Appendix

A. Least-squares projection

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

$$(\alpha^*, s^*) = \arg\min_{\alpha,s} \left\| z_T - \left[(1-\alpha)z_A + \alpha z_B + s\right] \right\|_2^2 \tag{12}$$

$$\text{s.t.} \quad \alpha \in [0,1]^{C\times H\times W}. \tag{13}$$

This yields the closed-form solution

$$\alpha^*_{ijc} = \Pi_{[0,1]}\left(\frac{(z_T^{ijc} - z_B^{ijc})(z_A^{ijc} - z_B^{ijc})}{\|z_A^{ijc} - z_B^{ijc}\|_2^2 + \epsilon} \right), \tag{14}$$

$$s^*_{ijc} = z_T^{ijc} - \left[(1-\alpha^*_{ijc})z_A^{ijc} + \alpha^*_{ijc} z_B^{ijc}\right], \tag{15}$$

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. Motivation for the α/s Decomposition

Heuristic blending broadcasts a single downsampled mask across all latent channels, limiting mask resolution to $1/f$ and ignoring cross-channel structure. Our SDF analysis (Figure 4) shows that the resulting reconstruction error concentrates sharply at mask boundaries, with a diffuse global component from decoder entanglement.

We exploit this structure by decomposing the compositor into a per-channel blend weight α and a residual correction s. Channel-adaptive $\alpha \in [0,1]^{C\times h\times w}$ already recovers most of the compositing signal: it operates at full latent resolution with independent per-channel control, and can be initialized from a content-agnostic mask prior. The closed-form projection (Appendix A) confirms that per-channel α^* alone yields high-resolution masking far beyond the $1/f$ broadcast limit. The residual s then captures what α cannot: off-axis corrections from decoder curvature and nonlocal coupling that manifest as boundary halos and global color shifts. Because s models a spatially sparse, low-magnitude signal, it is a far simpler learning target than the full compositing function.

This factorization also enables staged training (Appendix J): α converges first under a clean gradient signal, then s is introduced to resolve boundary residuals, with halo-weighted losses ramped in to focus capacity where error concentrates. Table 1 confirms that removing either component degrades performance, and that an unconstrained single-head baseline collapses both roles and substantially underperforms.

Why constrain α to $[0,1]$? Constraining α to $[0,1]$ yields a stable blend axis: each channel's prediction is a convex combination of the two source latents, providing a strong inductive prior that most of the compositing signal lies on the $z_A \leftrightarrow z_B$ line. If α is left unconstrained it absorbs the role of both blending and residual correction, collapsing the two heads into one and destabilizing training (Table 1, "Unconstrained Alpha, No Shift" row).

Disentangling α and s: failed regularizers. Before arriving at staged training, we experimented with several explicit regularizers to encourage separation: (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 s aligns with the blend direction $d = z_A - z_B$, and (iii) direct per-voxel supervision against the closed-form (α^*, s^*). In all cases these constraints over-regularized the model and hindered convergence—particularly (iii), which forced the model to match a projection-optimal decomposition that is not necessarily training-optimal. The staged schedule (Appendix J) achieves separation implicitly: α receives a clean gradient signal during the first phase, then s learns only the orthogonal residual that α cannot capture. Associated losses, including the halo-weighted L_1, are ramped in during the second phase, ensuring that α is not retroactively destabilized. This is particularly important since α alone cannot correct decoder leakage at mask boundaries—without a converged α baseline, s has no stable reference to correct against.

C. Blind Predictor Architecture

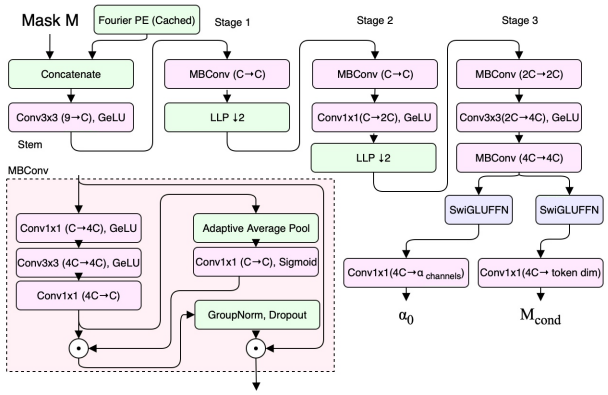

Figure 6. A lightweight CNN maps the input pixel mask (augmented with Fourier features) into latent-resolution, per-channel soft masks. The stem is a 3×3 conv with GELU, followed by three stages: Stage 1: MBConv block with depth-wise squeeze–excite, followed by a learnable low-pass filter and $2\times$ 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 DecFormer. 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.

D. DecFormer Extended Architecture

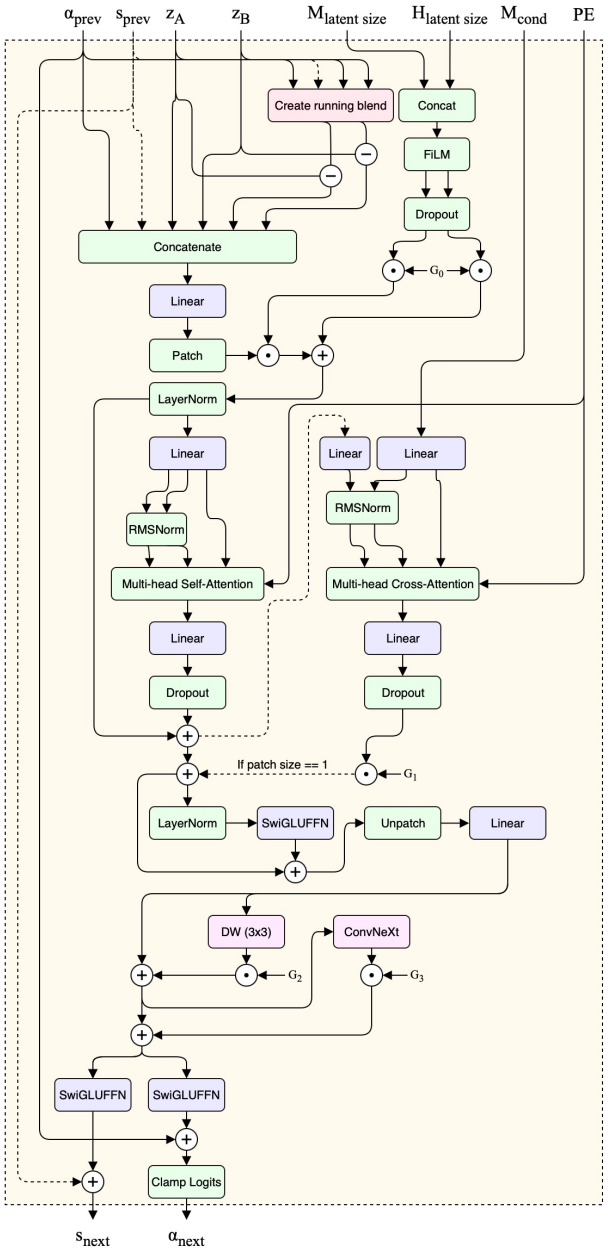

Figure 7. Extended DecFormer architecture diagram. The figure highlights more precisely the internal composition of each block, including the location and type of normalization layers, as well as the flow of intermediate projections and residual connections.

E. Gamma Correction Proof-of-Concept Results

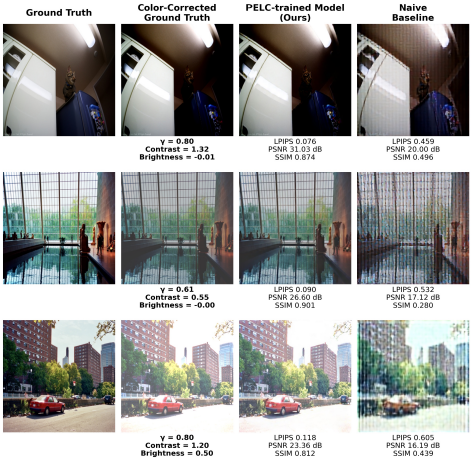

Figure 8. A qualitative example shows how the naive method fails catastrophically with severe artifacts when faced with more aggressive color correction, while our model's output is nearly indistinguishable from the ground truth. This illustrates that VAE latents are not pixel-like and require a principled framework like PELC.

Table 4. Color-correction generality experiment: quantitative metrics (mean $\pm$ 95% CI) on 1024 random samples from COCO-2017 validation. The baseline applies the color-transformation formula directly in latent space.

Metric	Baseline (heuristic)	PELC-trained (Ours)
LPIPS $\downarrow$	0.4996 ± 0.0076	0.0875 ± 0.0023
PSNR $\uparrow$	18.1630 ± 0.1968	27.2835 ± 0.2301
SSIM $\uparrow$	0.4359 ± 0.0112	0.8466 ± 0.0059

F. Extended DecFormer Metrics Tables

Table 5. 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 (σ=21)	DecFormer	$0.985_{\pm.003}$	$\mathbf{41.3}_{\pm.8}$	$\mathbf{0.027}_{\pm.005}$	$\mathbf{0.018}_{\pm.001}$
	DecFormer-Pretrain	$\mathbf{0.986}_{\pm.002}$	$40.9_{\pm.7}$	$0.028_{\pm.005}$	$\mathbf{0.018}_{\pm.001}$
	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	$\mathbf{0.964}_{\pm.017}$	$\mathbf{35.7}_{\pm1.5}$	$\mathbf{0.045}_{\pm.018}$	$\mathbf{0.060}_{\pm.006}$
	DecFormer-Pretrain	$0.961_{\pm.018}$	$34.8_{\pm1.5}$	$0.058_{\pm.022}$	$0.068_{\pm.005}$
	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	$\mathbf{0.968}_{\pm.016}$	$\mathbf{38.6}_{\pm1.5}$	$\mathbf{0.049}_{\pm.018}$	$\mathbf{0.037}_{\pm.005}$
	DecFormer-Pretrain	$0.965_{\pm.017}$	$37.9_{\pm1.4}$	$0.056_{\pm.021}$	$0.040_{\pm.005}$
	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	$\mathbf{0.967}_{\pm.014}$	$\mathbf{34.7}_{\pm1.5}$	$\mathbf{0.045}_{\pm.017}$	$\mathbf{0.073}_{\pm.005}$
	DecFormer-Pretrain	$0.960_{\pm.016}$	$33.4_{\pm1.4}$	$0.061_{\pm.020}$	$0.085_{\pm.005}$
	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 6. DecFormer vs. Heuristic bilinear at 512px resolution (mean ± 95% CI, n=50).

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

Table 7. DecFormer vs. Heuristic bilinear at 256px resolution (mean ± 95% CI, n=50)

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

G. Alpha and Shift Visualisation and Target Visualisations

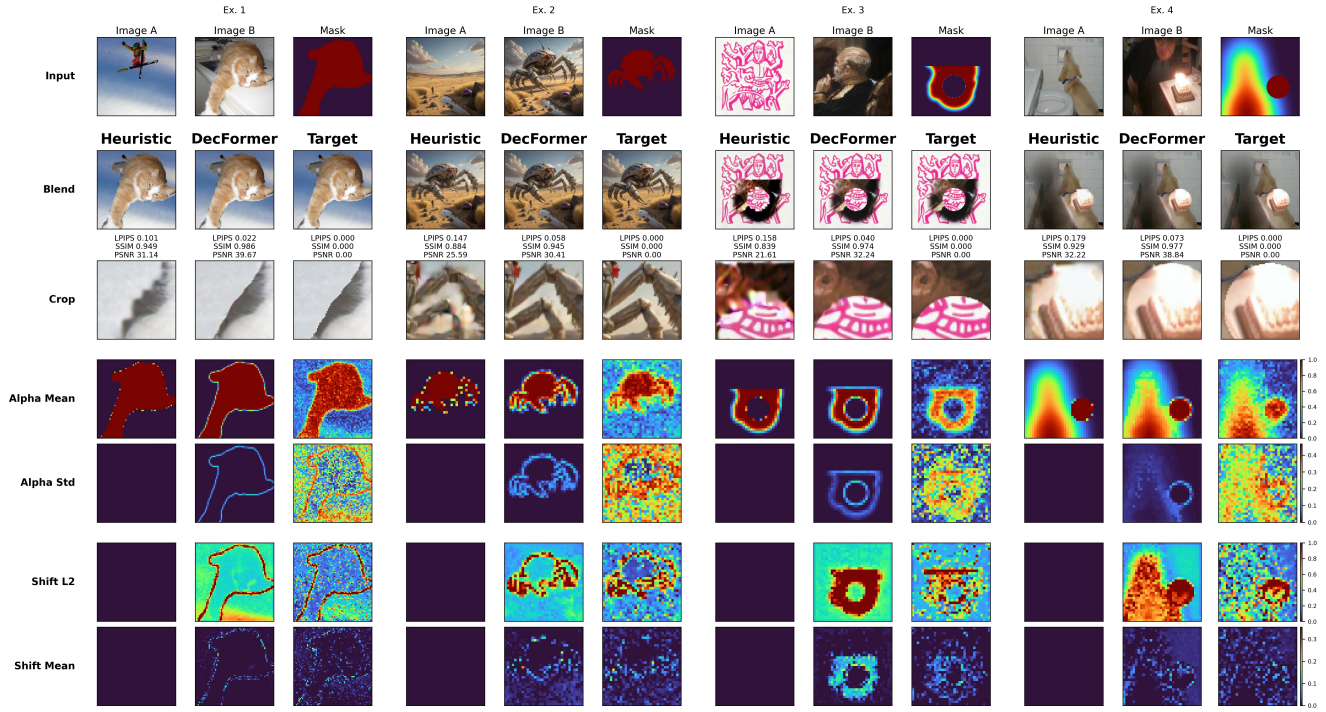

Figure 9. Qualitative comparison of DecFormer interpolation against a heuristic baseline and ground truth. For each method, we visualize the output alongside the corresponding α and shift predictions, compressed to 1-D profiles using multiple metrics. In the heuristic baseline, the naive mask collapses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

H. DecFormer Qualitative Extended

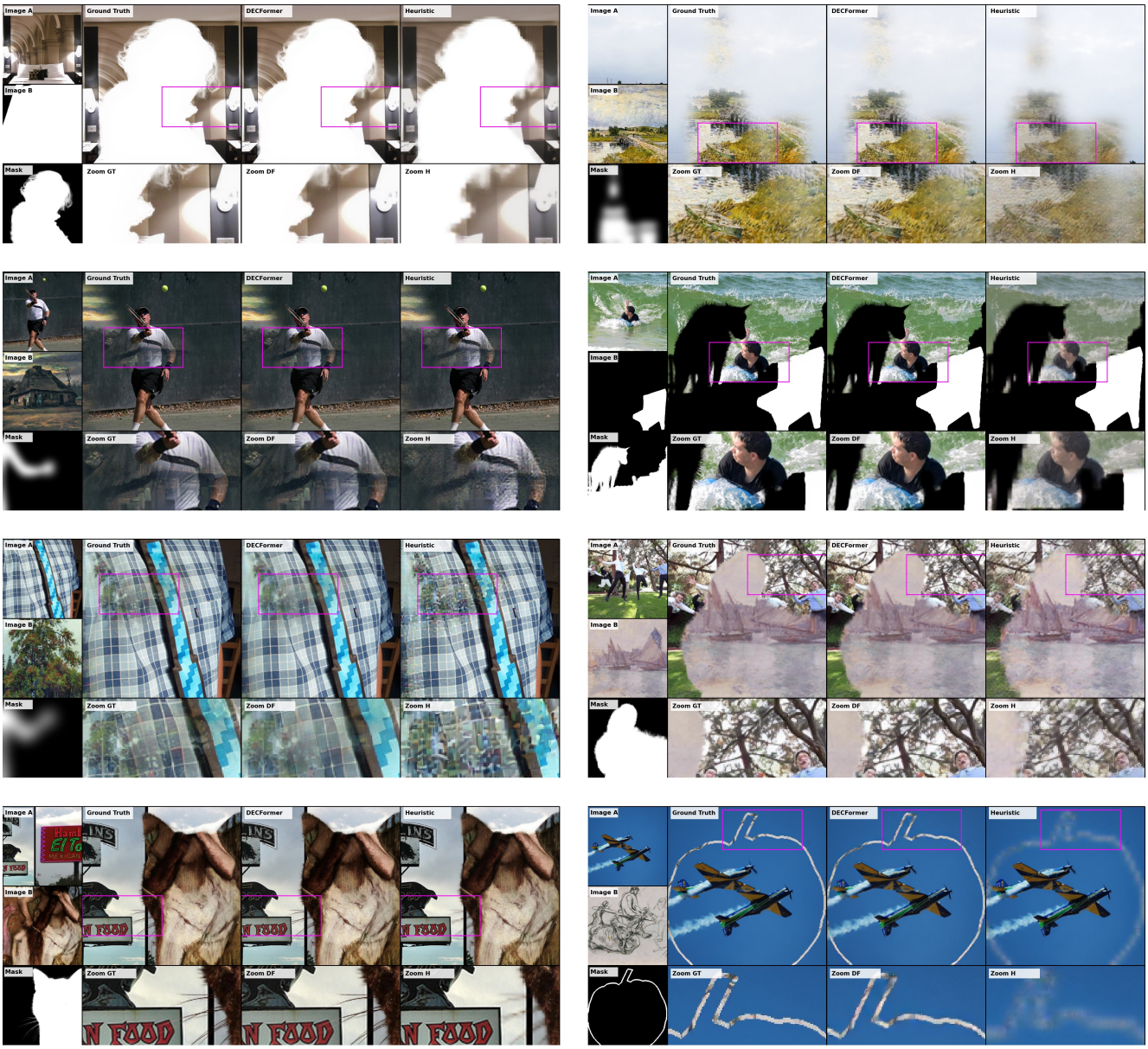

Figure 10. Further qualitative results illustrating the failure modes of heuristic interpolation and the improvements achieved by DecFormer (see Fig. 1).

I. Cross-Architecture Diagnostic: Qwen Image VAE

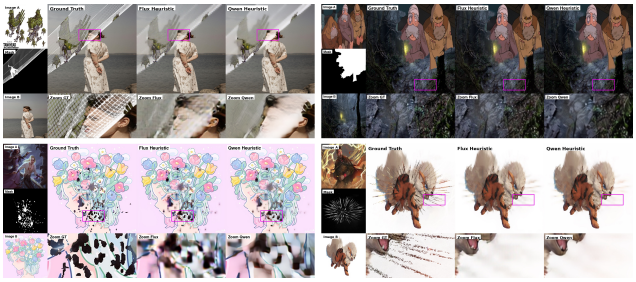

Figure 11. Heuristic latent compositing on Flux and Qwen Image VAEs. Both produce visibly degraded composites with blurred boundaries and color shifts, confirming that heuristic blending artifacts are not specific to the Flux VAE. Images are unselected.

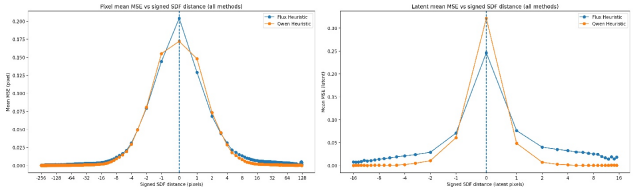

Figure 12. SDF-based error analysis comparing Flux and Qwen Image VAEs (cf. Figure 4). Left: mean MSE vs. pixel-space SDF. Right: mean MSE vs. latent-space SDF. Both VAEs exhibit the characteristic boundary error spike and decay, confirming that the failure of heuristic blending is a structural property of modern nonlinear VAEs.

J. 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. [3]

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.

[3] 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.

K. Receptive Field Analysis

We calculate the receptive field according to formula:

$$r_0 = \sum_{l=1}^{L} \left((k_l - 1) \prod_{i=1}^{l-1} s_i \right) + 1$$

we modify this formula to find influence field

$$i_0 = \sum_{l=1}^{L} \left((k_l - 1) \prod_{i=1}^{l-1} s_i \prod_{j=l-1}^{L} f_j \right) + \prod_{k=1}^{L} f_k$$

where f denotes the upscaling factor for each module, which will be 2 for linearly interpolating the hidden state, and 1 for other operations such as convolutions.

Table 8. Encoder receptive field

Layer	Effective Stride	Layer Sum	Cumulative field
Conv_in	1	2	3
Down L0	1	10	13
Down L1	2	20	33
Down L2	4	40	73
Down L3	8	64	137
Middle	8	64	201
Conv_out	8	16	217

Table 9. Decoder influence field

Layer	Effective Stride	Upscale Factor	Layer Sum	Cumulative field
Conv_in	1	1	2	3
Middle	1	1	8	11
Up L3	1	2	16	50
Up L2	2	2	32	156
Up L1	4	2	64	424
Up L0	8	1	96	520
Conv_out	8	1	16	536

Table 10. Note that for upscaling layers, the order is reversed, from starting from L3 to L0

Table 11. Decoder receptive field

Layer	Effective Stride	Layer Sum	Cumulative field
Conv_in	1	2	3
Middle	1	8	11
Up L3	1	13	24
Up L2	1/2	6.5	30.5
Up L1	1/4	3.25	33.75
Up L0	1/8	1.5	35.25
Conv_out	1/8	0.25	35.5

L. Dual-Sigma Noise for Mask-Aware Inpainting LoRA

Standard diffusion applies a single global noise level σ to all latent locations,

$$z_\sigma = (1 - \sigma) z_0 + \sigma\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I),$$

yielding a spatially uniform SNR. This provides no mechanism to distinguish pixels that are context from the inpainting region. More critically, in few-step samplers and velocity-based flows, the first update step largely fixes the denoising trajectory. Under uniform corruption, the model begins this first step from a state in which the true context is fully noised, forcing a blind prediction. The resulting early commitment often drives the masked and unmasked regions toward different modes, producing visible seams or the appearance of two incompatible images interpolated together.

To avoid this failure mode, we impose a SNR contrast between the two regions. The surrounding context is assigned lower noise (higher SNR), while the masked region receives higher noise (lower SNR). This ensures that the model sees an accurate representation of the context at the first denoising step, providing a reliable conditioning signal before any irreversible trajectory decisions are made.

Dual-sigma construction Let $m \in [0,1]^{H \times W}$ denote the latent-resolution mask. For each sample we draw $u \sim \mathcal{U}(0,1)$ and construct

$$\sigma_{\text{in}} = g(u), \qquad \sigma_{\text{out}} = g(\lambda u),$$

where $\lambda = 0.75 < 1$ is a fixed scalar controlling the maximum noise level applied to the context region.

and thus a corresponding SNR contrast

$$\text{SNR}_{\text{out}} = \frac{(1 - \sigma_{\text{out}})^2}{\sigma_{\text{out}}^2} \gg \frac{(1 - \sigma_{\text{in}})^2}{\sigma_{\text{in}}^2} = \text{SNR}_{\text{in}}.$$

Regionwise noising Per-region noisy latents are constructed as

$$z_\sigma = (1 - \sigma) z_0 + \sigma\, \epsilon,$$

and the composite latent is

$$z = m \odot z_{\sigma_{\text{in}}} + (1 - m) \odot z_{\sigma_{\text{out}}}.$$

This yields a piecewise noise field

$$\sigma(i,j) = m(i,j)\sigma_{\text{in}} + (1 - m(i,j))\sigma_{\text{out}},$$

in contrast to the spatially uniform σ used in standard diffusion. The denoiser naturally relies on the preserved context to guide reconstruction of the masked region, producing an inpainting-aware model without modifying architecture. This modification is not significantly out of distribution, and is easy to learn in a low rank manner. We train a 16 rank LoRA on this formulation.