make equivalence attainable. We therefore fully trained (80k steps, three seeds) a minimal set. Short screen-tests (30k) were used internally to cull ideas that did not materially improve performance or equivalence early. We report dataset-level means with 95% CIs for LPIPS, L1, and halo-band L1, plus an edge-contrast term focused on boundary sharpness. (Implementation details: FiLM conditioning on anti-aliased, dilated mask pools; optional latent-native halo conditioning; /shift supervision via (α^*, s^*) decomposition.)

4.3 DecFormer Achieves Decoder-Equivalent Compositing

We first evaluate DecFormer on the primary trained task of decoder-equivalent compositing, comparing its reconstruction fidelity against the standard heuristic baseline. We evaluate using random masks from the Compositions-1k dataset and randomly selected images from the Coco 2017 validation set. We use set seeds and dataset hashing to ensure the same experimental set is used each time.

Our results show a decisive improvement across all metrics and mask types at all resolutions 2 (extra resolutions, and comparisons to different naive downscaling methods are shown in Appendix D). DecFormer consistently outperforms the heuristic baseline by a significant margin. For both challenging thin and binary masks, DecFormer reduces the Halo L1 error. Perceptual error as measured by LPIPS is consistently more than halved, confirming that heuristic blending introduces significant, measurable artifacts that DecFormer effectively eliminates.

To understand where these improvements originate, we analyse the reconstruction error as a function of the signed distance from the mask boundary, shown in Figure 6. We evaluate reconstruction error at 10k images and masks, summing by signed distance. The heuristic baseline (grey) exhibits a prominent error spike at the boundary (distance = 0) that decays slowly into both the masked and unmasked regions. This confirms the visual phenomena of ”leaking” and halos. In contrast, DecFormer (green) not only achieves a substantially lower peak error but also demonstrates a much sharper error fall-off.

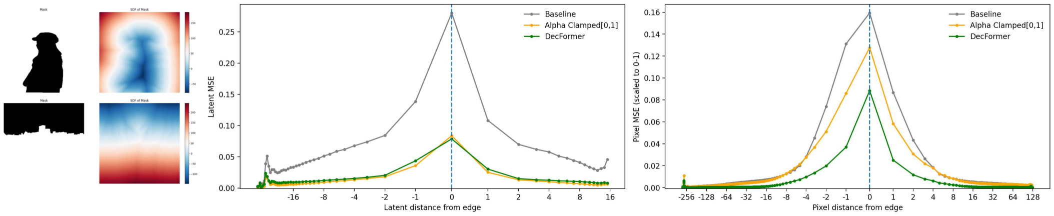


Figure 5: Signed-distance analysis of mask edges. Left: example masks and their signed distance fields (SDF). Middle: latent-space MSE as a function of signed distance to the edge (dashed line at 0; negative = inside the masked region). Right: pixel-space MSE versus pixel distance to the edge. Heuristic (grey) applies the downsampled binary mask in latent space; Alpha-clamped (orange) clamps the unconstrained alpha to [0,1]; DecFormer (green) learns compositing. DecFormer achieves the lowest error around the boundary with a sharper fall-off compared to baselines.

4.4 DecFormer is a strong Diffusion Inpainting Prior

We now evaluate DELC as an inpainting prior for a diffusion backbone, both frozen and tuned with a small LoRA. Our goal is to test whether repairing the masking operation during the diffusion denoising process yields end-to-end improvements for a masked generation task. We integrate DecFormer into Flux.1-Dev using the x_0 re-targeted scheduler from §**??**. At each sampling step (for 30 steps), we update the predicted velocity $v^{\star}$ using mask m and reference x_0^{ref}. For training a lightweight inpainting prior on the backbone, we use COCO-2017 *train* images with instance segmentations as masks. For evaluation, we report metrics on the *val* split, filtering examples with masked area $> 15\%$ of the image; prompts and guidance are held fixed across methods.

We compare five variants:

1. **Heuristic**: downsampled, broadcast latent mask blended at every step (§**??**).
2. **DecFormer (Blend)**: our compositor inserted at every step via x_0 re-targeting; backbone frozen.
3. **LoRA only**: LoRA on `Flux.1-Dev` trained for inpainting using the same masks; no compositor.
4. **DecFormer + LoRA (Blend+LoRA)**: DecFormer plus the inpainting LoRA.
5. **Flux.1-Fill**: a fully finetuned, mask-aware inpainting model used as a strong reference.

Results. We report SSIM, PSNR, LPIPS, and FID against ground-truth composites constructed from COCO segmentations. DecFormer, without any diffusion finetuning, improves end-to-end inpainting over the latent-heuristic baseline across all metrics (see table 3 and qualitative examples in Fig. 6). Adding a small LoRA (*Blend+LoRA*) further narrows the gap to a dedicated inpainting model: LPIPS and FID match or slightly outperform `Flux.1-Fill`, while PSNR remains slightly lower, reflecting that `Flux.1-Fill` better optimizes pixelwise fidelity inside the edited region. DecFormer yields a sizeable improvement, and *Blend+LoRA* achieves the best LPIPS/FID in the group while approaching `Flux.1-Fill` in SSIM/PSNR 3. Figure 6 illustrates common failure modes with the heuristic. Replacing the latent bilinear with DecFormer reduces boundary artifacts, and the edits are more aware of the masked area. LoRA primarily helps with semantic plausibility inside the masked region (e.g., object completions), whereas DecFormer governs how content is stitched, yielding complementary gains.

5 Conclusion

We revisited a widespread but fragile practice in diffusion inpainting: linear blending of VAE latents with a downsampled mask. Our analysis and measurements show that this heuristic violates decoder-equivalence, as modern VAEs are nonlinear and spatially entangled with large effective receptive fields. We formalized decoder-equivalent (DE) compositing as a criterion for latent operators and introduced DELC, a simple recipe to learn such operators from pixel-space supervision with a frozen autoencoder.

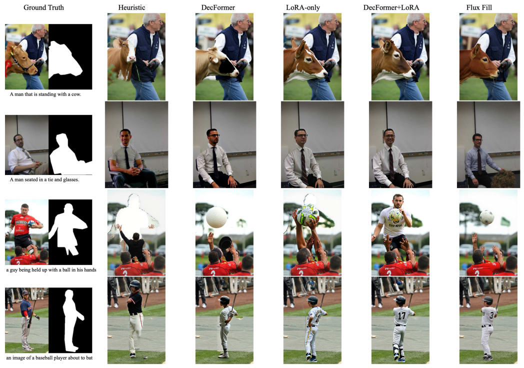

Ground Truth
Heuristic
DecFormer
LoRA-only
DecFormer+LoRA
Flux Fill
A man that is standing with a cow.
A man seated in a tie and glasses.
a guy being held up with a ball in his hands
an image of a baseball player about to bat

Figure 6: Inpainting/Editing quality comparisons for Flux.

Method	SSIM (mean±std)	PSNR (mean±std)	LPIPS (mean±std)	FID
Baseline (Heuristic)	0.643 ± 0.145	13.578 ± 2.915	0.354 ± 0.152	23.514
Blend	**0.682 ± 0.139**	13.943 ± 2.870	0.314 ± 0.143	20.556
LoRA with no Blend	0.653 ± 0.142	14.160 ± 2.620	0.331 ± 0.143	21.519
Flux Fill	0.681 ± 0.141	**16.750 ± 3.199**	0.313 ± 0.125	19.343
Blend with LoRA	0.680 ± 0.139	14.231 ± 2.742	**0.303 ± 0.138**	**19.280**

Table 3: Masked Editing results. Evaluated on all samples with masked area > 0.15 in COCO-2017 validation set.

As a concrete instantiation, we presented DecFormer, a lightweight (14M-parameter) transformer that predicts per-channel blend weights and a residual correction, yielding latent fusions whose decodes match pixel alpha compositing.

DecFormer substantially reduces boundary and global artifacts seen in heuristic masking, preserving a high level of detail on even thin or soft masks. It improves both perceptual and pixelwise metrics over heuristic masking (§4, Fig.6, Table2). Beyond inpainting, a controlled study on parametric color transforms demonstrated that DELC learns decoder-equivalent latent operators for non-compositing tasks as well (§4.1).

In end-to-end masked generation, DecFormer acts as a strong prior on its own and combines well with a small LoRA on the diffusion model, approaching the quality of a fully finetuned inpainting model. This approach avoids the extremely high cost of training a separate foundational backbone, and is still substantially smaller and easier to learn than a hypernetwork approach such as ControlNet. This show that principled adjustments of the denoise velocity keep the prediction decoder-equivalent during sampling.

Limitations and future work. DELC targets how masked regions are stitched, not what they should contain. For large semantic edits that require joint prompt–mask reasoning, mask-aware training is still needed. Further experiments on supporting mask reasoning and improving edit quality solely with frozen diffusion backbone. Extending DELC to additional operators in image and video VAE latent spaces are natural next steps. Exploring tighter theoretical guarantees for decoder-equivalence and better uncertainty estimates near boundaries could further stabilize editing in modern diffusion pipelines.

In summation, decoder-equivalent latent compositing is a simple, general principle that corrects a foundational flaw in current practice. It delivers cleaner seams with negligible overhead, plugs into existing pipelines without backbone changes, and opens a path to principled latent-space constraints for diffusion.

Acknowledgments

LLM Usage We used a large language model (LLM) to assist with phrasing and polish during manuscript preparation. It was not used to generate results, design experiments, or draft entire sections. All technical content, methods, and conclusions are our own.

References

Stable diffusion xl 1.0 inpainting 0.1. https://huggingface.co/diffusers/stable-diffusion-xl-1.0-inpainting-0.1, 2023. Adopts the same 5 additional input channels for inpainting.

Problem with fluxinpaintpipeline when doing a replace. Hugging Face Diffusers Issue #9486: https://github.com/huggingface/diffusers/issues/9486, 2024.

Flux-dev inpainting controlnet. Replicate: https://replicate.com/zsxkib/flux-dev-inpainting-controlnet, 2024.

Alimama Creative Team. Flux-controlnet-inpainting. GitHub: https://github.com/alimama-creative/FLUX-Controlnet-Inpainting, 2024a.

Alimama Creative Team. Flux.1-dev controlnet inpainting (alpha/beta). https://huggingface.co/alimama-creative/FLUX.1-dev-Controlnet-Inpainting-Beta, 2024b.

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18208–18218, 2022. URL https://arxiv.org/abs/2111.14818.

Black Forest Labs. Flux.1 fill: State-of-the-art inpainting and outpainting models. https://huggingface.co/black-forest-labs/FLUX.1-Fill-dev, November 2024. 12B parameter inpainting model.

Ciprian Corneanu, Raghudeep Gadde, and Aleix M. Martinez. Latentpaint: Image inpainting in latent space with diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4334–4343, January 2024.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.11427.

Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. *arXiv preprint arXiv:2403.06976*, 2024. URL https://arxiv.org/abs/2403.06976.

Eyal Levin and Ohad Fried. Differential diffusion: Giving each pixel its strength. *Computer Graphics Forum*, 42(6):e15040, 2023. URL https://arxiv.org/abs/2306.00950.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2108.01073.

Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing, 2024. URL https://arxiv.org/abs/2402.02583.

Alon Oring, Zohar Yakhini, and Yacov Hel-Or. Autoencoder image interpolation by shaping the latent space. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8281–8290. PMLR, 2021. URL https://proceedings.mlr.press/v139/oring21a.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022. URL https://arxiv.org/abs/2112.10752.

RunwayML. Stable diffusion inpainting model card (v1.5). https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-inpainting, 2022. UNet has 5 additional input channels (4 masked-image latents + 1 mask).

Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18381–18391, 2023. URL https://arxiv.org/abs/2211.13227.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3836–3847, 2023. URL https://arxiv.org/abs/2302.05543.

Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. *arXiv preprint arXiv:2312.03594*, 2024. URL https://arxiv.org/abs/2312.03594.

APPENDIX

A LEAST-SQUARES PROJECTION

Given target latent z_T and source latents z_A, z_B, we seek the optimal decomposition:

$$(\alpha^*, s^*) = \arg\min_{\alpha, s} \; \big\| z_T - \big[(1-\alpha) z_A + \alpha z_B + s\big] \big\|_2^2 \tag{1}$$

$$\text{s.t.} \quad \alpha \in [0,1]^{C \times H \times W}. \tag{2}$$

This yields the closed-form solution

$$\alpha^*_{ijc} = \Pi_{[0,1]} \left(\frac{(z_T^{ijc} - z_B^{ijc})(z_A^{ijc} - z_B^{ijc})}{\|z_A^{ijc} - z_B^{ijc}\|_2^2 + \epsilon} \right), \tag{3}$$

$$s^*_{ijc} = z_T^{ijc} - \big[(1 - \alpha^*_{ijc}) z_A^{ijc} + \alpha^*_{ijc} z_B^{ijc}\big], \tag{4}$$

where $\Pi_{[0,1]}$ denotes projection onto $[0,1]$.

This formulation interprets α^* as the projection of z_T onto the line spanned by (z_A, z_B), while s^* captures the orthogonal residual.

B BLIND PREDICTOR ARCHITECTURE

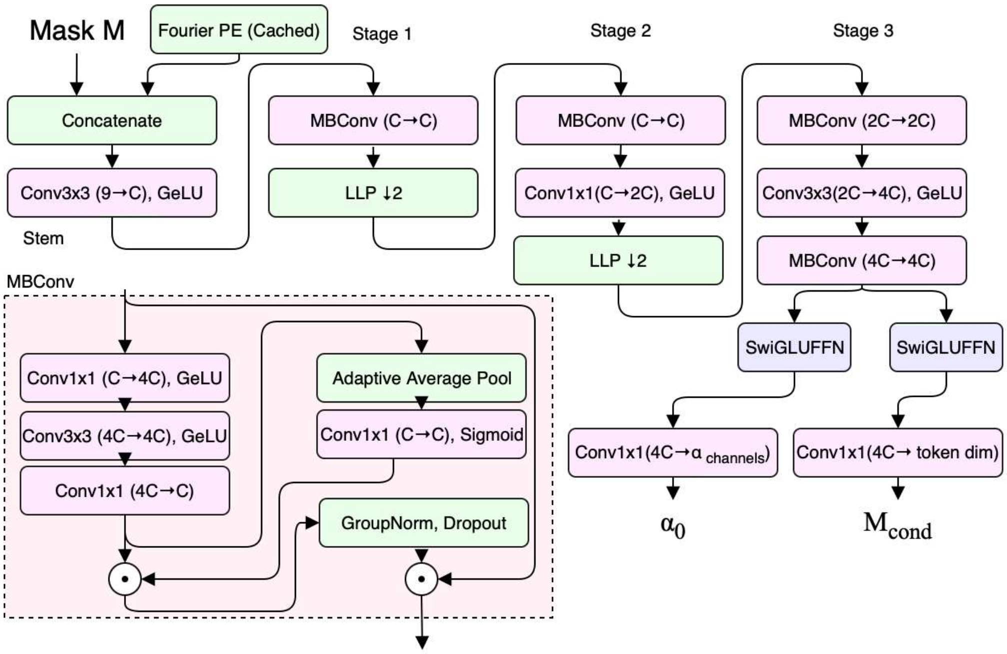


Figure 7: A lightweight CNN maps the input pixel mask (augmented with Fourier features) into latent-resolution, per-channel soft masks. The stem is a 3×3 conv with GELU, followed by three stages: Stage 1: MBConv block with depth-wise squeeze–excite, followed by a learnable low-pass filter and $2\times$ downsampling; Stage 2: MBConv $\rightarrow$ pointwise expansion $\rightarrow$ GELU $\rightarrow$ learnable low-pass, giving another $2\times$ downsample; Stage 3: MBConv $\rightarrow$ strided conv for $8\times$ total reduction $\rightarrow$ MBConv (extra receptive field). Final shared features branch into two FFNGlU heads: (i) an α head predicting per-channel blending masks with bounded activation, and (ii) a token head producing spatial embeddings for cross-attention in DeltaFormer. The diagram expands the MBConv block (pointwise–depthwise–pointwise with SE and normalization); the learnable low-pass filters are depth-wise convolutions initialized as binomial blur kernels.

C DeltaFormer Extended Architecture

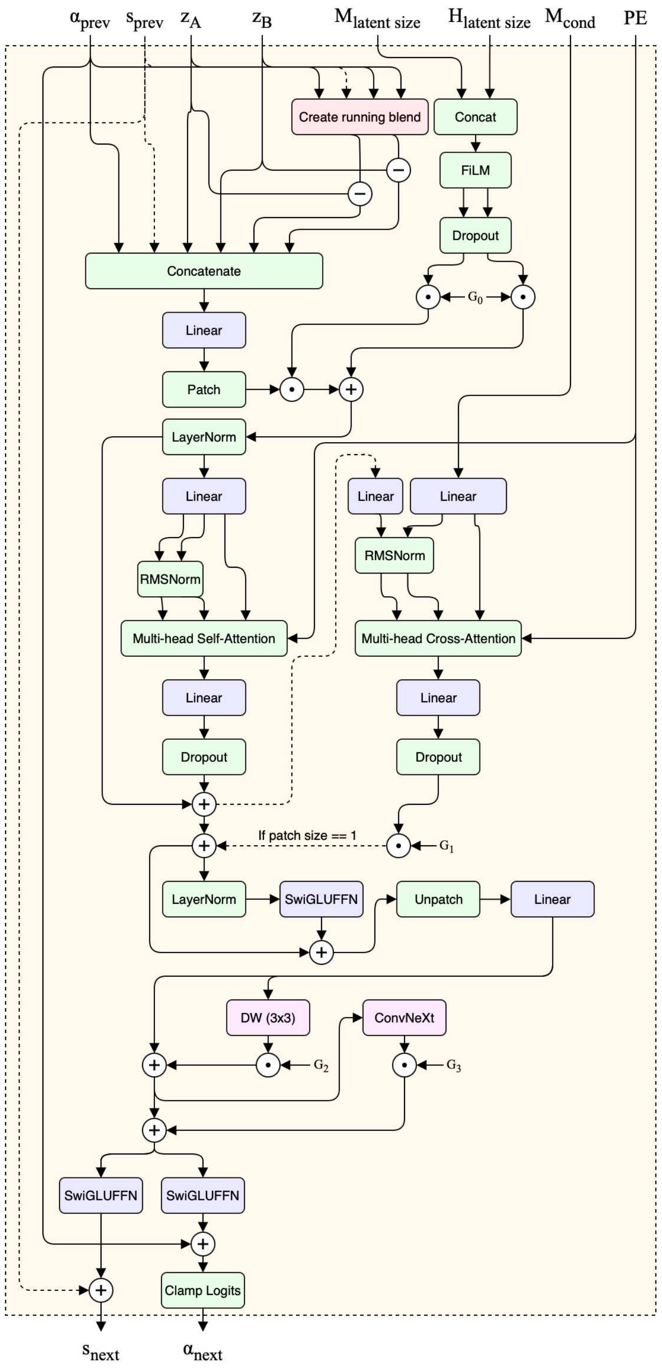


Figure 8: Extended DecFormer architecture diagram. The figure highlights more precisely the internal composition of each block, including the location and type of normalization layers, as well as the flow of intermediate projections and residual connections.

D Extended DecFormer Metrics Tables

Table 4: Complete method comparison at 1024px resolution (mean ± 95% CI, n=50). DecFormer variants compared against all heuristic baselines.

Mask Type	Method	SSIM ↑	PSNR (dB) ↑	LPIPS ↓	Halo L1 ↓
Soft ($sigma$=21)	DecFormer	0.985$_{\pm.003}$	**41.3**$_{\pm.8}$	**0.027**$_{\pm.005}$	**0.018**$_{\pm.001}$
	DecFormer-Pretrain	**0.986**$_{\pm.002}$	40.9$_{\pm.7}$	0.028$_{\pm.005}$	**0.018**$_{\pm.001}$
	DecFormer-Pretrain-NoLeak	0.980$_{\pm.002}$	38.8$_{\pm.6}$	0.042$_{\pm.006}$	0.023$_{\pm.001}$
	Heuristic Area	0.941$_{\pm.010}$	32.9$_{\pm1.1}$	0.088$_{\pm.016}$	0.050$_{\pm.005}$
	Heuristic Bilinear	0.941$_{\pm.010}$	32.9$_{\pm1.1}$	0.088$_{\pm.016}$	0.050$_{\pm.005}$
	Heuristic Nearest	0.940$_{\pm.010}$	32.4$_{\pm1.0}$	0.089$_{\pm.016}$	0.054$_{\pm.004}$
Binary	DecFormer	**0.964**$_{\pm.017}$	**35.7**$_{\pm1.5}$	**0.045**$_{\pm.018}$	**0.060**$_{\pm.006}$
	DecFormer-Pretrain	0.961$_{\pm.018}$	34.8$_{\pm1.5}$	0.058$_{\pm.022}$	0.068$_{\pm.005}$
	DecFormer-Pretrain-NoLeak	0.952$_{\pm.017}$	33.3$_{\pm1.2}$	0.058$_{\pm.013}$	0.075$_{\pm.005}$
	Heuristic Area	0.915$_{\pm.025}$	29.2$_{\pm1.2}$	0.112$_{\pm.029}$	0.135$_{\pm.007}$
	Heuristic Bilinear	0.913$_{\pm.025}$	28.4$_{\pm1.3}$	0.110$_{\pm.029}$	0.141$_{\pm.008}$
	Heuristic Nearest	0.903$_{\pm.028}$	26.3$_{\pm1.2}$	0.115$_{\pm.030}$	0.183$_{\pm.010}$
Original	DecFormer	**0.968**$_{\pm.016}$	**38.6**$_{\pm1.5}$	**0.049**$_{\pm.018}$	**0.037**$_{\pm.005}$
	DecFormer-Pretrain	0.965$_{\pm.017}$	37.9$_{\pm1.4}$	0.056$_{\pm.021}$	0.040$_{\pm.005}$
	DecFormer-Pretrain-NoLeak	0.957$_{\pm.017}$	35.8$_{\pm1.2}$	0.067$_{\pm.016}$	0.044$_{\pm.005}$
	Heuristic Area	0.919$_{\pm.024}$	31.5$_{\pm1.3}$	0.104$_{\pm.028}$	0.078$_{\pm.006}$
	Heuristic Bilinear	0.918$_{\pm.024}$	31.1$_{\pm1.4}$	0.104$_{\pm.028}$	0.080$_{\pm.007}$
	Heuristic Nearest	0.907$_{\pm.027}$	28.9$_{\pm1.2}$	0.110$_{\pm.030}$	0.110$_{\pm.009}$
Thin	DecFormer	**0.967**$_{\pm.014}$	**34.7**$_{\pm1.5}$	**0.045**$_{\pm.017}$	**0.073**$_{\pm.005}$
	DecFormer-Pretrain	0.960$_{\pm.016}$	33.4$_{\pm1.4}$	0.061$_{\pm.020}$	0.085$_{\pm.005}$
	DecFormer-Pretrain-NoLeak	0.961$_{\pm.016}$	33.0$_{\pm1.3}$	0.048$_{\pm.015}$	0.088$_{\pm.004}$
	Heuristic Area	0.922$_{\pm.029}$	28.6$_{\pm1.2}$	0.112$_{\pm.032}$	0.167$_{\pm.008}$
	Heuristic Bilinear	0.920$_{\pm.030}$	27.3$_{\pm1.2}$	0.111$_{\pm.031}$	0.174$_{\pm.009}$
	Heuristic Nearest	0.908$_{\pm.034}$	25.6$_{\pm1.3}$	0.116$_{\pm.032}$	0.207$_{\pm.011}$

Table 5: DecFormer vs. Heuristic bilinear at 512px resolution (mean ± 95% CI, n=50).

Mask Type	Method	SSIM ↑	PSNR (dB) ↑	LPIPS ↓	Halo L1 ↓
Soft ($sigma$=21)	DecFormer	**0.957**$_{\pm.008}$	**36.0**$_{\pm.9}$	**0.051**$_{\pm.009}$	**0.029**$_{\pm.003}$
	Heuristic Bilinear	0.859$_{\pm.024}$	28.6$_{\pm1.0}$	0.149$_{\pm.021}$	0.072$_{\pm.007}$
Binary	DecFormer	**0.924**$_{\pm.028}$	**31.0**$_{\pm1.5}$	**0.069**$_{\pm.022}$	**0.087**$_{\pm.011}$
	Heuristic Bilinear	0.848$_{\pm.037}$	25.2$_{\pm1.2}$	0.145$_{\pm.029}$	0.168$_{\pm.010}$
Original	DecFormer	**0.930**$_{\pm.026}$	**33.1**$_{\pm1.5}$	**0.068**$_{\pm.021}$	**0.064**$_{\pm.009}$
	Heuristic Bilinear	0.853$_{\pm.036}$	27.1$_{\pm1.3}$	0.139$_{\pm.029}$	0.123$_{\pm.011}$
Thin	DecFormer	**0.942**$_{\pm.018}$	**30.7**$_{\pm1.1}$	**0.063**$_{\pm.018}$	**0.091**$_{\pm.007}$
	Heuristic Bilinear	0.878$_{\pm.034}$	24.4$_{\pm1.0}$	0.139$_{\pm.028}$	0.193$_{\pm.013}$

Table 6: DecFormer vs. Heuristic bilinear at 256px resolution (mean ± 95% CI, n=50)

Mask Type	**Method**	**SSIM ↑**	**PSNR (dB) ↑**	**LPIPS ↓**	**Halo L1 ↓**
Soft ($sigma$=21)	DecFormer	**0.934**$_{\pm.007}$	**33.0**$_{\pm.7}$	**0.071**$_{\pm.007}$	**0.034**$_{\pm.003}$
	Heuristic Bilinear	0.804$_{\pm.019}$	26.0$_{\pm.7}$	0.204$_{\pm.018}$	0.082$_{\pm.006}$
Binary	DecFormer	**0.892**$_{\pm.029}$	**27.9**$_{\pm1.3}$	**0.098**$_{\pm.026}$	**0.097**$_{\pm.013}$
	Heuristic Bilinear	0.808$_{\pm.037}$	23.0$_{\pm1.0}$	0.187$_{\pm.032}$	0.172$_{\pm.013}$
Original	DecFormer	**0.902**$_{\pm.027}$	**29.7**$_{\pm1.3}$	**0.097**$_{\pm.026}$	**0.077**$_{\pm.009}$
	Heuristic Bilinear	0.812$_{\pm.037}$	24.3$_{\pm1.0}$	0.183$_{\pm.033}$	0.142$_{\pm.010}$
Thin	DecFormer	**0.911**$_{\pm.017}$	**27.4**$_{\pm1.0}$	**0.092**$_{\pm.017}$	**0.097**$_{\pm.006}$
	Heuristic Bilinear	0.809$_{\pm.032}$	21.5$_{\pm.8}$	0.202$_{\pm.027}$	0.199$_{\pm.010}$

E Alpha and Shift Visualisation and Target Visualisations

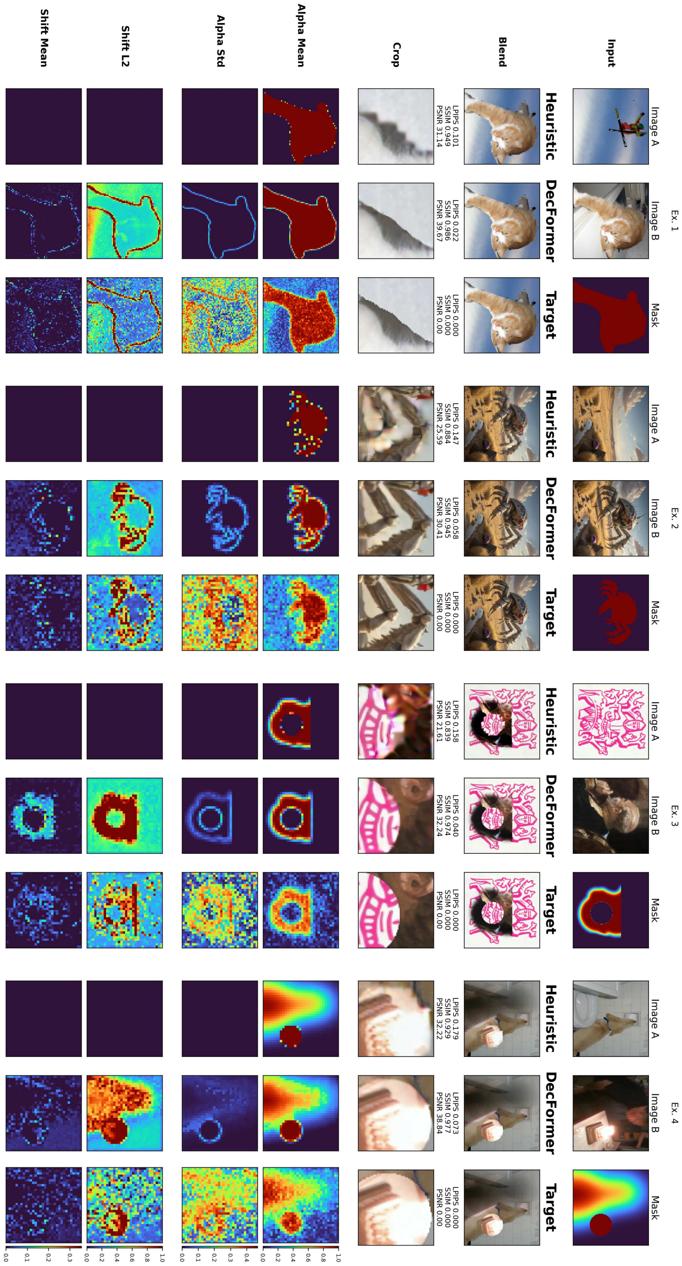


Figure 9: Qualitative comparison of DecFormer interpolation against a heuristic baseline and ground truth. For each method, we visualize the output alongside the corresponding α and shift predictions, compressed to 1-D profiles using multiple metrics. In the heuristic baseline, the naive mask collapses to a single scalar channel (zero variance), revealing its broadcast nature. The ground truth reference uses optimal $\alpha's$ values (Appendix A). Notably, the predicted shift and projected shift exhibits ring-like halos aligned with the mask boundaries.

F DecFormer Qualitative Extended

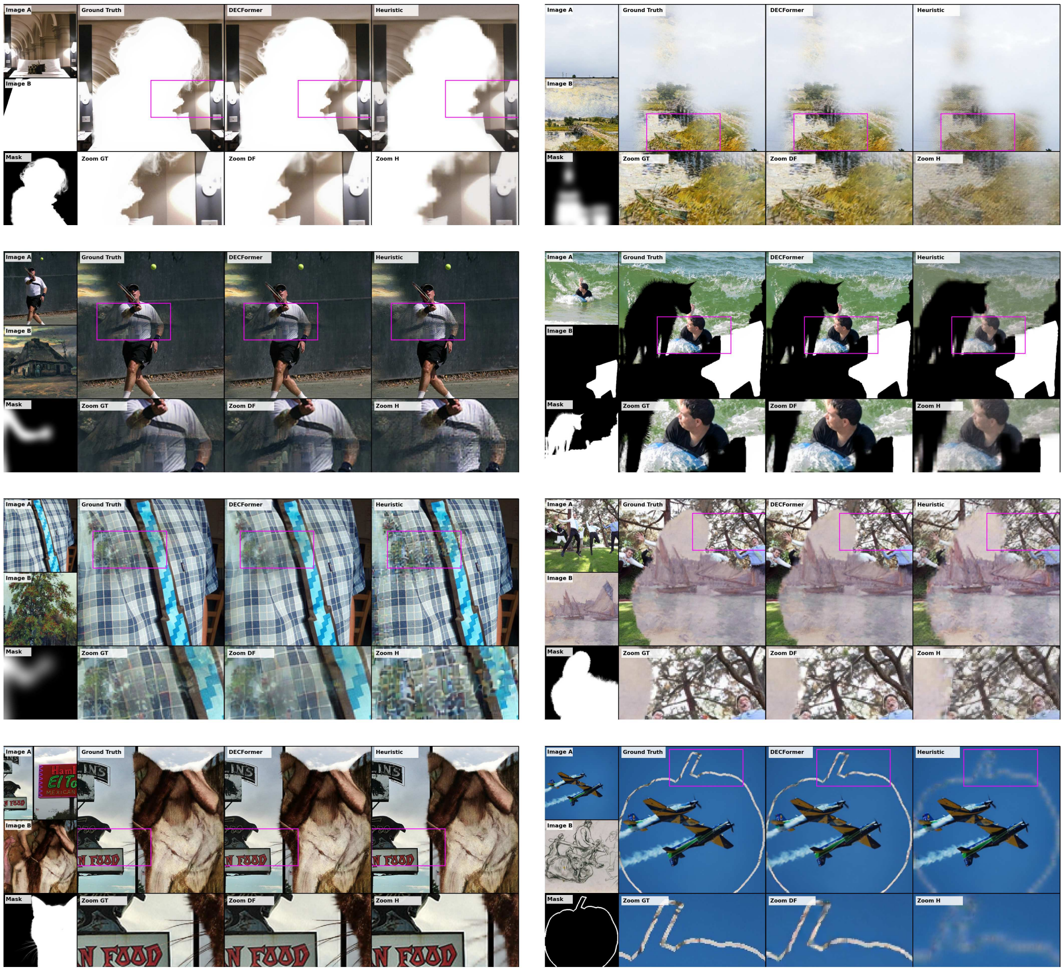


Figure 10: Further qualitative results illustrating the failure modes of heuristic interpolation and the improvements achieved by DecFormer (see Fig. 1).

G Halo calculation

Compute the 1-px edge set e of m (morphological XOR of 1-px dilate/erode). Convolve e with a linear disk kernel of radius R_{px} (We empirically find radius $R_{\text{px}} = 8$ (approximately one VAE receptive field) provides good coverage) to obtain a two-sided, softly decaying ring $w^{\text{px}} \in [0,1]^{H\times W}$. Anti-alias downsample m to m_ℓ, then reproduce the same ring construction at latent scale with radius $R_\ell = \max\big(1,\ \lfloor R_{\text{px}}/s \rceil\big)$, $s = \max(H/h,\ W/w)$, yielding $w^\ell \in [0,1]^{h\times w}$.

Staged training via local quadratic surrogate

Setup. Near a current iterate (α, s) we linearize the decoder D and form a local Gauss–Newton surrogate for the decoded losses. This yields a quadratic objective

$$\min_{\delta\alpha,\ \delta s}\ \frac{1}{2}\begin{bmatrix}\delta\alpha\\ \delta s\end{bmatrix}^{\top} \underbrace{\begin{bmatrix}M & B\\ B^{\top} & N\end{bmatrix}}_{H_{\text{loc}}} \begin{bmatrix}\delta\alpha\\ \delta s\end{bmatrix} - \left\langle g, \begin{bmatrix}\delta\alpha\\ \delta s\end{bmatrix}\right\rangle,$$

where $M \succ 0$ and $N \succ 0$ are the Gauss–Newton blocks for α and s, and B captures their interaction (concentrated near mask boundaries).

Block coordinate view. Training α first with $s{=}0$ and then s with α frozen is equivalent to applying a block Gauss–Seidel step on the local system. The second stage solves the Schur–complement system

$$S\,\delta s \;=\; r_s - B^{\top}M^{-1}r_\alpha, \qquad S \;=\; N - B^{\top}M^{-1}B,$$

which can be interpreted as preconditioning the joint problem by M along the blend axis.

Conditioning implication (local surrogate). Let $\kappa(\cdot)$ denote the spectral condition number. For the preconditioned local system one obtains the bound

$$\kappa_{\text{BCGD}} \;\le\; \kappa(M)\,\kappa(S), \qquad S = N - B^{\top}M^{-1}B \preceq N,$$

so κ_{BCGD} is no worse than using N alone and improves as the coupling B is explained by the α-update. Empirically, we observe a reduced spectrum of S (vs. N) concentrated at mask boundaries, aligning with faster convergence in phase 2 (Fig. **??**).[2]

Practical schedule. Motivated by this decomposition, we *stage* training: (i) optimize α with s gated off until validation stabilizes; (ii) warm up the shift head over 2k steps while reducing α's LR; (iii) ramp in halo-weighted losses to focus s on boundary residuals. This preserves a clean early gradient signal for α and directs s to the orthogonal residual where it is most needed.

H Gamma Correction Results

I Methods Extended

Mask priors. Naively downsampling m to latent size m_ℓ yields jagged, misaligned edges. We apply anti-aliased filtering and receptive-field–matched dilation, ensuring latent masks align with decoder ERF. A lightweight, two-headed CNN, run only once-per-mask, further maps the pixel mask to a content-agnostic prior α_0, seeding the blend and block 0 inputs, and providing dense mask tokens for cross-attention.

Conditioning. Blended latent errors concentrate near mask edges due to entangled channels. We therefore compute a pixel and latent-sized halo: a softly decaying band (≈8 px in pixel space, stride-scaled) around mask boundaries and soft regions. The halo serves two roles. First, it directly conditions the model (via FiLM) so that both α and s are aware of boundary context. Second, it provides a loss weighting that emphasizes precisely those regions where naïve interpolation fails.

[2] This statement is for the *local quadratic surrogate* induced by a frozen decoder Jacobian and squared decoded losses; in practice we use LPIPS and halo weightings, for which Gauss–Newton is a standard approximation.

Cross-attention to mask tokens is confined to the patch $= 1$ blocks. Global blocks already access coarse mask embeddings through FiLM; but fine-scale editing at pixel resolution requires precise spatial alignment, which attention provides. Restricting cross-attention to the final stage keeps compute low while preserving boundary fidelity. FiLM also recieves input from a latent sized mask image, which gives strong signal to the alpha head.

Update rule. Each block predicts $\Delta\alpha$ and Δs via $1{\times}1$ heads:

$$\alpha^{(b)} \leftarrow \mathrm{clip}_{[0,1]}\big(\alpha^{(b-1)} + \Delta\alpha^{(b)}\big), \quad s^{(b)} \leftarrow s^{(b-1)} + \Delta s^{(b)}.$$

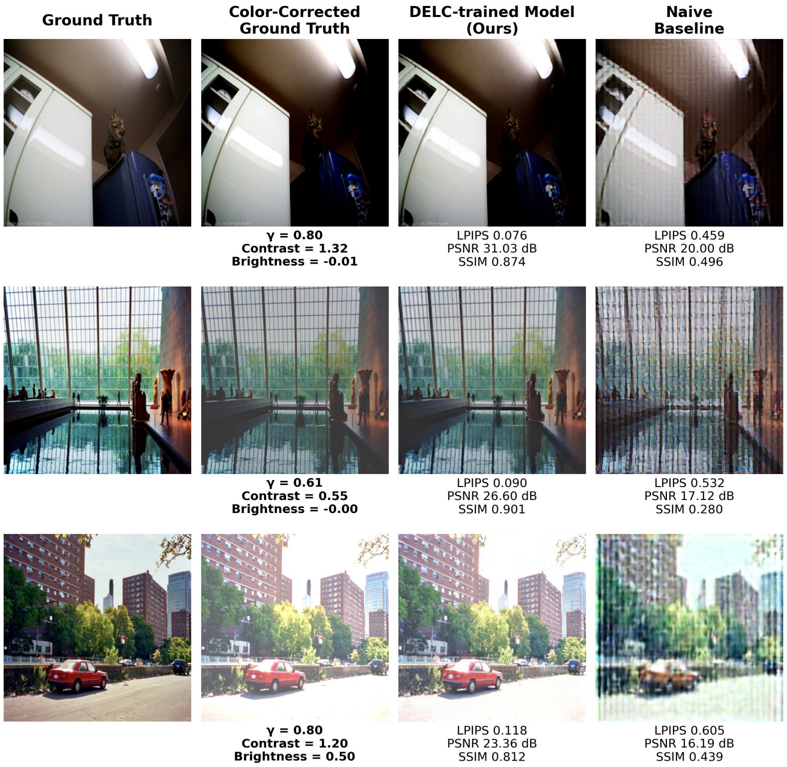
Updating in logit space for α prevents saturation and improves gradient flow.

Training objective. We construct a pixel-space blend $x_T = (1-m) \odot x_A + m \odot x_B$, encode once to obtain the target latent z_T, and predict $\hat{z}$. Supervision is applied after full reconstruction: losses are computed between $D(\hat{z})$ and the decoded ground truth $D(z_T)$, ensuring that errors from the VAE encoder–decoder cancel out rather than contaminate the objective. The loss combines a small latent MSE term with decoded LPIPS and a halo-weighted L1 that concentrates learning on boundary fidelity. Details of alpha, shift supervision, training and data details are in Appendix **??**.

Alpha-Shift separation To disentangle the roles of α and s, we experimented with several explicit regularizers: (i) a scale-aware L_1 penalty on s controlled by an EMA target magnitude, (ii) a cosine-hinge that penalizes $|\cos(s, d)|$ when aligned with $d = z_A - z_B$, and (iii) direct supervision against (α^*, s^*). However, we found that such constraints often over-regularized the model and hindered convergence. Instead, we adopt a staged training schedule: the shift head remains gated off until α has converged, at which point s is warmed up gradually (see Appx. G). Associated losses, including the halo-weighted L_1, are likewise ramped in during this phase, ensuring that α receives a clean learning signal early on—particularly important since α alone cannot correct decoder leakage at mask boundaries.

Training We train DecFormer on NVIDIA H100 GPUs. Each run uses a batch size of 8 and proceeds for 8×10^4 steps (approx 128 epochs, $\sim 10^6$ updates in total). Inputs are sampled at multi-resolution from $256{\times}256$ to $384{\times}384$ pixels with aspect ratios in $[0.5, 2.0]$. Mask augmentation includes graduated edge detection (0–15%) and feathering ramps over 1000 steps. Optimization employs AdamW with cosine SGDR: warm restarts at 4k, 12k, and 14k steps with $\eta_{\max} = 10^{-3}$, $\eta_{\min} = 2{\times}10^{-4}$, followed by cosine annealing from 30k to 60k steps down to 10^{-4}.

Data. We train DecFormer on a heterogeneous mix of images and masks. Our image set combines $\sim$30k natural photographs from Flickr30k, $\sim$10k artworks from WikiArt, and an additional $\sim$100k high-resolution images collected from internal web sources. Mask supervision is drawn from Composition-1k, P3M, GFM, together with procedurally generated random shapes.



(a) Qualitative Comparison