Stable-Layers: Fine-Tuning Image Layer Decomposition Models with VLM-Scored Reinforcement Learning

Ciara Rowles
Stability AI

Reshinth Adithyan
Stability AI

Nikhil Pinnaparaju
Stability AI

Vikram Voleti
Stability AI

Mark Boss
Stability AI

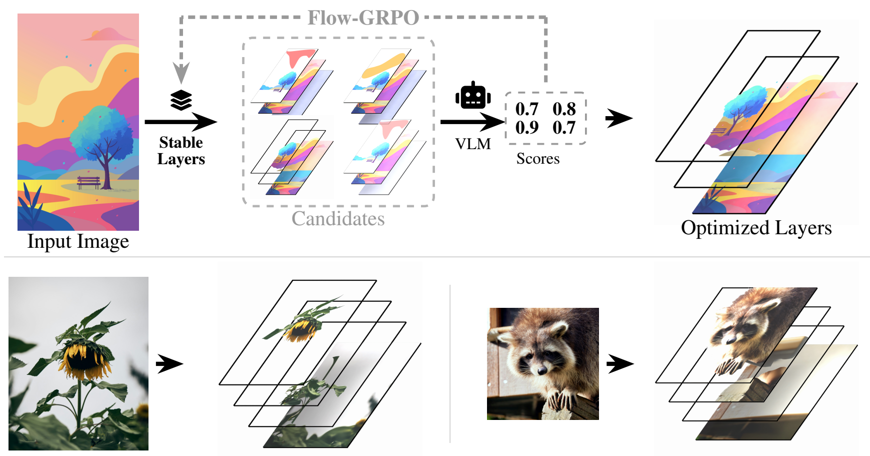

Figure 1: **Stable-Layers.** We finetune a layer decomposition model using Flow-GRPO and a VLM judge, and improve layerization without relying on paired data. The resulting layers have improved consistency, separation and handle in-painting of occluded areas better.

Abstract

We present Stable-Layers, a reinforcement learning framework that eliminates the need for paired supervision by fine-tuning a pretrained layer decomposition model using only feedback from a vision-language model (VLM). Starting from Qwen-Image-Layered, we apply Flow-GRPO with LoRA adaptation, sampling multiple candidate decompositions per image, scoring them with a VLM, and optimising the policy from group-relative advantages. The key challenge lies in designing a reliable reward signal: VLMs scoring samples in isolation tend to compress their judgements into a narrow band, leaving GRPO with little within-group variance to learn from. We address this with a two-stage evaluation pipeline that pairs structured per-sample scoring across five edit-centric criteria with a grid-based calibration step in which the VLM re-scores all candidates side-by-side. Trained entirely on unlabelled images, Stable-Layers produces decompositions with stronger layer separation, fewer blank or artifact-heavy layers, and lower per-layer reconstruction error on the Crello dataset compared to the base model.

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

1 Introduction

Image layer decomposition—separating an image into a small set of editable RGBA layers whose composition reconstructs the original (Figure 1)—is a fundamental primitive for professional editing and compositing [35]. While the task is easy to define, it is difficult to supervise: a single image admits many plausible decompositions, and quality is ultimately determined by downstream usability, including semantic separation, clean alpha mattes, minimal redundancy, and faithful handling of occluded content, rather than similarity to any single target decomposition. Existing methods circumvent this ambiguity using synthetic layered datasets [12, 13, 35], but such supervision imposes an inherent limitation: when multiple decompositions are equally valid, regression toward a single target penalizes alternative solutions. We address this limitation through a post-hoc reinforcement learning refinement stage that optimizes directly for perceived decomposition quality using a vision-language model (VLM) as the sole source of supervision [5].

However, applying VLM-as-judge feedback to layer decomposition introduces a reward design challenge not encountered by standard approaches. Decomposition quality is inherently multi-dimensional, spanning semantic disentanglement, alpha cleanliness, inpainting plausibility, feature allocation, and content validity, with strong correlations across these axes: candidates within a sampled group are often simultaneously good or bad along most criteria. Existing scalar VLM rewards [29] collapse these dimensions into a single compressed signal, while pairwise preference approaches [25] scale quadratically with group size and lose absolute calibration. Moreover, naive rubric scoring produces low within-group variance when candidates are visually similar, weakening the learning signal for GRPO. To address this, we introduce a two-phase evaluation protocol. In Phase 1, the VLM performs structured criterion-wise scoring using explicit rubric anchors for each quality dimension. In Phase 2, the candidate group is jointly re-evaluated on a labelled comparison grid to obtain finer relative calibration between similar decompositions. The two phases serve complementary roles: absolute scoring effectively captures categorical failures, while relative comparison sharpens discrimination between perceptually close candidates.

A second challenge is optimization stability. GRPO-Guard's RatioNorm [26] uses a spatial mean of per-element log-probabilities, but Qwen-Image-Layered packs N RGBA layers into a single latent sequence, inflating effective dimensionality D by $\sim 5\times$ and suppressing per-step log-ratio standard deviation as $1/\sqrt{D}$. To address this, we introduce a simple sum-and-rescale modification that restores $\mathcal{O}(1)$ ratio magnitudes while remaining broadly applicable to flow-matching RL settings with sequence-packed latents.

We instantiate our framework on Qwen-Image-Layered [35] using LoRA adaptation [9], training entirely on Fine-T2I [20] images without any layer annotations. Our contributions are: **(i)** a two-phase VLM reward protocol that alleviates score compression in within-group reinforcement learning; **(ii)** a RatioNorm reformulation tailored to packed latent representations in flow-matching RL; and **(iii)** substantially improved layer decompositions over the Qwen-Image-Layered baseline, yielding better semantic separation, cleaner layers, and lower per-layer reconstruction error on the Crello dataset [32] and held-out evaluations.

In summary, Stable-Layers serves as a general recipe for training edit-oriented generators using *judge feedback* instead of targets. The specific optimization machinery is a tool to achieve this goal: convert VLM judgments over sampled candidates into learning signals that directly enhance editability.

2 Related Work

RL and reward modelling for visual generation. DDPO [1] and DPOK [8] first cast diffusion sampling as a multi-step MDP and applied policy gradients to optimise non-differentiable rewards; gradient-based alternatives such as DRaFT [6] and AlignProp [21] backpropagate through the sampling chain when the reward is differentiable, while Diffusion-DPO [25] sidesteps online rollouts by optimising a preference-based objective on static comparison data. For flow-matching specifically, Flow-GRPO [16] introduces a marginal-preserving SDE that yields tractable log-probabilities for GRPO-style [23] clipped objectives, DanceGRPO [31] validates group-relative updates at scale, and GRPO-Guard [26] stabilises the importance ratio via normalisation and gradient reweighting. Reward signals for these methods range from learned scalar rewards (ImageReward [30], HPS v2 [28]) to VLM-derived rewards: scalar rewards from pretrained encoders [29], pairwise preferences for

DPO-style training [37], and self-improving VLM critics [14, 27]. TOPReward [4] extracts token-completion logits to bypass the brittleness of text-generated numeric scores, and MJ-Bench [5] studies VLM reliability as a judge. Our training loop follows Flow-GRPO with GRPO-Guard stabilisation, but replaces scalar or pairwise reward interfaces with structured multi-criteria VLM judgements (alpha cleanliness, semantic separation, content validity) that enable richer credit assignment over layer stacks.

Image layer decomposition and generation. Recovering or synthesizing layered image representations has been approached from multiple directions. Text-to-layer generation methods, including LayerDiff [12], DreamLayer [11], LayerFusion [7], PSDiffusion [10], and LayeringDiff [13], produce multi-layer raster outputs via inter-layer attention, harmonized decoding, or generate-then-disassemble pipelines. LayerDiffuse [36] encodes alpha-channel transparency in the latent manifold of a pretrained diffusion model for direct RGBA generation. On the decomposition side, LayerDecomp [33] and LASAGNA [34] separate foreground and background while preserving visual effects; Referring Layer Decomposition [2] conditions on user prompts; CLD [19] introduces fine-grained controllable multi-layer separation; and Chen *et al.* [3] repurpose inpainting models for layer recovery. Domain-specific methods target graphic designs [24], anime characters [15], and illustration production workflows [38]. Most closely related to our work, Qwen-Image-Layered [35] is an end-to-end diffusion model that decomposes a single RGB image into a variable number of RGBA layers using an RGBA-VAE, a Variable Layers Decomposition MMDiT, and multi-stage supervised training on Photoshop PSD data. We take Qwen-Image-Layered as our base model and show that GRPO-based reinforcement learning with VLM-as-judge rewards can further improve decomposition quality beyond what supervised training alone achieves.

3 Background

Our method builds on three components: flow matching as the generative framework, an SDE-augmented variant that exposes tractable per-step log-probabilities, and GRPO for policy optimisation.

Flow Matching and Rectified Flows Rectified flow [17, 18] interpolates $x_t = (1-t)\,x_0 + t\,x_1$ between data $x_0 \sim \pi_{\mathrm{ref}}$ and noise $x_1 \sim \mathcal{N}(0, I)$, and learns a velocity field $v_\theta(x_t, t)$ parameterized by θ via the regression loss $\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{t,x_0,x_1}\left[\|v_\theta(x_t,t) - (x_1 - x_0)\|^2\right]$. At inference, samples are produced by integrating the ODE $x_{t-\Delta t} = x_t + v_\theta(x_t, t)\,\Delta t$. This deterministic trajectory has no per-step randomness and thus no tractable log-probability for policy gradients; the SDE formulation below addresses this.

SDE-Augmented Flow Matching (Flow-GRPO) Flow-GRPO [16] augments the deterministic ODE with a stochastic differential equation that preserves the learned marginals, enabling tractable log-probability computation for RL. With diffusion coefficient $\sigma_t = a\sqrt{t/(1-t)}$ (we use a=0.7, within the $[0.7, 0.9]$ range recommended by Liu et al. [16]), the SDE transition for a step from t to $t - \Delta t$ is:

$$\mu_{t\to t-\Delta t} = x_t\left(1 - \frac{\sigma_t^2}{2t}\Delta t\right) - v_\theta(x_t,t)\left(1 + \frac{\sigma_t^2(1-t)}{2t}\right)\Delta t, \tag{1}$$

$$x_{t-\Delta t} = \mu_{t\to t-\Delta t} + \sigma_t\sqrt{\Delta t}\,\epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{2}$$

Group Relative Policy Optimization For a group of G samples $\{x^{(g)}\}_{g=1}^{G}$ from the same condition, GRPO [23] computes within-group advantage $\hat{A}^{(g)}$ from each sample's reward $r^{(g)}$ as:

$$\hat{A}^{(g)} = \frac{r^{(g)} - \bar{r}}{\sigma_r + \nu}, \tag{3}$$

where $\bar{r}$ and σ_r are the within-group mean and standard deviation of rewards and ν=10^{-4} is a small constant for numerical stability. GRPO optimises a clipped surrogate $\mathcal{L}_{\mathrm{GRPO}} = -\mathbb{E}_g[\min(\rho_g \hat{A}^{(g)},\ \mathrm{clip}(\rho_g, 1-\epsilon_c, 1+\epsilon_c)\hat{A}^{(g)})] + \beta\,\mathrm{KL}[\pi_\theta\|\pi_{\mathrm{ref}}]$, where $\rho_g = \exp(\log \pi_\theta - \log \pi_{\theta_{\mathrm{old}}})$ is the importance ratio.

Stabilised Ratio Clipping (GRPO-Guard) Wang et al. [26] observe that in flow-matching models the importance ratio ρ_g exhibits a systematic leftward shift (mean below 1) with timestep-dependent

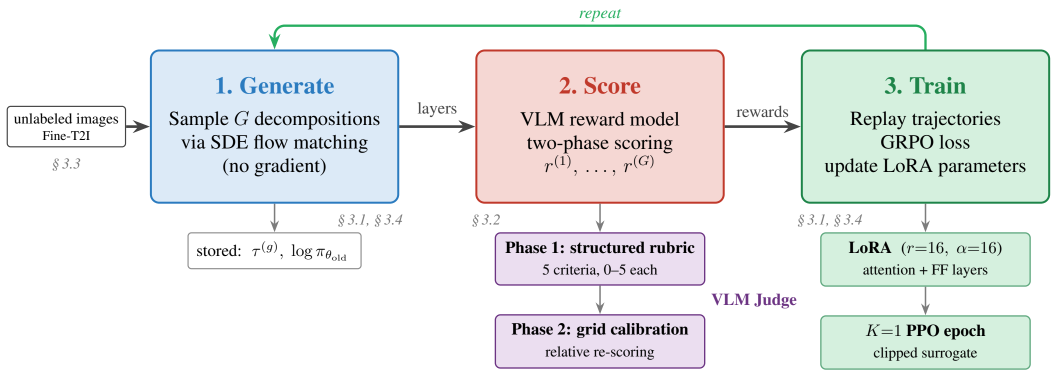

Figure 2: **Stable-Layers training pipeline.** Sample G candidates, score with the two-phase VLM reward, replay with GRPO updates to LoRA parameters.

variance, preventing the clipped surrogate from constraining overconfident positive-advantage updates. GRPO-Guard addresses this with two corrections: (i) *RatioNorm*, which standardizes $\log \rho_g$ per denoising step so that the ratio distribution is centered near 1 with uniform variance across steps, restoring effective clipping; and (ii) *gradient reweighting* by $\delta = 1/\Delta t$, which equalizes per-step gradient magnitudes and prevents low-noise timesteps from dominating the update, allowing the KL term to be omitted.

4 Method

Stable-Layers fine-tunes a pretrained layer decomposition model using only unlabeled images and post-hoc VLM judgements as supervision—no layer annotations, paired examples, or synthetic decomposition targets are required. We adopt Flow-GRPO's three-phase training loop (Figure 2): each step generates G candidate decompositions via SDE sampling, scores them with the VLM reward protocol of Section 4.2, and replays the stored trajectories to compute GRPO updates. The reward design (Section 4.2), data strategy (Section 4.3), and adaptation to an editing model (Section 4.1) are our contributions; the SDE formulation and GRPO objective follow Flow-GRPO directly.

4.1 Model Architecture and Adaptation

The base model we use is Qwen-Image-Layered [35], a flow-matching transformer [17, 18] that generates N-layer RGBA decompositions conditioned on an input image. The architecture comprises a 3D variational autoencoder (VAE) that encodes 4-channel RGBA frames with $8\times$ spatial compression into a 16-channel latent space, a sequence-based transformer operating on 2×2 patch-packed latents (yielding token dimension $16 \times 4 = 64$), and a text encoder for prompt conditioning. The condition image is encoded through the same VAE pipeline and concatenated along the sequence dimension of the transformer input, with per-frame spatial metadata provided to the attention mechanism to distinguish generated layer tokens from conditioning tokens.

We apply Low-Rank Adaptation [9] with rank r=16 and α=16 to all attention projection layers and feed-forward layers, keeping all other parameters frozen.

4.2 VLM Reward Design

The central challenge in applying RL to layer decomposition is defining a reward signal that captures the multi-dimensional notion of decomposition quality without requiring ground-truth targets. We address this with a two-phase VLM scoring pipeline, designed to avoid specific failure modes of the base model while maintaining inter-group discrimination. Each criterion in our rubric is anchored by explicit descriptions of qualifying high- and low-score conditions, which we found necessary for consistent VLM judgements; the full rubric is provided in Appendix B.

4.2.1 Image Presentation for the VLM Judge

VLMs are trained predominantly on RGB and cannot meaningfully interpret raw alpha channels, so we composite each layer onto a solid white background before presenting it: transparent regions

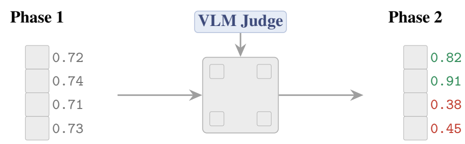

Figure 3: **Phase 2 grid calibration.** We re-score candidates relative to each other, spreading compressed Phase 1 scores and restoring within-group variance for GRPO advantage normalisation.

render as white, and the VLM assesses alpha quality by observing where content transitions to the white background. Each sample is presented as the RGB composite alongside N white-background layer images at 320×320 for Phase 1.

4.2.2 Phase 1: Structured Individual Scoring

Each generated sample is sent independently to the VLM, which evaluates five criteria on a 0–5 integer scale, each anchored by explicit descriptions of the worst-case (0) and best-case (5) visual conditions. The criteria are: **semantic separation** (each foreground layer isolates one distinct object), **alpha cleanliness** (foreground masks are crisp and binary-like), **background inpainting** (layer 0 is a plausible scene completion), **feature distribution** (content is spread across layers rather than concentrated), and **content validity** (layers are not blank or noise-only). The five scores sum to a total in $[0, 25]$ and are normalised to $[0, 1]$. The full prompt with per-criterion anchor descriptions is in Appendix B.

4.2.3 Phase 2: Relative Grid Calibration

To sharpen within-group discrimination when Phase 1 scores ($r_{\text{ind}}^{(g)}$) compress, we tile all G composites into a labelled comparison grid at a resolution of 256x256 (Figure 3) and ask the VLM to re-score each sample relative to the others ($r_{\text{cal}}^{(g)}$), given the Phase 1 scores as context. Construction details (cell size, label format, full prompt) are in Appendix B.

We use the calibrated score directly: $r^{(g)} = r_{\text{cal}}^{(g)}$. Phase 2 conditions on the Phase 1 scores (Appendix B). The no-calibration baseline (Section 6.5) substitutes $r_{\text{ind}}^{(g)}$.

4.3 Training Data: Judge-Only Supervision without Synthetic Targets

The primary source is Fine-T2I [20], an aesthetically filtered subset of photographs and artworks. All images are resized to 640×640 and normalised to $[-1, 1]$; The images are shuffled at each epoch. At each training step, the number of output layers is sampled uniformly from [min_layers, max_layers] (typically $[2, 5]$), exposing the model to variable decomposition complexity throughout training. While the base Qwen-Image-Layered model supports decompositions of 20 layers, the memory and compute cost of GRPO scales with both the group size G and the number of layers per sample (each additional layer adds tokens to the transformer's sequence and a separate VLM scoring pass), so we restrict training to at most five layers per sample. The trained LoRA can be applied with the full range of possible output layer numbers during inference.

4.4 GRPO Training with Trajectory Replay

We follow Flow-GRPO's trajectory replay procedure with one modification to GRPO-Guard's RatioNorm: because Qwen-Image-Layered packs multiple RGBA layers into a single high-dimensional latent sequence at 640×640, the standard spatial-mean log-ratio collapses toward zero. We instead sum log-probabilities over spatial dimensions and normalise by $\sqrt{D}$ before applying GRPO-Guard's per-step centring, preserving $\mathcal{O}(1)$ ratios while retaining RatioNorm's centring and variance-stabilisation properties. Full hyperparameters, SDE step schedules, and CFG settings are in Appendix F.

5 Experimental Setup

The images from the training dataset are resized to 640×640 and normalised to $[-1, 1]$. The number of output layers per step is sampled uniformly from $[2, 5]$ during training. For the Crello dataset evaluation (Table 1) we restrict generation to $L \in \{2, 3, 4\}$ for direct comparison against the base model on the same layer counts.

5.1 Baselines

We compare Stable-Layers (full two-phase reward with grid calibration) against two reference points:

Base model. The base Qwen-Image-Layered checkpoint with no RL fine-tuning. This establishes the baseline to compare to for the recipe.

Flow-GRPO without calibration. The same GRPO training pipeline with identical hyperparameters, LoRA configuration, and data, but using only Phase 1 individual scoring as the reward signal, no grid calibration phase).

Comparison with LayerD. We additionally compare against LayerD [24], which represents a different point in the design space: a method that declines to decompose under uncertainty, frequently returning the input largely intact as a single layer rather than producing a multi-layer separation. This is a valid design choice with different downstream implications than ours, and the comparison is included to characterise the behavioural contrast. Setup details are in Appendix H.

We do not compare against supervised fine-tuning (SFT) with reconstruction loss, as this requires paired ground-truth layer decompositions that do not exist for natural images—the supervision gap that motivates Stable-Layers.

6 Results

6.1 Reward Progression

The calibrated VLM reward rises from $\sim$0.70 to $\sim$0.83 over the first $\sim$100 steps as the policy eliminates the worst failure modes, then plateaus with high per-step variance for the remainder of training (Figure 8), even as held-out evaluation metrics continue to improve (Figure 4). This plateau is expected under GRPO: because advantages are normalised *within* each group (Equation (3)), learning requires only sufficient within-group variance to distinguish better candidates from worse, not rising absolute scores—once the coarse failure modes are resolved, all candidates in a group tend to improve together, keeping the group mean roughly stationary while relative ranking continues to provide gradient signal. The residual variance in the curve reflects conditioning-image difficulty across the dataset more than policy quality.

Figure 4: **Held-out evaluation metrics.** Three automated metrics on 480 LAION-Aesthetics [22] images across training. **Top:** bad layers per decomposition (blank + glaze; lower is better) fall from $\sim$1.65 to $\sim$0.4. **Middle:** feature distribution evenness (higher is better) rises from $\sim$0.53 to $\sim$0.73. **Bottom:** layer 0 inpainting quality (higher is better) rises from $\sim$0.38 to $\sim$0.62. Bands show $\pm 1\sigma$ across the set.

6.2 Qualitative Results

Figure 5 compares decompositions from the base model (Qwen-Image-Layered) and the Stable-Layers model on two held-out inputs: a natural photograph (a person crossing a red rope bridge) and a vector-style illustration (a tree and bench in a stylised landscape). Two failure modes of the base model are visible across both examples and addressed after fine-tuning. First, layer 0 is degenerate—a fully black layer for the bridge scene and a flat cream fill for the illustration, neither

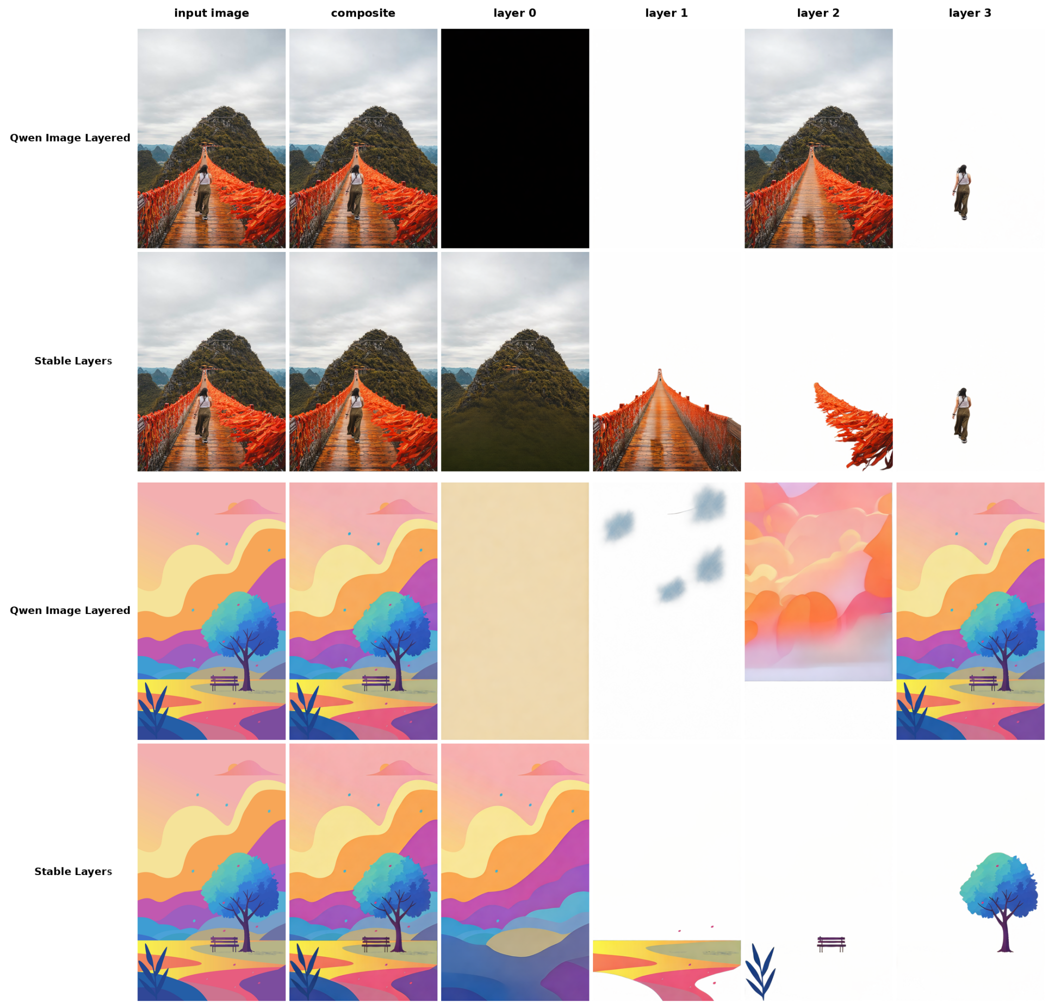

Figure 5: **Qualitative comparison on held-out images.** Base model (*Qwen Image Layered*, top of each pair) vs. Stable-Layers-fine-tuned model (*Stable-Layers*, bottom). Columns show the input, the composite, and individual layers on white backgrounds. The fine-tuned model produces plausible background inpainting on layer 0 and isolates distinct semantic elements across foreground layers, where the base model leaves layer 0 degenerate and duplicates the composite across foreground slots.

of which represents a usable inpainted background. After fine-tuning, layer 0 contains a plausible scene completion: the mountain and sky behind the bridge, and the rolling hills and clouds of the illustration, demonstrating that the VLM reward successfully penalises trivial inpainting. Second, the base model duplicates near-complete copies of the input across foreground layers (most clearly in the bridge example, where layer 2 is essentially the entire composite minus the person). The fine-tuned model instead isolates distinct semantic elements onto separate layers—the bridge deck, the rope railings, and the person in the photograph; the foreground path, plants, bench, and tree in the illustration—with cleaner alpha masks and less colour bleed into transparent regions.

6.3 Quantitative Evaluation

We evaluate per-layer reconstruction quality on the Crello dataset [32] test set using a metric adapted from Yin et al. [35], with one modification: best-match assignment in place of fixed-index comparison, since RL fine-tuning can reorder layers without changing decomposition quality (full justification in Appendix I).

Table 1 reports per-layer RGB L1 against best-matched ground-truth layers, stratified by output layer count. Stable-Layers achieves lower mean error than the base model across all layer counts. At the

Table 1: Per-layer RGB L1 vs Crello ground-truth test set layers — each predicted layer is scored against its closest GT layer; lower is better). *Mean* is the mean over all predicted layers; *pred* k reports layer k individually. m = number of matched Crello test set images. Bold marks the best *Mean* within each L group.

Layers	Variant	m	Mean ↓	Pred 0 ↓	Pred 1 ↓	Pred 2 ↓	Pred 3 ↓
L=2	Qwen-Image-Layered	3	0.1706	0.2938	**0.0473**	—	—
	Stable-Layers	3	**0.1635**	**0.2511**	0.0760	—	—
L=3	Qwen-Image-Layered	29	0.0879	0.0734	0.0938	**0.0965**	—
	Stable-Layers	29	**0.0767**	**0.0502**	**0.0786**	0.1012	—
L=4	Qwen-Image-Layered	90	0.0712	0.0954	0.0848	0.0590	**0.0457**
	Stable-Layers	90	**0.0660**	**0.0795**	**0.0678**	**0.0573**	0.0594

Table 2: Comparison on the held-out LAION-Aesthetics set (n=480) at four output layers. LayerD frequently returns fewer than four layers; unfilled slots are padded with empty layers for metric computation. Higher is better for both columns; bold marks the best per column.

Variant	Distrib. ↑	Layer 0 Q ↑
Qwen-Image-Layered	0.5282	0.3817
Stable-Layers	**0.7339**	0.6148
LayerD	0.0585	**0.7136**

individual-layer level, the fine-tuned model reduces dominance of layer 0 (Pred 0), consistent with the improved background inpainting seen in Figure 5. Small per-slot regressions (e.g. Pred 1 at L=2 and Pred 3 at L=4) reflect content reorganisation under best-match assignment: the fine-tuned model redistributes content across slots, so the slot that previously held the smallest residual (a near-empty layer in the base model) is now populated with real content and matches a different GT layer. Mean error is the relevant aggregate, and improves at every L.

Comparison with LayerD. LayerD [24] and Stable-Layers take different approaches to decomposition uncertainty. LayerD is conservative: when separation is hard, it tends to return the input largely intact as a single layer, producing fewer but more confident outputs. Stable-Layers always populates the requested number of layers. The two strategies have different downstream implications, and Table 2 reflects this: Stable-Layers achieves substantially higher distribution evenness because it actually fills the requested slots with distinct content, while LayerD scores marginally higher on Layer 0 quality because an unmodified copy of the input is by construction a plausible scene. For most editing workflows— where the value of a decomposition is in having usable separate layers—Stable-Layers's behaviour is the more useful one; the Layer 0 gap reflects an artifact of the metric rewarding faithful pixel content rather than a genuine quality advantage.

6.4 Ablation: Effect of Text Conditioning

We compare two fixed prompts applied uniformly across all training images: a *basic* prompt and a *detailed* prompt mirroring the reward rubric's evaluation axes (full text in Appendix J). Ablation runs use G=8, all other hyperparameters match Appendix F.

The detailed-prompt run underperforms the main run on every axis. Bad layers fall more slowly, feature distribution evenness plateaus lower, and Layer 0 quality actively degrades from ∼0.44 to ∼0.32 where the main run improves from ∼0.40 to ∼0.74. Conditioning on a prompt that describes an idealised multi-object scene may give the policy model a sense of direction the VLM judge might evaluate.

6.5 Ablation: Effect of Grid Calibration

To isolate the contribution of the relative grid calibration phase (Section 4.2.3), we train two otherwise identical runs—one with the full two-phase reward, one with Phase 1 individual scoring alone—and evaluate every 40 steps on a held-out set of 480 LAION-Aesthetics images [22]. Results are in Table 4.

Table 3: Text conditioning ablation across training steps (480-image eval). Comparing a basic fixed prompt against a detailed fixed prompt mirroring the reward rubric, both applied uniformly across training images. "Bad" is the average number of blank or over-glazed layers per generation; *Distrib.* is feature distribution evenness; *L0 Q.* is Layer 0 quality. Arrows indicate desired direction.

	Basic Prompt			Detailed prompt		
Step	Bad↓	Distrib.↑	L0 Q.↑	Bad↓	Distrib.↑	L0 Q.↑
0	1.600	0.526	0.403	1.692	0.536	0.435
40	0.792	0.631	0.517	1.433	0.660	0.349
80	0.368	0.692	0.554	1.267	0.713	0.369
120	0.245	0.726	0.573	1.362	0.688	0.316
160	0.226	0.739	0.643	1.015	0.690	0.297
200	0.335	0.733	0.738	0.612	0.691	0.323

Table 4: Calibration ablation across training steps (480-image eval). "Bad" is the average number of blank or over-glazed layers per generation; *Qual.*, *Sharp.*, and *SSIM* are Layer 0 quality metrics.

	No Calibration				With Calibration			
Step	Qual.↑	Sharp.↑	SSIM↑	Bad↓	Qual.↑	Sharp.↑	SSIM↑	Bad↓
0	0.611	0.236	0.579	1.008	0.611	0.236	0.579	1.008
40	0.585	0.189	0.500	0.600	0.595	0.198	0.495	0.581
80	0.560	0.154	0.437	0.346	0.590	0.184	0.489	0.560
120	0.564	0.139	0.420	0.298	0.598	0.198	0.514	0.500
160	0.577	0.164	0.444	0.392	0.618	0.238	0.556	0.627
200	0.587	0.205	0.522	0.658	0.637	0.278	0.548	0.398

Bad layer reduction is largely unaffected. Both runs reduce bad layers from 1.008 to 0.4–0.7 across mid-training, with neither variant consistently leading. Phase 1's content-validity and alpha-cleanliness criteria already produce enough variance on these binary-like defects for GRPO to learn from.

Image quality benefits from calibration. All three Layer 0 quality metrics separate the two runs from step 80 onward. SSIM averages 0.52 (calibrated) vs. 0.45 (uncalibrated) across steps 80–200; combined quality and edge sharpness show the same pattern. The gap matches the score-compression hypothesis behind Phase 2: when all candidates in a group are non-degenerate, the remaining quality differences (inpainting plausibility, edge quality) compress into a narrow Phase 1 band, yielding near-uniform advantages. Forcing explicit relative judgments restores the within-group variance the policy update requires. More broadly, coarse failure modes are well-captured by absolute scoring; fine-grained perceptual quality benefits from relative calibration.

7 Conclusion

We have presented Stable-Layers, a method for improving image layer decomposition models through reinforcement learning with VLM-provided rewards. Combining Flow-GRPO's SDE-augmented policy optimisation with a two-phase VLM scoring protocol—structured per-sample evaluation followed by relative grid calibration—we convert black-box judge feedback into a learning signal discriminative enough to drive fine-grained improvements in layer separation, content validity, and feature distribution, without task-specific reward training, synthetic targets, or human annotation.

The recipe generalises: any conditional generator whose outputs can be meaningfully evaluated by a VLM (style transfer, inpainting, relighting, scene rearrangement) could in principle be fine-tuned with the loop. Promising extensions include replacing the grid calibration's free-form numeric output with logit-based pairwise preferences (avoiding the failure mode TOPReward [4] targets) and automating rubric design itself by having a VLM critique its own scoring criteria.

Limitations. Stable-Layers relies on a proprietary VLM as the reward model, which introduces API cost per training step and a dependency on a specific model snapshot whose score distribution may drift across versions. Our evaluation relies on automated metrics and qualitative inspection rather than human studies; the metrics correlate with editing usefulness but do not directly measure it. Finally, training was capped at five layers per sample for compute reasons, so behaviour on high-layer-count decompositions (the base model supports up to 20) is not directly evaluated.

References

[1] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024.

[2] Fangyi Chen, Yaojie Shen, Lu Xu, Ye Yuan, Shu Zhang, Yulei Niu, and Longyin Wen. Referring layer decomposition. *arXiv preprint arXiv:2602.19358*, 2026.

[3] Jingxi Chen, Yixiao Zhang, Xiaoye Qian, Zongxia Li, Cornelia Fermuller, Caren Chen, and Yiannis Aloimonos. From inpainting to layer decomposition: Repurposing generative inpainting models for image layer decomposition. *arXiv preprint arXiv:2511.20996*, 2025.

[4] Shirui Chen, Cole Harrison, Ying-Chun Lee, Angela Jin Yang, Zhongzheng Ren, Lillian J Ratliff, Jiafei Duan, Dieter Fox, and Ranjay Krishna. Topreward: Token probabilities as hidden zero-shot rewards for robotics. *arXiv preprint arXiv:2602.19313*, 2026.

[5] Zhaorun Chen, Yichao Du, Zichen Wen, Yiyang Zhou, Chenhang Cui, Zhenzhen Weng, Haoqin Tu, Chaoqi Wang, Zhe Tong, Qing Huang, Canyu Chen, Qinghao Ye, Zhihong Zhu, Yuqing Zhang, Jiawei Zhou, Zhuokai Zhao, Rafael Rafailov, Chelsea Finn, and Huaxiu Yao. MJ-bench: Is your multimodal reward model really a good judge for text-to-image generation? *arXiv preprint*, 2024.

[6] Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations (ICLR)*, 2024.

[7] Yusuf Dalva, Yijun Li, Qing Liu, Nanxuan Zhao, Jianming Zhang, Zhe Lin, and Pinar Yanardag. LayerFusion: Harmonized multi-layer text-to-image generation with generative priors. *arXiv preprint*, 2024.

[8] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Neural Information Processing Systems (NeurIPS)*, 2023.

[9] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint*, 2021.

[10] Dingbang Huang, Wenbo Li, Yifei Zhao, Xinyu Pan, Yanhong Zeng, and Bo Dai. PSDiffusion: Harmonized multi-layer image generation via layout and appearance alignment. In *Winter Conference on Applications of Computer Vision (WACV)*, 2026.

[11] Junjia Huang, Pengxiang Yan, Jinhang Cai, Jiyang Liu, Zhao Wang, Yitong Wang, Xinglong Wu, and Guanbin Li. Dreamlayer: Simultaneous multi-layer generation via diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3357–3366, 2025.

[12] Runhui Huang et al. LayerDiff: Exploring text-guided multi-layered composable image synthesis via layer-collaborative diffusion model. In *European Conference on Computer Vision (ECCV)*, 2024.

[13] Kyoungkook Kang, Gyujin Sim, Geonung Kim, Donguk Kim, Seungho Nam, and Sunghyun Cho. Layeringdiff: Layered image synthesis via generation, then disassembly with generative knowledge. *arXiv preprint arXiv:2501.01197*, 2025.

[14] Zongxia Li, Wenhao Yu, Chengsong Huang, Rui Liu, Zhenwen Liang, Fuxiao Liu, Jingxi Che, Dian Yu, Jordan Boyd-Graber, Haitao Mi, et al. Self-rewarding vision-language model via reasoning decomposition. *arXiv preprint arXiv:2508.19652*, 2025.

[15] Jian Lin, Chengze Li, Haoyun Qin, Kwun Wang Chan, Yanghua Jin, Hanyuan Liu, Stephen Chun Wang Choy, and Xueting Liu. See-through: Single-image layer decomposition for anime characters. *arXiv preprint arXiv:2602.03749*, 2026.

[16] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

[17] Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint*, 2022.

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

[19] Zihao Liu, Zunnan Xu, Shi Shu, Jun Zhou, Ruicheng Zhang, Zhenchao Tang, and Xiu Li. Controllable layer decomposition for reversible multi-layer image generation. *arXiv preprint*, 2025.

[20] Xu Ma, Yitian Zhang, Qihua Dong, and Yun Fu. Fine-T2I: An open, large-scale, and diverse dataset for high-quality T2I fine-tuning. *arXiv preprint*, 2026.

[21] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint*, 2023.

[22] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. *Neural Information Processing Systems (NeurIPS)*, 35, 2022.

[23] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint*, 2024.

[24] Tomoyuki Suzuki, Kang-Jun Liu, Naoto Inoue, and Kota Yamaguchi. LayerD: Decomposing raster graphic designs into layers. In *International Conference on Computer Vision (ICCV)*, 2025.

[25] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purber, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[26] Jing Wang, Jiajun Liang, Jie Liu, Henglin Liu, Gongye Liu, Jun Zheng, Wanyuan Pang, Ao Ma, Zhenyu Xie, Xintao Wang, et al. Grpo-guard: Mitigating implicit over-optimization in flow matching via regulated clipping. *arXiv preprint*, 2025.

[27] Wen Wen, Tianwu Zhi, Kanglong Fan, Yang Li, Xinge Peng, Yabin Zhang, Yiting Liao, Junlin Li, and Li Zhang. Self-evolving vision-language models for image quality assessment via voting and ranking. *arXiv preprint*, 2025.

[28] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint*, 2023.

[29] Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, et al. Visionreward: Fine-grained multi-dimensional human preference learning for image and video generation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 11269–11277, 2026.

[30] Jiazheng Xu et al. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Neural Information Processing Systems (NeurIPS)*, 2023.

[31] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. DanceGRPO: Unleashing GRPO on visual generation. *arXiv preprint*, 2025.

[32] Kota Yamaguchi. CanvasVAE: Learning to generate vector graphic documents. *International Conference on Computer Vision (ICCV)*, 2021.

[33] Jinrui Yang, Qing Liu, Yijun Li, Soo Ye Kim, Daniil Pakhomov, Mengwei Ren, Jianming Zhang, Zhe Lin, Cihang Xie, and Yuyin Zhou. Generative image layer decomposition with visual effects. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7643–7653, 2025.

[34] Jinrui Yang, Qing Liu, Yijun Li, Mengwei Ren, Letian Zhang, Zhe Lin, Cihang Xie, and Yuyin Zhou. Controllable layered image generation for real-world editing. *arXiv preprint*, 2026.

[35] Shengming Yin, Zekai Zhang, Zecheng Tang, Kaiyuan Gao, Xiao Xu, Kun Yan, Jiahao Li, Yilei Chen, Yuxiang Chen, Heung-Yeung Shum, et al. Qwen-image-layered: Towards inherent editability via layer decomposition. *arXiv preprint arXiv:2512.15603*, 2025.

[36] Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. *ACM Transactions on Graphics (TOG)*, 2024.

[37] Ruohong Zhang, Liangke Gui, Zhiqing Sun, Yihao Feng, Keyang Xu, Yuanhan Zhang, Di Fu, Chunyuan Li, Alexander G Hauptmann, Yonatan Bisk, et al. Direct preference optimization of video large multimodal models from language model reward. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 694–717, 2025.

[38] Tianyu Zhang, Dongchi Li, Keiichi Sawada, and Haoran Xie. Workflow-aware structured layer decomposition for illustration production. *arXiv preprint*, 2026.

A Algorithm Pseudocode

Algorithm 1 Stable-Layers Training Step (adapted from Flow-GRPO with GRPO-Guard stabilisation)

Require: Policy π_θ (LoRA), reference π_{ref}, reward model R, group size G, latent dimensionality D, advantage clip c_{adv}, ratio clip ϵ_c, KL weight β, learning rate η

Sample condition image x_{cond} and x_{text} prompt.
Encode condition: $z_{\text{cond}} \leftarrow \text{VAE.encode}(x_{\text{cond}}) + x_{\text{text}}$.
// Phase 1: Generate
for $g = 1, \ldots, G$ **do**
  $z_T^{(g)} \sim \mathcal{N}(0, I)$
  $(z_0^{(g)},\ \tau^{(g)},\ \log \pi_{\theta_{\text{old}}}^{(g)}) \leftarrow \text{SDE\_Generate}(z_T^{(g)}, z_{\text{cond}})$
  $\text{layers}^{(g)} \leftarrow \text{VAE.decode}(z_0^{(g)})$
end for
// Phase 2: Score
for $g = 1, \ldots, G$ **do**
  $r^{(g)} \leftarrow R\big(\text{Composite}(\text{layers}^{(g)}),\ \text{layers}^{(g)}\big)$
end for
$\hat{A}^{(g)} \leftarrow \text{clip}\Big(\frac{r^{(g)} - \bar{r}}{\sigma_r + \nu},\ -c_{\text{adv}},\ c_{\text{adv}}\Big)$
// Phase 3: Train
for $g = 1, \ldots, G$ **do**
  for each SDE step i in $\tau^{(g)}$ **do**
    $\log \pi_\theta^{(g,i)},\ \log \pi_{\text{ref}}^{(g,i)} \leftarrow \text{Replay}(\tau_i^{(g)}, \pi_\theta, \pi_{\text{ref}})$
    $\log \rho^{(g,i)} \leftarrow \Big(\sum_d \log \pi_\theta^{(g,i)}[d] - \sum_d \log \pi_{\theta_{\text{old}}}^{(g,i)}[d]\Big) \ / \ \sqrt{D}$ ▷ sum-and-rescale RatioNorm
    $\log \rho^{(g,i)} \leftarrow \big(\log \rho^{(g,i)} - \mu_i\big)/s_i$ ▷ per-step centring (mean μ_i, scale s_i across group)
    $\rho^{(g,i)} \leftarrow \exp\big(\log \rho^{(g,i)}\big)$
    $\widehat{\text{KL}}^{(g,i)} \leftarrow \log \pi_\theta^{(g,i)} - \log \pi_{\text{ref}}^{(g,i)}$
    $\mathcal{L}^{(g,i)} \leftarrow -\frac{1}{\Delta t_i} \min\big(\rho^{(g,i)} \hat{A}^{(g)},\ \text{clip}(\rho^{(g,i)}, 1-\epsilon_c, 1+\epsilon_c)\, \hat{A}^{(g)}\big) + \beta\, \widehat{\text{KL}}^{(g,i)}$ ▷ Gradient reweight by $1/\Delta t$
  end for
end for
$\theta \leftarrow \theta - \eta \nabla_\theta \frac{1}{G} \sum_g \frac{1}{|\tau^{(g)}|} \sum_i \mathcal{L}^{(g,i)}$

B Reward Prompt Template

This appendix provides the exact prompts sent to the VLM reward model (`gemini-3-flash-preview`) during training.

Reward model version. All reward model calls use `gemini-3-flash-preview` via the Google AI Studio API, pinned to the model snapshot available between October 2025 and the date of submission. Researchers extending the method should expect score distributions to drift across model versions; we recommend re-calibrating the Phase 1 anchor descriptions when switching reward models.

B.1 System Prompt

The following system message is prepended to all reward model calls:

```
You are an expert image compositor evaluating layer decomposition
quality.  You will see an original composite image and its
decomposition into separate layers.  Layer 0 is ALWAYS the
background flat.  The remaining layers are foreground elements.
Score ONLY based on what you see.  Be harsh and discriminating -
give different scores to samples that genuinely differ in quality.
```

```
Do NOT give identical scores to all samples. Respond ONLY with
valid JSON, no other text.
```

B.2 Phase 1: Individual Scoring Prompt

Each sample is presented as the composite image followed by N layer images (composited onto white backgrounds), with the following instruction appended. The placeholder `{num_layers}` is replaced with the number of layers for that training step.

```
Score this {num_layers}-layer decomposition. Layer 0 is the
background; layers 1+ are foreground.
CRITERIA (score each 0-5):
1. semantic_separation (0-5): Each foreground layer should contain
ONE distinct, complete object or semantic element (e.g. a person,
a car, a tree). Score 0 if a single object is arbitrarily split
across multiple layers or if layers contain random crops/slices
of the scene rather than meaningful elements. Score 5 if every
foreground layer isolates a complete, distinct object and no object
is split across layers.
2. alpha_cleanliness (0-5): Foreground layers should have crisp,
binary-like alpha with clean edges. Score 0 if layers show a
semi-transparent haze, ghosting, colour bleed, or a milky/glazed
wash over areas that should be fully transparent. Transparent
regions must be FULLY transparent with zero colour residue. Score
5 if alpha masks are sharp, edges are clean, and transparent regions
are completely clear with no residual colour or haze.
3. background_inpainting (0-5): Layer 0 (the background) should
look like a plausible complete scene with foreground objects removed
and their regions filled in convincingly. Score 0 if the background
is blurry, has obvious holes, smeared patches, or copy-paste
artifacts where foreground objects were removed. Score 5 if the
inpainted regions blend seamlessly with the surrounding background,
maintaining consistent texture, lighting, and detail.
4. feature_distribution (0-5): Visual content should be
meaningfully spread across layers. Score 0 if most content is
crammed into one layer while others are blank or near-empty. Score
5 if layers have a balanced, meaningful distribution of the scene's
content.
5. content_validity (0-5): Penalize blank, empty, or noise-only
layers. Score 0 if most layers are blank or contain only noise/blur.
Score 5 if all layers have clear, recognizable content.
- total (0-25): Sum of all five scores.
Return ONLY valid JSON: {"semantic_separation":X,
"alpha_cleanliness":Y, "background_inpainting":Z,
"feature_distribution":W, "content_validity":V, "total":T}
```

B.3 Phase 2: Grid Calibration Prompt

Grid construction. The G Phase 1 RGB composites are arranged left-to-right, top-to-bottom in a $\lceil\sqrt{G}\rceil \times \lceil G/\lceil\sqrt{G}\rceil\rceil$ tiling on a white canvas (for our default $G{=}16$, a $3{\times}2$ grid). Each cell is rendered at $320{\times}320$ with a uniform white margin between cells, and the integer index $0, \ldots, G{-}1$ is rasterised in the top-left corner of each cell as a black sans-serif numeral on a white background. The completed grid is sent to the VLM as a single PNG.

Prompt. The grid is sent alongside the following instruction. Placeholders `{G}`, `{Gm1}`, and `{scores_csv}` are replaced with the group size, $G{-}1$, and the comma-separated Phase 1 scores respectively.

```
The grid shows {G} layer-decomposition samples arranged
left-to-right, top-to-bottom, labeled 0-{Gm1}.
Initial individual scores: {scores_csv}
```

```
Re-score each sample RELATIVE to the others.  Give higher scores to
better decompositions and lower to worse ones.  Be discriminating -
spread the scores.  Pay special attention to:
- Which samples keep whole objects on single layers vs. splitting
them?
- Which samples have that semi-transparent glaze/ghosting vs. clean
alpha?
- Which samples have convincing background inpainting vs. blurry
fills?
Reply with ONLY {G} comma-separated decimal values in [0,1], one per
sample in order:
```

C Additional Qualitative Examples

Figure 6 presents an extended gallery of layer decompositions produced by the Stable-Layers-fine-tuned model on held-out images from LAION-Aesthetics, spanning a range of subject matter: natural photographs (landscapes, wildlife, portraits), studio product shots, automotive renders, and scene compositions with varying foreground complexity.

D Additional Calibration Ablation Metrics

Figure 7 reports two additional Layer 0 quality metrics for the calibration ablation of Section 6.5: a combined quality score and an edge-density sharpness measure. Both show the same qualitative pattern as the SSIM result in the main text—the calibrated run maintains a small but consistent lead over the uncalibrated run from the mid-training checkpoints onward—providing converging evidence that grid calibration improves fine-grained background quality without affecting bad-layer reduction.

E Architecture Details

The base Qwen-Image-Layered [35] is a flow-matching transformer [17, 18] that produces N-layer RGBA decompositions conditioned on an input image. The architecture comprises:

- A 3D variational autoencoder (VAE) that encodes 4-channel RGBA frames with $8\times$ spatial compression into a 16-channel latent space.
- A sequence-based transformer operating on 2×2 patch-packed latents, yielding token dimension $16 \times 4 = 64$.
- A text encoder for prompt conditioning.

The condition image is encoded through the same VAE pipeline and concatenated along the sequence dimension of the transformer input, with per-frame spatial metadata provided to the attention mechanism to distinguish generated layer tokens from conditioning tokens.

We apply Low-Rank Adaptation [9] with rank r=16 and α=16 to all attention projection layers and feed-forward layers, keeping all other parameters frozen.

F Training Implementation Details

SDE schedule and CFG. During the generation phase we use a reduced schedule of T_{train} SDE steps (typically 8, compared to 50 at inference) to keep memory and compute costs tractable across G group samples. Following Flow-GRPO, classifier-free guidance is disabled during training (CFG $=$ 1.0, halving the number of forward passes per step) but enabled at evaluation (CFG $= 4.0$); Liu et al. [16] found that this asymmetry did not degrade final sample quality while substantially reducing training cost.

Trajectory replay. For each stored SDE step i, the current policy's transition mean μ_i^θ is recomputed via a forward pass through the LoRA-adapted transformer with gradients enabled. The KL reference is computed by running the same forward pass with LoRA adapters disabled, yielding the pre-adaptation base model's prediction at zero additional memory cost.

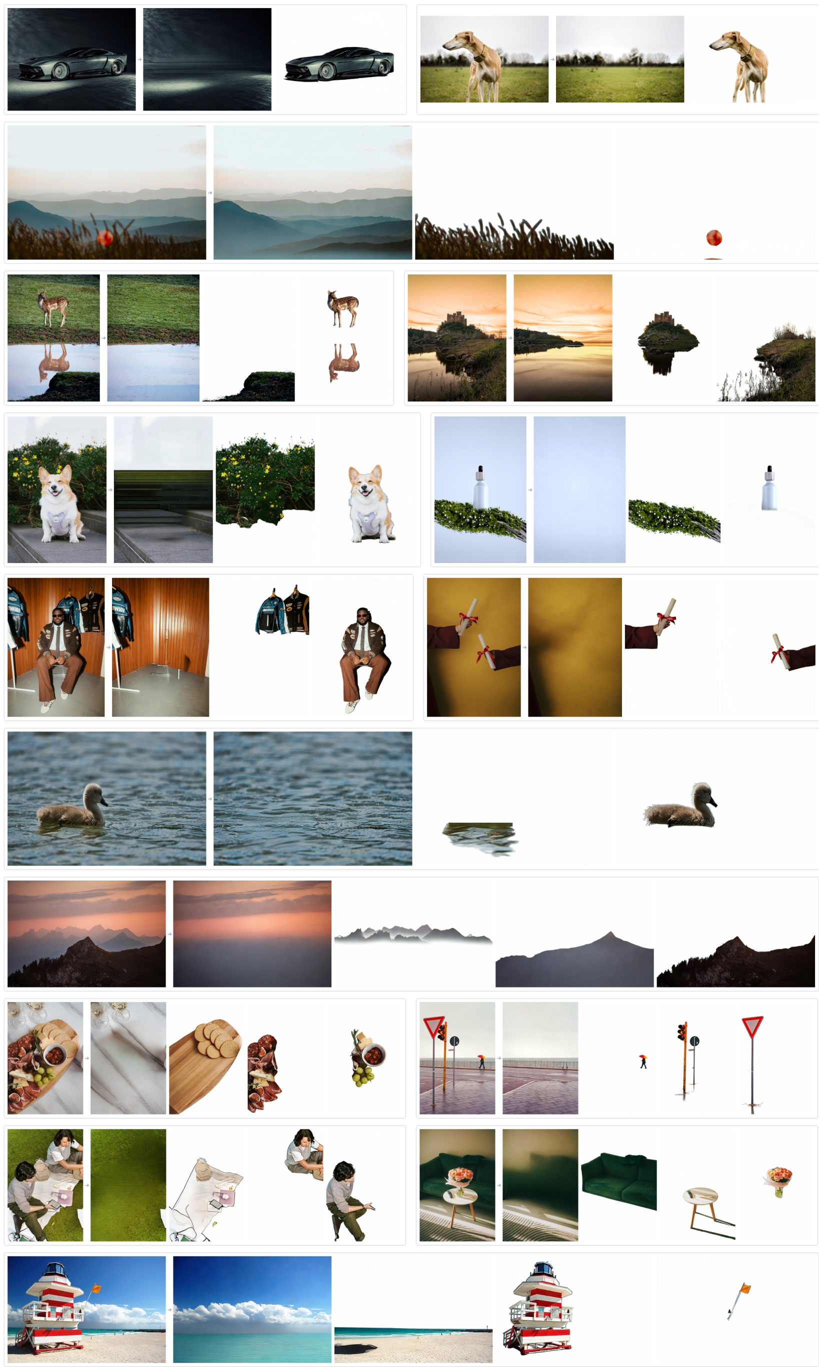

Figure 6: **Extended qualitative gallery.** Layer decompositions from the Stable-Layers-fine-tuned model on a diverse set of held-out inputs. Each row shows the reconstructed composite and the individual layers composited onto white backgrounds. Examples illustrate consistent behaviour across subject matter: clean isolation of foreground objects (corgi, swan, deer, jacket), plausible background inpainting where foreground elements are removed (lighthouse, statue island, mountain scene), and reasonable separation of multiple foreground elements onto distinct layers (charcuterie board, yield-sign scene, couch and side table).

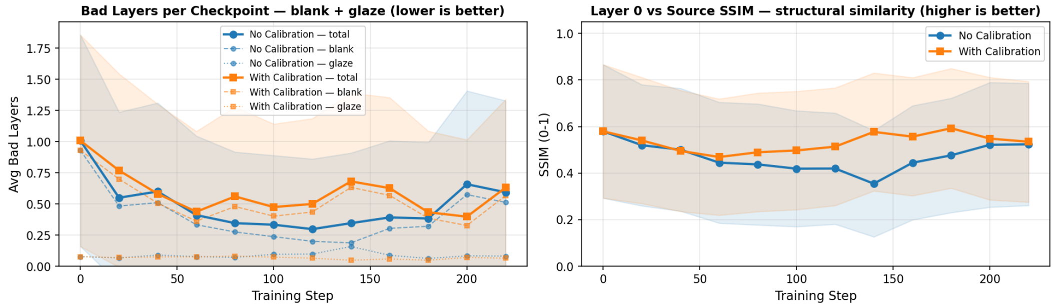

Figure 7: **Calibration ablation (additional Layer 0 metrics).** Layer 0 combined quality (left) and edge-density sharpness (right) over training, comparing the full two-phase reward with grid calibration against Phase 1 individual scoring alone. Both metrics show the calibrated run maintaining a small but consistent lead from approximately step 120 onward, corroborating the SSIM result reported in Table 4.

Ratio normalisation and gradient reweighting. We adopt the GRPO-Guard [26] stabilisation scheme (Section 3) with one modification. GRPO-Guard's default RatioNorm computes a spatial mean of per-element log-probabilities scaled by the noise standard deviation. For Qwen-Image-Layered, this suppresses the magnitudes of the log-ratio to near zero: the model packs multiple RGBA layers into a single latent sequence at 640×640 resolution, producing a dimensionality per-step D considerably higher than the single-image 512×512 latents of SD3.5 on which GRPO-Guard was developed. We therefore compute the log-probability per-step as a sum over spatial dimensions and normalise by $\sqrt{D}$ before applying GRPO-Guard's per-step centring (Algorithm 1, lines 15–16), preserving $\mathcal{O}(1)$ ratio magnitudes while retaining RatioNorm's centring and variance-stabilisation properties. The per-step policy loss is additionally scaled by $\delta = 1/\Delta t$ (gradient reweighting) to equalize gradient magnitudes across the noise schedule, following GRPO-Guard without modification.

Training and compute. $K{=}1$ PPO-style epoch per round, $G{=}16$, $\epsilon_c{=}0.2$, $\beta{=}10^{-3}$, AdamW with $\eta{=}10^{-5}$, advantage clip $c_{\text{adv}}{=}5.0$, gradient clip $\|\nabla\|_{\max}{=}1.0$. The main 600-step run was trained on $8\times$ NVIDIA H200 GPUs in $\sim$48 hours.

G Data Preprocessing

The primary source is Fine-T2I [20], an aesthetically filtered subset of photographs and artworks. All images are resized to 640×640 and normalised to $[-1, 1]$; associated captions serve as text prompts for the model's text conditioning. Images are shuffled at each epoch. The number of output layers per training step is sampled uniformly from [min_layers, max_layers] (typically $[2, 5]$), exposing the model to variable decomposition complexity.

H LayerD Comparison Setup

We compare against LayerD [24] on the held-out LAION-Aesthetics set used in Figure 4. LayerD's output count is variable and frequently smaller than the four layers our metrics score—the method tends to leave the input largely intact rather than separating it, sometimes returning a single layer containing the full image. To make the metrics computable on a fixed slot count, we pad LayerD's outputs with empty (white) layers up to four. We treat empty slots literally: for downstream editing purposes, an unfilled slot is an empty slot, and the resulting scores characterise the difference in decomposition strategy rather than quality on a shared axis (see Section 6.3).

I Crello Evaluation Metric

Yin et al. [35] measure per-layer reconstruction quality on the Crello dataset [32] by comparing each predicted layer against the corresponding ground-truth layer at the same index, using the LayerD [24] evaluation protocol with order-aware Dynamic Time Warping. Since our RL reward signal may

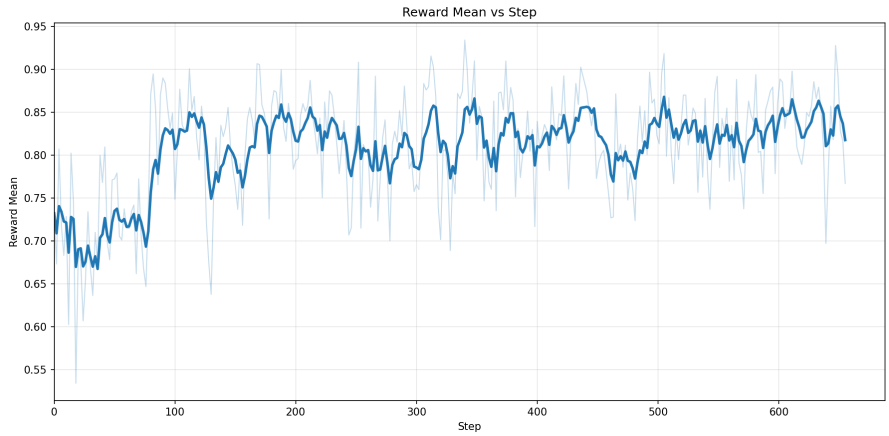

Figure 8: **Mean VLM reward during training.** Phase 2 calibrated reward per step (light) and rolling average (dark). The mean reward rises over the first $\sim$100 steps as the policy eliminates the worst failure modes, then plateaus around 0.83–0.85 with high per-step variance. Under GRPO's within-group normalisation (Equation (3)), learning requires only relative discrimination among group members, not rising absolute scores; the residual variance reflects conditioning-image difficulty rather than policy quality.

encourage the model to reorder layers relative to the base model's conventions (e.g., placing the most prominent foreground element on layer 1 rather than layer 2), a fixed-index comparison would penalise semantically correct decompositions that simply assign layers in a different order. We therefore modify the metric to use *best-match* assignment: for each reference layer, we select the predicted layer with the highest RGB similarity and compute the reconstruction error against that match, rather than relying on positional correspondence. This isolates decomposition quality from layer ordering, ensuring that improvements in semantic separation and content distribution are not masked by index-level misalignment introduced by the reward signal.

Held-out evaluation set. We evaluate on a fixed set of 480 images sampled from LAION-Aesthetics [22] and held constant across all checkpoints and ablations. The evaluation set is fully disjoint from the training data, which is drawn from a separate dataset (described in Section 4.3); no LAION-Aesthetics images appear in training.

J Text Conditioning Ablation Prompts

The text conditioning ablation in Section 6.4 compares two fixed prompts applied uniformly across all training images, replacing the per-image dataset captions used in the main run. The exact prompt strings are:

Basic prompt.
```
a clean, well composed image.
```
Detailed prompt.
```
a high quality image with multiple distinct objects clearly separated
from a clean background, sharp edges, vivid colors, balanced lighting,
well-defined foreground elements against a coherent backdrop,
professional composition with clear depth layers.
```

The detailed prompt was chosen to mirror the reward rubric's evaluation axes (object separation, alpha cleanliness, background coherence, feature distribution), testing whether prompt-rubric alignment helps or hinders training. As reported in Section 6.4, the detailed prompt fails to reduce bad layers despite this surface alignment.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We clearly stated that investigated RL fine-tuning an editing model to improve the performance and demonstrated it with thorough ablations (Sections 6.4 and 6.5) and comparisons (Section 6.2).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: They are clearly stated in Section 7.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We apply existing concepts in novel ways and for novel tasks.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The method, architecture, hyperparameters, and reward prompts are described in sufficient detail for reproduction in Section 4 and appendices B and F, and the base model and dataset are publicly available. We plan to release the trained LoRA adapters.

   Guidelines:

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

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

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

   Answer: [Yes]

   Justification: Code release is under consideration. The reward prompts (Appendix B), training procedure (Appendix F), and dataset and base model (Sections 4.1 and 4.3) are documented in full to enable reproduction without code release.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provided detailed documentation of our setup in the method section Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: All evaluation plots include shaded error bands representing ± 1 standard deviation (1σ) computed across all test images within each checkpoint variant. For example, a checkpoint evaluated on 480 images reports the mean metric value with the standard deviation of that metric across the 480 per-image scores. The error bands capture variability due to input image diversity (different SVG complexities, content types, and colour distributions) under fixed model weights and sampling parameters. Standard deviations are computed directly. We do not assume normally distributed errors; the bands are shown as symmetric $\pm 1\sigma$ for visual clarity, but we note that metrics bounded in $[0, 1]$ may have asymmetric tails near the boundaries.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The main run used 8$\times$ NVIDIA H200 GPUs for approximately 48 hours (600 steps). Ablation runs used the same hardware with fewer steps. Total compute including preliminary experiments exceeded the reported runs by approximately 3$\times$. Details are provided in Section 5.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We conform with the Code of Ethics fully.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: Stable-Layers improves the quality of automated layer decomposition, which lowers the technical barrier for image compositing and editing. Beneficial uses include accessibility tooling, education, and creative workflows. The same capability could marginally ease the production of misleading composite imagery, though the model operates on existing images rather than synthesising them and does not provide capabilities beyond those of existing editing tools. The reward signal incorporates a penalty on unsafe content during training, which is expected to reduce the prevalence of such outputs relative to the base model.

Guidelines:

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

11. **Safeguards**

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

Answer: [Yes]

Justification: The model performs decomposition of user-supplied images rather than open-ended synthesis, which limits the misuse surface relative to general-purpose image generators. The reward signal additionally penalises unsafe content during training. Release plans for the trained adapters have not yet been finalised; any release would be accompanied by appropriate safeguards.

Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We discuss our data in Section 4.3 and base model use in Section 4.

    Guidelines:

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

13. **New assets**

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

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification:

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification:

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: We use gemini-3-flash-preview as the VLM reward model, which is a core component of our training pipeline providing the sole source of supervision. Its role is described in Section 4.2 and the full scoring prompts are provided in Appendix B.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.