FLOWSAPO: SEMANTICS-AWARE REGIONAL OPTIMIZATION FOR VISUAL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Online reinforcement learning aligns diffusion and flow models using rewards on complete images or videos, while optimizing denoising transitions over a latent grid. A global clipping statistic can hide where likelihood changes occur, whereas independent positionwise clipping does not explicitly coordinate related positions. We introduce FlowSAPO, which combines positionwise clipping of exact local likelihood ratios with a regional objective. During rollout, we use the model's own text attention to group nearby positions with similar attention patterns into connected, overlapping regions. A size limit that varies with the noise level controls their growth, and the regions remain fixed during replay. Each region combines its members' local log-ratios into one weight for a shared clipping decision within that regional term. The terminal reward and scalar advantage remain unchanged. Under a fixed-state shared-covariance Gaussian analysis, we show that global aggregation loses spatial resolution, while regional grouping preserves the positionwise gradient direction up to scale at rollout and introduces grouping-dependent interactions during replay. Across the reported image and video comparisons, FlowSAPO improves over the local objective, including GenEval attribute binding from 0.86 to 0.96 and All Frame OCR on Wan2.1 from 0.339 to 0.427. It also raises VideoReward visual quality from 2.302 to 3.901 and motion quality from 0.811 to 1.596 compared with FlowGRPO. These results support regional optimization as a useful complement to positionwise clipping for visual generation.

1 INTRODUCTION

Online reinforcement learning fine-tunes diffusion and flow models with rewards evaluated on complete images or videos (Black et al., 2024; Fan et al., 2023; Zhang et al., 2024; Liu et al., 2025a; Xue et al., 2025b). Each sampled denoising transition carries a weight comparing the current policy with the policy that produced it, and PPO-style clipping stops the gradient of that transition once the weight has left a fixed band in the direction its advantage favours (Schulman et al., 2017). A transition, however, spans a latent grid whose positions carry different objects, attributes, boundaries, and background, and one update can move them in different directions. This raises a question that the sample-level reward does not answer and that current objectives settle in advance, without regard to what is being generated: which positions should share a clipping decision?

A common design in visual GRPO treats an entire denoising transition as a single unit for clipping. All positions share one decision, regardless of their relationships in the generated content. Changes in one part of the scene can therefore suppress surrogate contributions from another, even when the two are unrelated. The limitation is not merely coarse resolution: the scope of coordination is fixed by the latent grid rather than its semantic structure. A shared reward need not imply a shared clipping decision.

A natural alternative is to clip each latent position separately. Under the diagonal-Gaussian transitions considered here, exact local likelihood ratios yield the positionwise, or 1×1, objective. This retains local information and improves PickScore and OCR in our experiments. However, sampler factorization does not imply that visual structures decompose at the same resolution: related positions are still evaluated separately. ViPO (Ni et al., 2026) introduces perceptual structure by reweighting positionwise advantages, but retains independent local clipping. STAR (Shen et al., 2026) uses rollout attention for advantage allocation and local gradient routing, while retaining a common forward

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

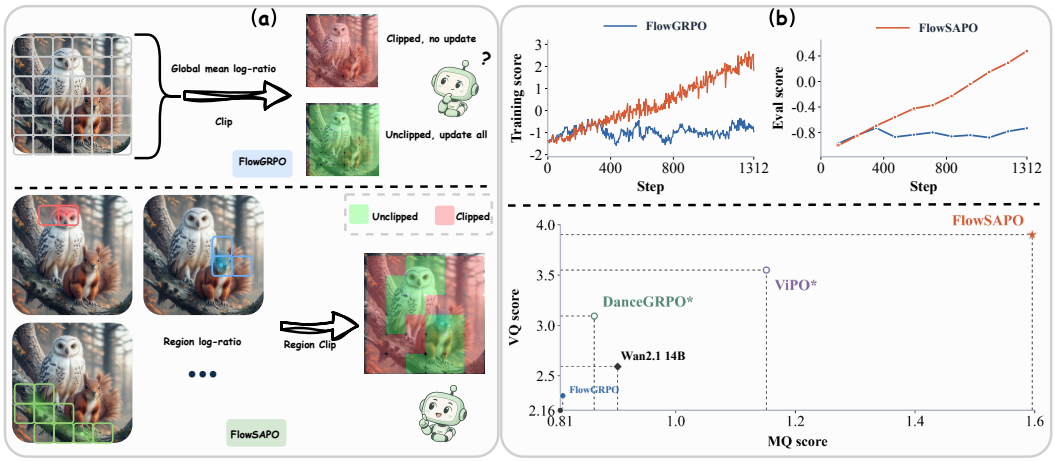

Figure 1: **(a) Global and regional clipping; (b) video results.** (a) Top: FlowGRPO reduces the latent-grid log-ratio changes to one mean-log weight and makes a single clipping decision. Bottom: FlowSAPO aggregates exact local log ratios within attention-derived regions, so different regions retain (green) or suppress (red) their updates. (b) Wan2.1-T2V 1.3B under VideoReward: training and evaluation scores against the optimization step (top), and visual quality (VQ) against motion quality (MQ) on our VidProM subset (bottom), using the inference resolution, frame count, and sampling step count reported by ViPO; the unlabeled point at the origin is the untrained Wan2.1, and the DanceGRPO and ViPO points, marked by asterisks, are literature reference values.

weight for clipping. These methods change the spatial emphasis of the learning signal; we instead ask which local changes should enter the same regional surrogate.

We introduce FlowSAPO, which measures policy change with exact local likelihood ratios and groups related positions for regional optimization. Alongside the 1×1 objective, it adds a regional objective that combines the local likelihood changes within each region into a common weight for clipping. Positions in the same region therefore share a clipping decision within that regional term. Regions can overlap, allowing a position to contribute to several regional terms while retaining its local term. The terminal reward and scalar group advantage remain unchanged. Region construction requires an association signal that reflects which latent positions should be considered together under the current generation condition. External visual features, such as DINO (Caron et al., 2021) or ResNet (He et al., 2016) representations, and segmentation masks from SAM (Kirillov et al., 2023) could provide alternative grouping cues. However, in their standard form, they do not directly represent how the current generator associates latent positions with the prompt at a particular denoising state and timestep, and using them would require selecting an image estimate and mapping the resulting features or masks back to the latent grid. We therefore use text attention recorded from the conditional rollout branch, which is already tied to the sampled state, timestep, prompt tokens, and the generator's latent representation. This choice avoids a separate external perception model while providing a prompt-conditioned association signal for regional optimization. We do not assume that these regions correspond to complete object masks; their role is to determine which local policy changes enter the same regional surrogate. Alternative grouping signals are discussed further in Appendix I.

FlowSAPO attains the highest reported optimized score among the compared methods in each image setting and in video text rendering, and its gains are largest where the prompt ties content to specific parts of the output. On both image backbones, adding the regional branch to the 1×1 objective improves all six reported preference metrics. On GenEval, FlowSAPO reaches 0.97 Overall against 0.95 for FlowGRPO and 0.94 for the 1×1 objective, and attribute binding rises from 0.89 with FlowGRPO to 0.96, where the 1×1 objective reaches 0.86. In video text rendering on Wan2.1-T2V 1.3B, All-Frame OCR rises from 0.246 with FlowGRPO to 0.339 with the 1×1 objective and 0.427 with FlowSAPO. Under VideoReward, the training and evaluation scores of FlowGRPO stay flat while those of FlowSAPO rise throughout training (Figure 1b). On our VidProM subset, FlowSAPO improves both visual and motion quality over FlowGRPO, which stays close to the pretrained model.

On VBench, which is not used for training, FlowSAPO scores higher than FlowGRPO on seven of the eight reported dimensions.

Our contributions are threefold.

- We identify the unit of the clipping decision as a choice that visual GRPO fixes in advance, and show, for Gaussian transitions at a fixed state, that the global mean-log statistic sees the spatial pattern of policy change only through its total and concentrates as the grid grows.
- We introduce FlowSAPO, whose semantics-aware regional clipping lets positions that attend alike to the prompt share one clipping decision while exact positionwise ratios measure policy change, and prove that the stored cover preserves the surrogate gradient direction at the rollout policy up to scale, while affecting both regional weights and shared clipping during replay.
- On two image backbones and one video backbone, FlowSAPO attains the highest reported optimized score among the compared methods in each image setting and in video text rendering, with the largest gains in attribute binding and video text rendering, and in every setting where the 1×1 objective is run, the regional branch improves on it.

2 PRELIMINARIES

Denoising transitions as actions. Let $\tau = (z_T, \dots, z_0)$ be a reverse denoising trajectory conditioned on text c, where $z_t \in \mathbb{R}^{C\times N_f}$ holds C latent channels over the grid Ω_f with $N_f = |\Omega_f|$ positions. For video, a position also carries its frame index. Each step is one action: the state is $s_t = (z_t, t, c)$, the action is $a_t = z_{t-1}$, and the stochastic sampler assigns it the transition density

$$\pi_\theta(a_t \mid s_t) = \mathcal{N}\big(a_t; \mu_{\theta,t}(s_t), \sigma_{\mathrm{tr},t}^2 I_{CN_f}\big)\,. \tag{1}$$

The current and rollout policies share the diagonal covariance, with per-coordinate variance $\sigma_{\mathrm{tr},t}^2 > 0$. The density therefore factorizes over coordinates; this is a property of the sampler, not an assumption about the network, data, or reward. During optimization, stored transitions are reused for updates, which we call *replay*: the state, action, rollout-policy mean, and variance stay fixed, and only the current-policy mean is recomputed. For sample i and coordinate $d = (k, p)$, channel k at position p, the change in log density is

$$\Delta\ell_{i,t,d} = \log \pi_{\theta,t,d}(a_{i,t,d} \mid s_{i,t}) - \log \pi_{\mathrm{old},t,d}(a_{i,t,d} \mid s_{i,t}). \tag{2}$$

Because the density is a product over coordinates, these changes add up to the log of the joint transition ratio,

$$r_{i,t}^{\mathrm{joint}} = \frac{\pi_\theta(a_{i,t} \mid s_{i,t})}{\pi_{\mathrm{old}}(a_{i,t} \mid s_{i,t})} = \exp\left(\sum_d \Delta\ell_{i,t,d}\right). \tag{3}$$

This ratio equals one at the rollout policy and changes as the current policy changes the density assigned to the stored action.

Group advantage and PPO clipping. Sample i receives one scalar advantage from the terminal rewards of its rollout pool $\mathcal{P}$. Its reward R_i is centered on the mean m_{c_i} over samples sharing its prompt and scaled by the pool standard deviation:

$$A_i = \frac{R_i - m_{c_i}}{\mathrm{Std}_{i'\in\mathcal{P}}(R_{i'})}. \tag{4}$$

Numerical safeguards are listed in Appendix B.1. For a positive policy weight r, the signed PPO loss is

$$\ell_\epsilon(r, A) = \max(-Ar, -A\,\mathrm{clip}(r, 1-\epsilon, 1+\epsilon))\,. \tag{5}$$

Clipping stops a term from encouraging further change once its weight has moved far enough in the direction favored by the advantage: for $A > 0$, when $r > 1+\epsilon$; for $A < 0$, when $r < 1-\epsilon$. Standard PPO uses $r = r_{i,t}^{\mathrm{joint}}$. Here we compare weights formed over the whole grid, individual positions, or regions, using the same loss and scalar advantage.

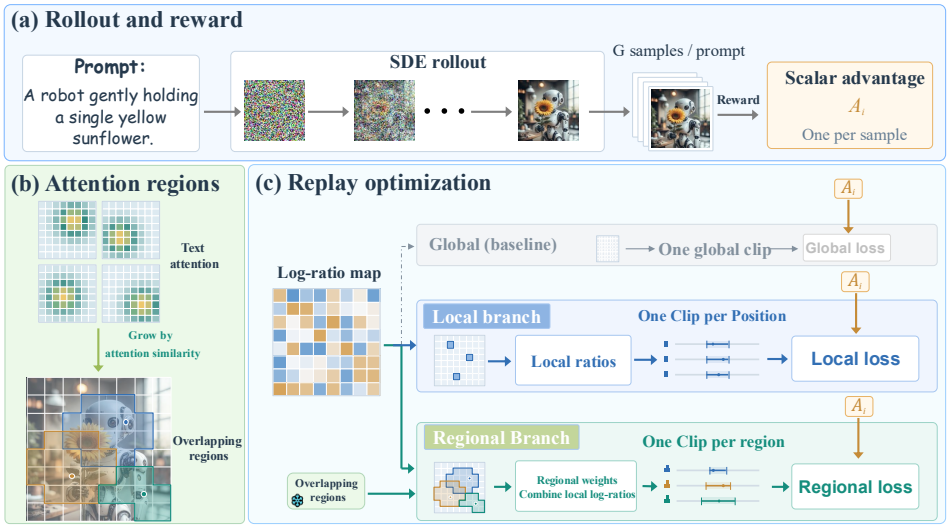

Figure 2: **FlowSAPO training framework.** Rewards from rollout samples produce scalar advantages. The local branch sums coordinate log ratios over the channels of each latent position. The regional branch sums these local log ratios over attention cells, weights each cell by inverse coverage, and exponentiates the resulting regional score. Rollout text attention determines the stored cover used by this branch. The two clipped objectives are combined with weights $1-\lambda$ and λ; attention does not rescale the scalar advantage. Equations (8), (10), and (14) specify these operations.

3 Method

FlowSAPO measures policy change at each latent position and lets related positions contribute through an additional regional objective. The local branch evaluates each position separately. The regional branch combines the changes within each group into one weight and applies a shared clipping decision to that term. We construct the groups from rollout text attention and keep them fixed during replay. The sampler, terminal reward, and scalar advantage remain unchanged. Figure 2 summarizes the method.

3.1 Local Measurement of Policy Change

A single clipping weight for the whole transition can hide changes at individual positions. The global mean-log statistic considered here is

$$r_{i,t}^{\mathrm{geo}} = \exp\left(\frac{1}{CN_f}\sum_{d}\Delta\ell_{i,t,d}\right) = \left(r_{i,t}^{\mathrm{joint}}\right)^{1/(CN_f)}. \tag{6}$$

For example, an increase at one position can cancel a decrease at another. Suppressing i,t, let $p_1 \neq p_2$ and

$$\Delta\ell_{(k_1,p_1)} = \alpha, \qquad \Delta\ell_{(k_2,p_2)} = -\alpha, \qquad \Delta\ell_d = 0 \text{ otherwise.} \tag{7}$$

Then $r^{\mathrm{geo}} = 1$ for any α, just as if no change had occurred. Appendix H.1 characterizes this loss of spatial information for Gaussian transitions at a fixed stored state.

We instead retain the likelihood change at each latent position by summing only over its C channels:

$$\delta_{i,t,p} = \sum_{k=1}^{C}\Delta\ell_{i,t,(k,p)}, \qquad r_{i,t,p}^{\mathrm{fine}} = e^{\delta_{i,t,p}} = \frac{\pi_{\theta,t,p}(a_{i,t,p}\mid s_{i,t})}{\pi_{\mathrm{old},t,p}(a_{i,t,p}\mid s_{i,t})}. \tag{8}$$

Here $\pi_{\theta,t,p}$ is the density of the channel vector at p, conditioned on the full stored state. The sampler's diagonal covariance makes this an exact local density ratio. Applying PPO clipping separately to these ratios gives

$$\mathcal{L}_{\mathrm{fine}} = \frac{1}{BTN_f}\sum_{i,t}\sum_{p\in\Omega_f}\ell_{\epsilon_f}\left(r_{i,t,p}^{\mathrm{fine}}, A_i\right), \tag{9}$$

where B is the batch size and ϵ_f is the local clipping width. This is the 1×1 objective used in our comparisons and retained as the local branch of FlowSAPO.

Each position now has its own clipping decision, but positions associated with the same visual structure still enter separate terms. We retain this local objective and add regional terms that consider related changes together. The next subsection describes how these regions are constructed.

3.2 Semantics-Aware Region Construction

Region construction must decide which nearby positions are related and how many should enter a shared term. We use text attention to guide the first decision and a size limit to control the second. Regions are built on the attention grid Ω_a, with $N_a = |\Omega_a|$ cells. Each cell q corresponds to a block $B(q)$ of b latent positions, determined by the backbone's patch size, so $N_f = bN_a$.

Attention at the sampled state. At each sampled denoising step, we record attention from visual tokens to prompt tokens in the conditional rollout branch. We average the selected heads and layers and set padded prompt entries to zero without renormalizing. The resulting vector $f_{i,t}(q)$ describes how cell q attends to the prompt. It comes from the same latent state, prompt, and denoising step as the stored transition, without an external perception model. We compare these vectors by cosine similarity, using the pattern of attention over prompt tokens rather than its total magnitude. Appendix J.3 details attention extraction for the image and video backbones.

Growing connected regions. For one transition, we omit i, t for clarity. Every cell j starts a region $R_j = \{j\}$ and remains its reference, or *seed*. We measure each candidate's similarity to that seed as $\kappa_j(q) = \cos(f(j), f(q))$. Using the same reference throughout growth ensures that each candidate is compared with the original attention pattern, not just with the most recently added cell.

A candidate $q \notin R_j$ must be one of the eight neighbors of an accepted cell $v \in R_j$. Let $W_3(v)$ be the valid cells in the 3×3 window centered at v. We compare the candidate with the average seed similarity over the current region and this window:

$$\vartheta_j(v; R_j) = \frac{1}{|R_j \cup W_3(v)|} \sum_{x \in R_j \cup W_3(v)} \kappa_j(x).$$

The candidate q is eligible through v when $\kappa_j(q) \geq \vartheta_j(v; R_j)$. Among eligible candidates, we add the one with the highest seed similarity and recompute the thresholds. Thus, growth depends on the region and its neighborhood rather than a fixed similarity cutoff. Appendix I specifies tie breaking and gives the full algorithm. For video, regions are grown independently within each latent frame.

Controlling region size. Growth stops when no candidate is eligible or the region reaches its maximum size $S_{i,t}^{\max}$. We vary this limit with the normalized noise level $\rho_{i,t} \in [0, 1]$:

$$S_{i,t}^{\max} = \mathrm{round}(S_{\mathrm{noisy}}\rho_{i,t} + S_{\mathrm{clean}}(1 - \rho_{i,t})) .$$

We use a smaller limit at high noise and allow larger regions as noise decreases. This is an upper bound, not a target size: the similarity rule can stop growth earlier. A region can therefore cover part of an object without having to include the whole object. Since attention also changes across steps, the actual regions need not grow monotonically. Appendix I defines the noise normalization, and Appendix B.1 lists the size limits.

Storing overlapping regions. Let $R_{i,t,j}$ denote the region seeded at cell j, and let $n_{i,t}(q)$ count the regions containing cell q. Seeding every cell ensures that the regions cover the grid and that $n_{i,t}(q) \geq 1$. Regions may overlap. We store their membership and coverage counts with each transition and keep them fixed during replay, with no gradients through attention capture or region construction.

3.3 Regional Policy Optimization

Grouping positions would not change their clipping decisions if their losses were still evaluated separately. We therefore form one weight from the local changes in each region and apply PPO clipping to that weight. The regional terms are added alongside the local objective, so each position can contribute both individually and through its regions.

Regional weights and overlap. We first sum the local log-ratios within an attention cell, $u_{i,t}(q) = \sum_{p \in B(q)} \delta_{i,t,p}$. Because regions overlap, adding this full score to every containing region would count it repeatedly. We instead divide it by the number of regions containing the cell:

$$s_{i,t,j} = \sum_{q \in R_{i,t,j}} \frac{u_{i,t}(q)}{n_{i,t}(q)}, \qquad \bar{w}_{i,t,j} = e^{s_{i,t,j}}. \tag{10}$$

For example, a cell belonging to three regions contributes one third of its score to each. Across all regions, each cell's log score is counted once:

$$\sum_{j \in \Omega_a} s_{i,t,j} = \sum_{q \in \Omega_a} u_{i,t}(q) = \sum_{p \in \Omega_f} \delta_{i,t,p} = \log r_{i,t}^{\text{joint}}. \tag{11}$$

At $\theta = \theta_{\text{old}}$, this division also makes the regional loss gradient proportional to the local one, regardless of the grouping. The scale difference comes from averaging over regions rather than positions (Appendix H.2). Because shared cells contribute fractional powers of their local ratios, a regional weight is generally not a density ratio. Opposing changes can still cancel within a region, but only among its members.

Clipping and the combined objective. Each regional weight enters one PPO term, giving its members a shared clipping decision within that term. We set the clipping width according to the number of latent positions it covers, $a_{i,t,j} = b|R_{i,t,j}|$:

$$\epsilon_{i,t,j} = \epsilon_f a_{i,t,j}^{\nu}. \tag{12}$$

The exponent ν controls how the clipping width increases with region size. The analysis in Appendix H.2 motivates considering size, but does not determine ν. When $\epsilon_{i,t,j} \geq 1$, the lower boundary is nonpositive and cannot be reached by the positive regional weight. The local branch retains its own clipping rule.

The regional loss averages one term per seeded region. We combine it with the local loss as

$$\mathcal{L}_{\text{region}} = \frac{1}{BTN_a} \sum_{i,t} \sum_{j \in \Omega_a} \ell_{\epsilon_{i,t,j}}(\bar{w}_{i,t,j}, A_i), \tag{13}$$

$$\mathcal{L}_{\text{policy}} = (1-\lambda)\mathcal{L}_{\text{fine}} + \lambda \mathcal{L}_{\text{region}}. \tag{14}$$

Here λ weights the regional loss. Both branches use the same A_i; attention does not assign rewards or advantages to individual positions. Appendix B.1 lists the clipping parameters, size limits, and loss coefficient.

Under the combined objective, each position contributes through its local term and the regional terms containing it. Within a region, each member's contribution depends on the other members through their common weight, even before clipping occurs. When that regional term is clipped, the position may still contribute through other regions and its local term. Appendix H.2 gives the gradient analysis.

During replay, we recompute the current policy's transition mean and evaluate both losses on the stored actions and regions. Log-ratios are computed in FP32, and both exponents are bounded to $[-20, 20]$ before exponentiation. Exact local density-ratio statements apply before this bound. Appendix J gives the reference regularization and complete training procedure.

4 Experiments

We conduct experiments on image preference alignment, compositional generation, and video generation. For each task, we assess FlowSAPO through quantitative comparisons and qualitative analysis. We further examine the contributions of the local objective and regional branch through ablation studies. More experimental details are provided in Appendix B.

4.1 Experimental setup

Tasks and models. We use SD3.5 Medium (Esser et al., 2024; Stability AI, 2024) and FLUX.1 [dev] (Black Forest Labs, 2024) for image generation, optimizing PickScore for preference alignment

Table 1: PickScore optimization on FLUX.1 [dev] (top) and SD3.5 Medium (bottom). Best and second best scores among aligned models are in blue and green. Higher is better. Training and checkpoint selection are described in Appendix B.4.

Method	PickScore	CLIPScore	HPSv2.1	Aesthetic	ImageReward	UnifiedReward
FLUX.1 [dev]						
Base model	22.848	0.275	0.295	5.723	0.953	3.325
FlowGRPO	23.909	0.271	0.325	6.006	1.297	3.495
DiffusionNFT	23.998	0.276	0.313	5.869	1.295	3.568
FlowSAPO(1×1)	24.078	0.274	0.328	5.978	1.380	3.522
FlowSAPO	24.155	0.278	0.330	6.090	1.396	3.578
SD3.5 Medium						
Base model	22.345	0.284	0.279	5.383	0.847	3.094
FlowGRPO	23.438	0.280	0.304	6.024	1.310	3.338
DiffusionNFT	23.713	0.269	0.315	6.150	1.249	3.423
FlowSAPO(1×1)	23.561	0.272	0.294	6.038	1.277	3.220
FlowSAPO	23.828	0.275	0.315	6.084	1.354	3.382

and GenEval (Ghosh et al., 2023) on SD3.5 Medium for compositional generation. Video experiments use Wan2.1 with 1.3B parameters (Wan et al., 2025). Following ViPO, we jointly optimize the visual quality (VQ) and motion quality (MQ) components of VideoReward (Liu et al., 2025b). OCR reward is optimized separately for video text rendering. Each reward setting is trained independently with LoRA (Hu et al., 2021).

Evaluation and baselines. Following DiffusionNFT (Zheng et al., 2026), we evaluate image preferences on DrawBench (Saharia et al., 2022), reporting PickScore and five independent evaluators: CLIPScore, HPSv2.1, aesthetic score, ImageReward, and UnifiedReward. GenEval provides an overall score and six categories. Video evaluation uses the standard VBench 1.0 prompt suite (Huang et al., 2024) and a separate 1,000 prompt subset of VidProM (Wang & Yang, 2024); OCR evaluation measures recognition and frame exact matching. We compare with the pretrained backbones, FlowGRPO, and the 1×1 variant, adding DiffusionNFT for images. Published ViPO and DanceGRPO scores in Figure 1 provide literature reference points. Appendix B collects the training data, configuration, evaluation protocols, metric definitions, and checkpoint selection rules.

4.2 Image Experiments

FLUX.1 [dev]. FlowSAPO achieves the highest scores on all six metrics in Table 1 (top). Agreement across the five evaluators not used for training indicates that the gains extend beyond PickScore to semantic alignment, aesthetics, and learned human preferences.

Figure 3 illustrates the balance between visual preference and prompt adherence. For "A banana on the left of an apple," the other aligned models produce multiple bananas, whereas FlowSAPO preserves the requested count and spatial relation with detailed textures and natural lighting. The red dog and underwater palace examples likewise preserve the requested color and coloring page style. Together with the quantitative results, these observations show that FlowSAPO combines visual appeal with instruction adherence in these examples. Further comparisons appear in Appendix E.

SD3.5 Medium. To test generality on a lightweight backbone, we also evaluate FlowSAPO on SD3.5 Medium. In Table 1 (bottom), it achieves the highest PickScore and ranks first or second on every reported metric among aligned models, supporting the effectiveness of regional optimization across backbones.

Compositional generation. In Table 2, FlowSAPO achieves the highest overall score, with the clearest advantage in attribute binding. This is consistent with a regional objective that groups positions with similar prompt attention into a shared surrogate term. Together with the prompt adher-

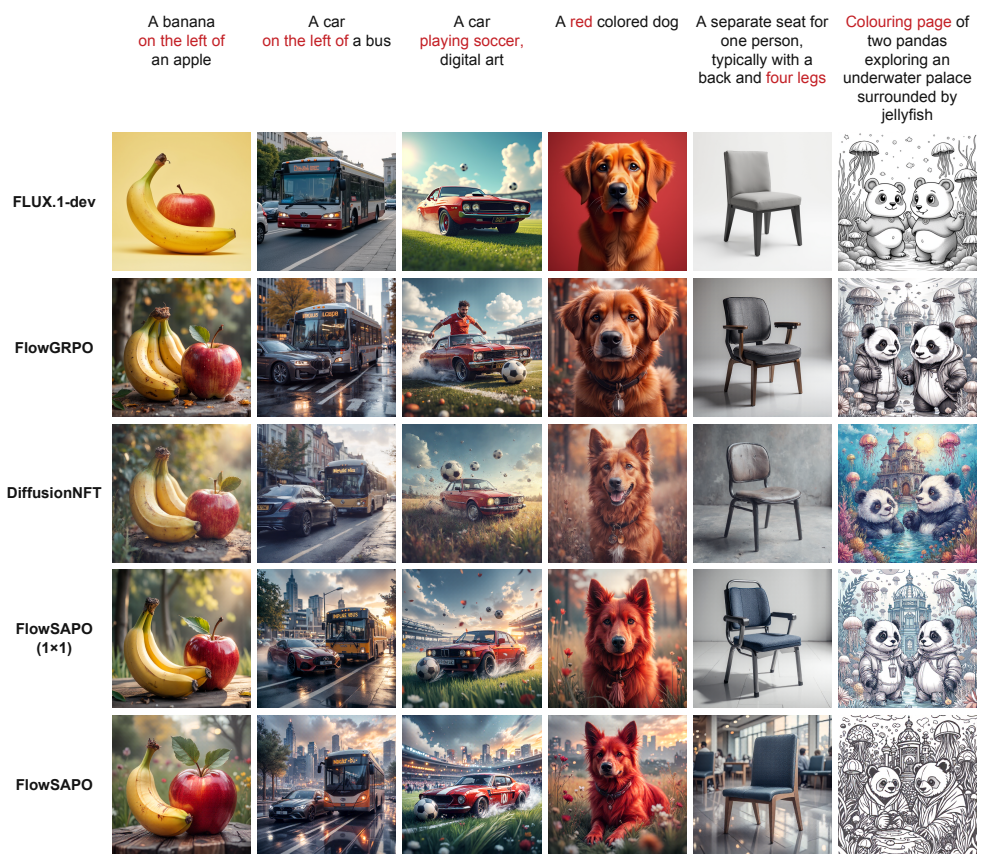

Figure 3: **Qualitative comparison on FLUX.1-dev.** Each column uses the prompt shown above it. Rows show the base model, FlowGRPO, DiffusionNFT, FlowSAPO (1×1), and FlowSAPO.

Table 2: GenEval optimization on SD3.5 Medium. Blue and green indicate the best and second-best scores among aligned models at the displayed precision. Higher is better.

Dimension	Base model	FlowGRPO	DiffusionNFT	FlowSAPO (1×1)	FlowSAPO
Overall	0.65	0.95	0.96	0.94	0.97
Single Object	0.97	1.00	1.00	1.00	1.00
Two Objects	0.82	0.99	1.00	1.00	0.99
Counting	0.53	0.95	0.98	0.95	0.98
Colors	0.82	0.92	0.94	0.93	0.94
Position	0.22	0.99	0.99	0.90	0.97
Attr. Binding	0.55	0.89	0.87	0.86	0.96

Table 3: VBench after VQ and MQ optimization on Wan2.1 (1.3B). Blue indicates the higher unrounded score among aligned models.

Dimension	Wan2.1	FlowGRPO	FlowSAPO
Subject Cons.	0.954	0.970	0.982
Background Cons.	0.984	0.970	0.973
Motion Smooth.	0.975	0.988	0.986
Aesthetic Qual.	0.619	0.639	0.651
Imaging Qual.	0.630	0.651	0.656
Object Class	0.772	0.741	0.749
Multiple Objects	0.536	0.597	0.612
Color	0.907	0.900	0.913

ence observed in the FLUX examples, this result supports regional optimization for compositional generation.

4.3 Video Experiments

Wan2.1 with VideoReward. Video generation additionally requires consistency across frames. Figure 1b shows that FlowSAPO continues to improve after FlowGRPO plateaus, with gains in both VQ and MQ on VidProM. With only 1.3B parameters, FlowSAPO also achieves higher VQ and MQ scores than those reported for the Wan2.1 14B baseline and the model optimized by ViPO, under

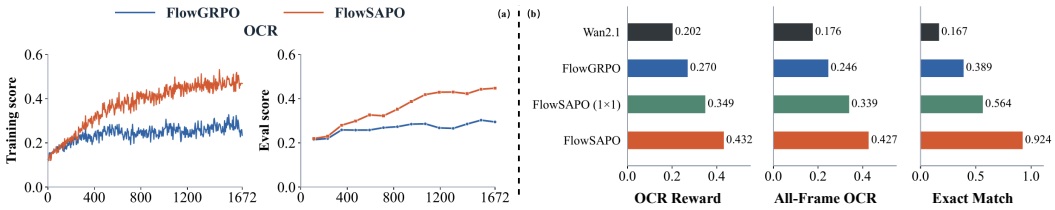

Figure 4: **OCR optimization on Wan2.1 (1.3B).** (a) Training and evaluation scores for FlowGRPO and FlowSAPO. (b) OCR Reward, All Frame OCR, and Frame Exact Match (%) for the base model, FlowGRPO, the 1×1 variant, and FlowSAPO. Appendix B.3 defines all three metrics. Higher is better.

the respective evaluation settings (Appendix B.4). The improvement also extends to independent VBench metrics, including subject consistency, imaging quality, and multiple objects (Table 3). Figures 13 and 14 show clearer object boundaries and more regular scene structure. In Figure 15, the handset remains defined as it is lowered while the subject and background retain their appearance, complementing the VBench consistency results.

Wan2.1 with OCR. Text rendering tests whether localized character structure can be preserved over time. Figure 4a shows sustained improvement beyond the FlowGRPO plateau, with a similar trend in evaluation. The higher All Frame OCR score in panel (b) shows that the gains persist when every sampled frame contributes, including frames with zero recognition reward.

4.4 Ablation Studies

FlowGRPO $\rightarrow$ FlowSAPO(1×1). The 1×1 variant replaces the global statistic with exact local ratios and omits the regional branch. This improves PickScore and All Frame OCR (Table 1; Figure 4), but does not consistently improve independent image metrics or compositional generation. Local measurement therefore benefits reward optimization without fully resolving alignment.

FlowSAPO$(1 \times 1) \rightarrow$ FlowSAPO. Adding the regional branch to the 1×1 objective improves all six preference metrics on both image backbones and strengthens attribute binding and video text recognition (Tables 1 and 2; Figure 4). Together with the gains over FlowGRPO, these results demonstrate the effectiveness of combining exact local ratios with regional optimization across image and video tasks.

Additional ablations of fixed patch sizes, objective weights, the region size limit, and training dynamics are provided in Appendix D.

5 Limitations and Conclusion

FlowSAPO combines exact positionwise likelihood ratios with regional surrogate weights built from rollout attention. The reward and scalar advantage remain unchanged, while a frozen cover determines which local log ratios enter each exponential weight and shared clipping decision. Across two image backbones and one video backbone, FlowSAPO improves on FlowGRPO in image preference alignment, compositional generation, video quality, and text rendering. The regional branch further improves over positionwise optimization alone, particularly in attribute binding and video text recognition. Together, these results demonstrate the effectiveness of combining exact local ratios with regional optimization for visual generation.

Limitations. Our experiments focus on generation conditioned on text, using a 1.3B backbone for video. Image to video generation, larger video models, longer sequences, and higher resolutions remain to be explored. Beyond the fixed scalar reward objectives used here, explicit multiobjective optimization could further examine tradeoffs among visual quality, motion quality, and prompt adherence.

AI USE STATEMENT

We used generative AI tools for LaTeX assistance, table formatting, and literature discovery for related work. We did not use generative AI tools to generate or alter data, run experiments, replace author decisions about the method, or make unsupported scientific claims. All AI-assisted text and code were reviewed by the authors. The authors designed the method and experiments, executed and tested the code, checked all numerical results against the recorded outputs, and manually verified the relevant literature. We take responsibility for the final content of this work, including any text, claims, code, or other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies optimization for generative image and video models and does not introduce a dataset containing personal or sensitive information. The method could strengthen both benign and harmful generation capabilities; any release should therefore retain the licenses, safety controls, and data documentation of its base models and evaluation resources. The comparisons in this paper use automated evaluators and qualitative examples.

REPRODUCIBILITY STATEMENT

The appendix gives the complete region construction (Appendix I), the implementation and training procedure (Appendix J), the experimental configuration and evaluation protocols (Appendix B), and the proofs of all formal results (Appendix K).

REFERENCES

Sergios Agapiou, Omiros Papaspiliopoulos, Daniel Sanz-Alonso, and Andrew M Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, pp. 405–431, 2017.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

Black Forest Labs. FLUX.1 [dev]. `https://github.com/black-forest-labs/flux`, 2024.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099–1135, 2018.

Kevin Clark, Paul Vicol, Kevin Swersky, and David Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, volume 2024, pp. 4793–4822, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

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. *Advances in neural information processing systems*, 36:79858–79885, 2023.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Seungyul Han and Youngchul Sung. Dimension-wise importance sampling weight clipping for sample-efficient reinforcement learning. In *International conference on machine learning*, pp. 2586–2595. PMLR, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. TempFlow-GRPO: When timing matters for GRPO in flow models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7mCo3R3Wyn.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

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 arXiv:2106.09685*, 2021.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_VBench_Comprehensive_Benchmark_Suite_for_Video_Generative_Models_CVPR_2024_paper.html.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

LAION-AI. LAION-Aesthetics Predictor V1. https://github.com/LAION-AI/aesthetic-predictor/tree/6d122adad522ab246644d9dc1c6d7a3810ee255f, 2022. Software artifact, revision 6d122ad.

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. *Advances in neural information processing systems*, 38:40783–40818, 2025a.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Wenyu Qin, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, Pengfei Wan, Di Zhang, Kun Gai, Yujiu Yang, and Wanli Ouyang. Improving video generation with human feedback. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/76227feb18ea0ee40bd15cf02c33e18e-Abstract-Conference.html.

Yifu Luo, Haoyuan Sun, Xinhao Hu, Penghui Du, Keyu Fan, Bo Li, Sinan Du, Xu Wan, Zhiyu Chen, Bo Xia, Yongzhe Chang, Kai Wu, Kun Gai, Tiantian Zhang, and Xueqian Wang. Principled RL for flow matching emerges from the chunk-level policy optimization. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://openreview.net/forum?id=tPd6GmXPbd.

Ziqi Ni, Yuanzhi Liang, Rui Li, Yi Zhou, Haibin Huang, Chi Zhang, and Xuelong Li. Seeing what matters: Visual preference policy optimization for visual generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27260–27269, 2026.

Bowen Ping, Xiangxin Zhou, Penghui Qi, Minnan Luo, Liefeng Bo, and Tianyu Pang. Flow-dppo: Divergence proximal policy optimization for flow matching models. *arXiv preprint arXiv:2606.11025*, 2026.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. Pmlr, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Jinjie Shen, Wei Deng, Xian Hu, Daiguo Zhou, and Jian Luan. Star: Spatiotemporal adaptive reward allocation for text-to-image rl post-training. *arXiv preprint arXiv:2606.17979*, 2026.

Stability AI. Stable diffusion 3.5 medium model card. `https://huggingface.co/stabilityai/stable-diffusion-3.5-medium`, 2024. Model artifact.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238. IEEE, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5988–5998, 2026.

Wenhao Wang and Yi Yang. Vidprom: A million-scale real prompt-gallery dataset for text-to-video diffusion models. *Advances in Neural Information Processing Systems*, 37:65618–65642, 2024.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

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 arXiv:2306.09341*, 2023.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11269–11277, 2026.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models. *arXiv preprint arXiv:2509.25050*, 2025a.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025b.

Yinan Zhang, Eric Tzeng, Yilun Du, and Dmitry Kislyuk. Large-scale reinforcement learning for diffusion models. In *European Conference on Computer Vision*, pp. 1–17. Springer, 2024.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL https://arxiv.org/abs/2507.18071.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, volume 2026, pp. 134129–134150, 2026.

Overview of the Appendix

The appendix begins with related work and experimental evidence, followed by the analysis and implementation details.

- **Appendix A** discusses policy optimization and spatial allocation for visual generation.
- **Appendix B** collects training configurations, evaluation protocols, and compared methods for Section 4.
- **Appendix C** evaluates text alignment as a separate video reward objective.
- **Appendix D** reports the parameter and training ablations.
- **Appendices E, F, and G** present additional image comparisons, adaptive region visualizations, and video comparisons.
- **Appendix H** analyzes global, positionwise, and regional clipping.
- **Appendix I** gives the region construction and its properties, and **Appendix J** details ratio computation, reference regularization, and training.
- **Appendix K** provides the proofs.

A Related Work

Diffusion and flow models are aligned with a reward by backpropagating through it, by offline preference optimization, or by advantage-weighted matching, none of which replays a transition ratio (Clark et al., 2024; Wallace et al., 2024; Xue et al., 2025a). FlowSAPO belongs instead to online policy optimization over sampled transitions (Black et al., 2024; Fan et al., 2023; Zhang et al., 2024; Liu et al., 2025a; Xue et al., 2025b), where a clipped importance ratio controls each update (Schulman et al., 2015; 2017), and its question is the unit over which that ratio is formed. Importance sampling over a complete high dimensional action can require a sample size that grows exponentially with policy divergence (Chatterjee & Diaconis, 2018; Agapiou et al., 2017), motivating smaller factors or normalized weights. Group sequence policy optimization clips the length-normalized ratio of a whole response, on the principle that the unit of optimization should match the unit of reward (Zheng et al., 2025); visual GRPO objectives clip the same statistic taken over the latent grid (Liu et al., 2025a; Xue et al., 2025b), and later variants keep that unit while regulating the clipping rule (Wang et al., 2026), reweighting timesteps (He et al., 2026), or merging consecutive steps into one action (Luo et al., 2026). A related approach controls updates with a divergence gate to reduce sensitivity to transition sampling noise (Ping et al., 2026). At the fine end, the weight of a factorized Gaussian policy can be clipped per action dimension (Han & Sung, 2019), and ViPO clips each latent position on its own (Ni et al., 2026). These approaches motivate examining how spatial factors are combined. FlowSAPO aggregates local log ratios into overlapping regional weights, each with its own forward value and clipping decision, while retaining a positionwise objective.

Spatial allocation methods modify where the learning signal acts. ViPO multiplies the scalar advantage by a perceptual map and clips local likelihood ratios (Ni et al., 2026). STAR derives allocation maps from text attention and uses a stop gradient construction: its forward ratio equals the global weight, while its derivative follows the local log probability (Shen et al., 2026). Thus, spatial gradient routing need not imply distinct forward weights. FlowSAPO keeps the advantage scalar and forms a separate exponential weight for each region from local log ratios. Text attention recorded during rollout (Hertz et al., 2022) supplies the grouping, which remains frozen during replay. This distinguishes regional aggregation from spatial advantage allocation without treating local factorization or the use of attention as new in themselves.

B Experimental Details

This appendix collects the complete experimental setup for Section 4: training data and configuration, image and video evaluation protocols, compared methods, checkpoint selection, and training cost.

B.1 TRAINING SETTINGS, REWARDS, AND METRICS

Tasks and training data. For image preference alignment, we train SD3.5 Medium and FLUX.1 [dev] with PickScore using prompts from its associated preference dataset (Kirstain et al., 2023). We separately train SD3.5 Medium with GenEval for compositional generation. Video preference alignment uses Wan2.1 with 1.3B parameters and prompts from VisionRewardDB (Xu et al., 2026). Following ViPO, we jointly optimize the VQ and MQ components of VideoReward. OCR reward is used in a separate video text rendering experiment. GenEval and OCR use the corresponding prompt splits released by FlowGRPO. Each reward setting is trained independently. Table 4 summarizes the optimized rewards and evaluation metrics; metrics not optimized during training provide independent checks.

Table 4: Settings of the reported experiments in Section 4.

Setting	Backbone	Optimized reward	Reported metrics	Ref.
Preference	SD3.5 Medium; FLUX	PickScore	PickScore; CLIPScore, HPSv2.1, aesthetic score, ImageReward, UnifiedReward	1
Composition	SD3.5 Medium	GenEval reward	GenEval Overall and its six categories	2
Video preference	Wan2.1-T2V 1.3B	VideoReward VQ and MQ	VQ, MQ; selected VBench quality and semantic dimensions	Fig. 1; 3
Video text rendering	Wan2.1-T2V 1.3B	OCR reward	OCR Reward, All Frame OCR, Frame Exact Match (%)	Fig. 4

Training configuration. All FlowSAPO runs use LoRA and AdamW with BF16 model computation and FP32 ratio arithmetic. SD3.5 Medium and FLUX use learning rate 3×10^{-4} and group size 24; Wan uses 10^{-4} and group size 16. The LoRA rank and scaling parameter are 32 and 64 for SD3.5 Medium and Wan, and 64 and 128 for FLUX. Image runs use eight NVIDIA H800 GPUs, and video runs use 16 ZW810E PPUs.

For SD3.5 Medium, we use 10 rollout steps, 40 evaluation steps, and guidance scale 4.5. FLUX uses 12 rollout steps, 28 evaluation steps, and guidance scale 3.5. Both image backbones train at 512×512 resolution; SD3.5 Medium is evaluated at this resolution, while final FLUX evaluation uses 1024×1024. Wan training uses 20 rollout steps, generating 33 frames at 240×416 resolution. Final video evaluation uses the sampling settings in Appendix B.3.

We set $(\lambda, 1-\lambda) = (0.25, 0.75)$ and $(S_{\text{noisy}}, S_{\text{clean}}) = (4, 64)$. The reference regularization coefficients are 0.01 for SD3.5 Medium with PickScore, 0.04 for SD3.5 Medium with GenEval, and 0.004 for Wan; FLUX uses no reference penalty. Table 5 lists the remaining configuration details.

Table 5: Configuration of the SD3.5 Medium PickScore runs.

Component	Setting	Value
Model	backbone / resolution	SD3.5 Medium / 512×512
Sampling	training steps / evaluation steps / guidance scale	10 / 40 / 4.5
Rollout	batch per device / samples per prompt / batches per epoch	6 / 24 / 24
Training	batch / gradient accumulation / replay epochs	6 / 3 / 1
Training	denoising steps trained per trajectory	9 of 10
Training	precision / ratio arithmetic	BF16 / FP32
Training	advantage cap / gradient-norm cap	5 / 1.0
Optimizer	type / learning rate / β_1, β_2	AdamW / 3×10^{-4} / 0.9, 0.999
Optimizer	weight decay / ε	10^{-4} / 10^{-8}
LoRA	rank / alpha / targets	32 / 64 / image and text attention q, k, v, out projections
EMA	decay / update interval	0.9 with warm-up / 8 optimizer steps
Reward	reward / centering / scale	PickScore / prompt mean / pool standard deviation $+10^{-4}$
Local branch	factor / clip width / guard	all channels of one latent position / 0.10 / $[-20, 20]$
Attention	layers / sampled steps	6, 12, 17 / all
Attention	reduction / padding / guidance branch	mean over heads, then layers / padded tokens zeroed / conditional only
Regions	seeds / connectivity / threshold	every cell / 8 / seed-relative, contextual (Appendix I)
Regional branch	coverage weighting / base width / area exponent ν	inverse coverage / 0.10 / 0.7
Reference regularizer	coefficient	0.01
FlowGRPO	statistic / clip width	global mean-log / 10^{-4}

B.2 Image evaluation

The preference results in Table 1 are evaluated on DrawBench (Saharia et al., 2022), following the prompt dataset choice of DiffusionNFT (Zheng et al., 2026). The training reward is PickScore. The other five metrics are used only for evaluation: CLIPScore (Hessel et al., 2021), HPSv2.1 (Wu et al., 2023), aesthetic score (LAION-AI, 2022), ImageReward (Xu et al., 2023), and UnifiedReward (Wang et al., 2025). The GenEval comparison in Table 2 uses the test prompts released with the FlowGRPO implementation and reports the overall score together with single object, two objects, counting, colors, position, and attribute binding scores.

B.3 Video evaluation

VBench. For Table 3, we use the standard VBench 1.0 suite of 944 unique prompts (Huang et al., 2024). Each model generates 33 frames at 8 FPS and 480×832 resolution, using 50 sampling steps, a flow shift of 3.0, and an empty negative prompt. We follow the VBench generation seed protocol and pair five seeds across models for each prompt. For temporal flickering, we generate 25 candidates per prompt and retain the first five qualifying static videos using the upstream filtering procedure. We evaluate the 16 VBench dimensions with the official implementation. Table 3 presents eight dimensions covering subject and background consistency, motion smoothness, visual quality, and object composition. Following the evaluation design of ViPO, we use VBench to assess performance beyond the optimized VQ and MQ rewards; the reported dimensions are listed in Table 3.

VidProM. We construct an evaluation subset of 1,000 VidProM prompts (Wang & Yang, 2024) after removing normalized exact matches with the public DanceGRPO video training prompts (Xue et al., 2025b) and our VideoReward training prompts. Each model generates three videos per prompt using seeds 42, 43, and 44. Videos contain 53 frames at 16 FPS and 480×832 resolution. Generation uses 50 sampling steps, flow shift 3.0, and an empty negative prompt. The resolution, frame count, and sampling step count follow Ni et al. (2026). Following its choice of VideoReward metrics,

we report the mean raw VQ and MQ scores over the resulting 3,000 videos per model in Figure 1. This evaluation uses the same trained models as Table 3, with a different prompt set and generation protocol.

OCR. The OCR evaluation uses 1,018 prompts and three seeds per prompt, giving 3,054 videos per model. We sample frames at indices 0, 4, 8, 12, 16, 20, 24, 28, and 32, for a total of 27,486 frames per model. The target is the first quoted text in the prompt. Both the recognized and target strings are normalized by removing spaces and converting to lowercase. Each frame receives a score of $1 - \min(d, L)/L$, where d is the Levenshtein distance and L is the target length. OCR Reward averages the positive frame scores within each video, assigning zero if none are positive. All Frame OCR averages all nine frame scores, including zeros. Frame Exact Match is the percentage of sampled frames for which the normalized strings are identical. We first compute each metric per video and then average over videos. Since every video contributes nine frames, the latter two metrics are also equivalent to averaging over all sampled frames. Frame Exact Match is therefore a frame statistic, not a video success rate. In Figure 4, FlowSAPO yields 254 exact matches among 27,486 sampled frames, corresponding to 0.924% at the displayed precision.

B.4 Compared methods

FlowGRPO clips the global statistic of Equation (6), with width 10^{-4} for images and 10^{-3} for videos. The 1×1 variant uses the local branch of Equation (9) alone, with $\epsilon_f = 0.10$. FlowSAPO combines the local and regional branches with the coefficients given in Appendix B.1. We reproduce DiffusionNFT using its official code, with matched training configurations and epoch budgets for each image task. The DanceGRPO and ViPO scores marked with asterisks in Figure 1 are taken directly from Ni et al. (2026). TeleBoost publicly implements ViPO's structured advantage computation, but includes its own choices for constructing the allocation map and explicitly does not claim "bit-for-bit official paper reproduction."[1] ViPO uses Wan2.1 with 14B parameters in its video experiments, whereas our runs use the model with 1.3B parameters. These published scores therefore serve as literature comparisons under the respective reported settings.

Training seed and checkpoint selection. We fix the base training seed to 3407 for all experiments run in this work. Within each task, our FlowGRPO, 1×1, and FlowSAPO runs use matched training batch sizes and optimizer update budgets. For each method, we select the checkpoint with the highest evaluation reward for the training objective within the common step budget. The selected checkpoints used for the reported results are at step 712 for Wan, 720 for FLUX with PickScore, 5040 for SD3.5 Medium with GenEval, and 4320 for SD3.5 Medium with PickScore. These step counts refer to optimizer updates rather than denoising sampling steps. DiffusionNFT uses the same reward based selection criterion within the matched epoch budget. These rules apply to our reproduced runs; results taken from other papers retain their original settings. Following the benchmark reporting format of FlowGRPO, DiffusionNFT, and ViPO (Liu et al., 2025a; Zheng et al., 2026; Ni et al., 2026), we report aggregate scores for the evaluated checkpoints. Paired generation seeds are used as specified in the evaluation protocols. We therefore do not require paired confidence intervals or estimates of variability across independent training runs for the task comparisons reported in this work.

B.5 Training cost

Table 6 compares the average time per optimizer step and sampled peak memory for PickScore optimization on FLUX and SD3.5 Medium.

[1] https://github.com/Tele-AI/TeleBoost/blob/main/teleboost/algorithms/vipo.py

Table 6: Training cost of PickScore optimization on eight NVIDIA H800 GPUs. Average time per step is the cumulative elapsed time divided by the number of optimizer updates. Memory is the sampled peak per GPU.

Backbone	Method	Average time (s/step)	Sampled peak memory (GiB / GPU)
FLUX	FlowGRPO	185.59	72.84
	FlowSAPO	211.40	73.03
SD3.5 Medium	FlowGRPO	19.65	62.97
	FlowSAPO	25.07	62.97

Relative to FlowGRPO, FlowSAPO increases average time per step by 13.9% on FLUX and 27.6% on SD3.5 Medium, with little change in sampled peak memory. This additional cost accompanies faster reward improvement: on FLUX, Figure 8 shows that FlowSAPO surpasses the final displayed FlowGRPO reward with fewer updates. Together with the higher PickScore results on both backbones in Table 1, this supports improved optimization efficiency in terms of training updates.

C Text Alignment as a Separate Reward

Setting. We conduct an additional experiment on Wan2.1 (1.3B) using only the text alignment (TA) component of VideoReward (Liu et al., 2025b) as the training reward. This experiment tests whether FlowSAPO remains effective under a different reward objective. Our focus is generality across reward functions; balancing multiple objectives is outside the scope of this work. Evaluation follows the VBench prompt suite and generation settings in Appendix B.3. We report all three VideoReward components and the same eight VBench dimensions as in Table 3.

Table 7: Wan2.1 (1.3B) results with TA as the sole training reward. Better scores between FlowGRPO and FlowSAPO are highlighted in blue, based on unrounded values. Higher is better.

Method	VideoReward			VBench Qual.					VBench Sem.		
	VQ	MQ	TA	Subject Cons.	Background Cons.	Motion Smooth.	Aesthetic Qual.	Imaging Qual.	Object Class	Multiple Objects	Color
FlowGRPO (TA only)	2.702	0.893	2.143	0.958	0.967	0.980	0.637	0.651	0.740	0.631	0.919
FlowSAPO (TA only)	3.155	0.899	2.288	0.972	0.967	0.988	0.636	0.609	0.782	0.737	0.914

Results. Table 7 shows that FlowSAPO raises TA from 2.143 to 2.288 relative to FlowGRPO and also achieves higher VQ and MQ. The advantage is reflected in the VBench multiple objects score, which increases from 0.631 to 0.737, providing an independent indication of improved semantic composition. These results extend the evidence from visual and motion quality to text alignment, supporting the applicability of FlowSAPO across reward objectives.

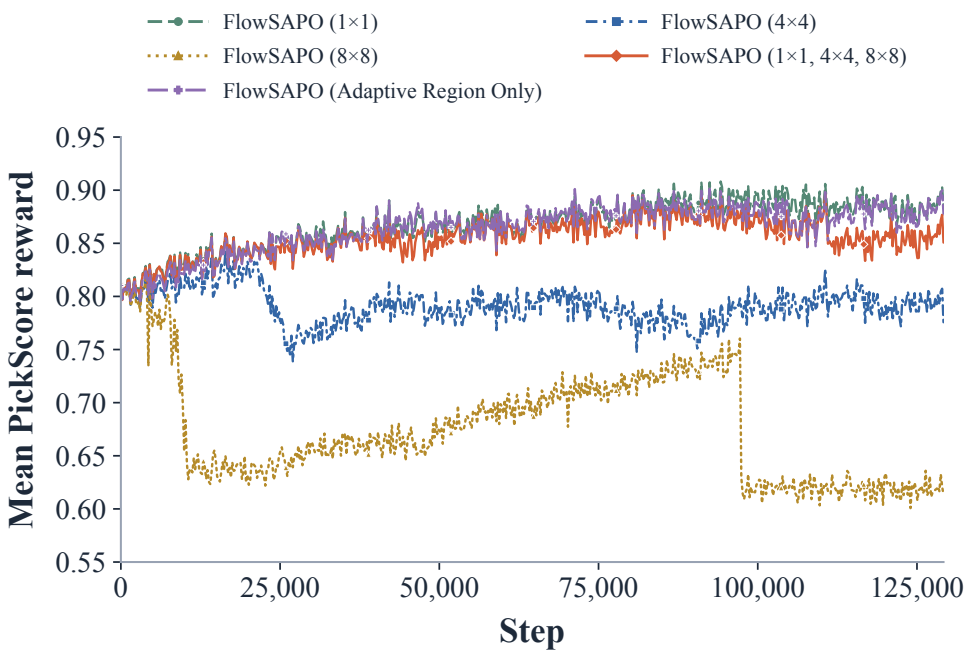

Figure 5: **Fixed patch size and regional adaptation on SD3.5 Medium.** Parentheses indicate the fixed patch sizes used to aggregate policy changes. The mixed variant combines 1×1, 4×4, and 8×8 patches. Adaptive Region Only uses the regional objective without the 1×1 branch. Curves show the logged mean training reward without smoothing.

D Supplementary Experiments

We extend Section 4.4 by examining fixed patch sizes and the balance between the two objectives, and clarify how the region cap controls spatial granularity. We then compare training stability on FLUX and inspect clipping behavior on SD3.5 Medium. All experiments optimize PickScore.

D.1 Fixed patches and adaptive regions

Figure 5 tests whether enlarging fixed patches is sufficient to improve local optimization. Both 4×4 and 8×8 patches exhibit substantial reward deterioration, with the larger patches showing repeated sharp drops. Combining all three sizes reduces the most severe degradation but remains below the 1×1 curve during the later stages. Simply increasing the aggregation area or mixing fixed scales therefore does not provide a reliable improvement.

Using adaptive regions alone produces rewards close to the 1×1 baseline, without a consistent advantage. Adaptation of the grouping is thus most useful in combination with the local objective. This motivates the next experiment, which examines how the two branches should be balanced.

D.2 Relative weights of the two objectives

The 1:3 mixture achieves the highest reward and sustains its improvement in the later stages of Figure 6. Reducing the regional weight to 1:9 leads to a late decline, while increasing it to 3:1 produces more pronounced intermediate drops and a lower final reward. The best result therefore comes from a balance between local measurement and regional coordination.

Although the local branch receives the larger coefficient in the 1:3 setting, the deterioration at 1:9 shows that its contribution alone does not account for the improvement. An intermediate regional weight is therefore useful in the tested setting. These are coefficients of the losses. Proposition 4

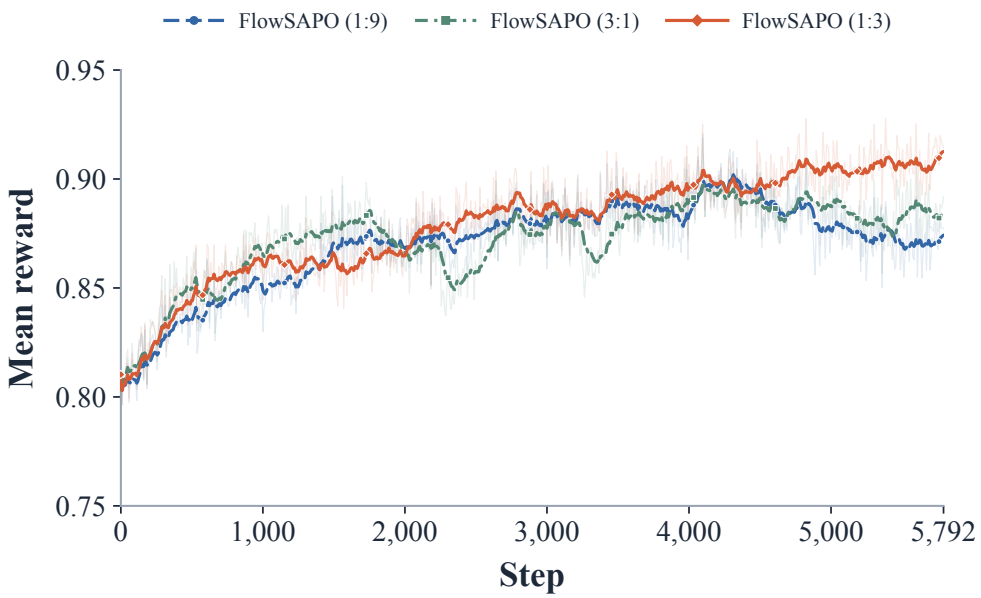

Figure 6: **Objective weight ablation on SD3.5 Medium.** Ratios specify the weight of the adaptive regional objective relative to the 1×1 objective. The settings 1:9, 1:3, and 3:1 correspond to $\lambda = 0.10$, 0.25, and 0.75 in Equation (14). Dark curves show exponential moving averages with a half life of 40 logged steps; faint traces show the unsmoothed rewards.

shows that their relative gradient contributions depend on normalization at rollout and on regional weights and clipping decisions during replay.

D.3 Maximum region size

We vary S_{clean} over 24, 48, 64, and 81 on SD3.5 Medium with PickScore, keeping all other training settings fixed. Figure 7 shows that caps of 48 and 64 achieve higher rewards at the later evaluation points than 24 or 81, with 64 giving the highest final value. Increasing the permitted region size therefore does not improve performance monotonically. This supports controlling the extent of each regional term alongside its attention based membership. The cap affects the full objective through membership, coverage, and the area dependent clipping width.

Within the schedule in Equation (29), the cap bounds the number of attention cells per region, and the similarity test can stop growth before the bound is reached. A cap of one removes interaction across attention cells within each regional term, but each cell still contains b latent positions. Larger caps permit coarser aggregation without forcing a region to span the grid. In either case, the regional objective retains its inverse coverage weighting and area dependent clipping width. These limits therefore describe spatial granularity; they do not make the objective identical to the 1×1 or global baseline. Proposition 5 explains how shared region membership affects the objective beyond its gradient at the rollout point.

D.4 Training stability on FLUX

Figure 8 examines whether regional coordination also helps once the stronger FLUX backbone has reached a high reward. The 1×1 objective improves rapidly at first, but undergoes an abrupt drop at the final evaluation point, near step 720. FlowSAPO maintains its progress through this stage and achieves the highest reward at the end of the displayed interval. Together with Table 1, these curves show that the regional branch supports both a stronger final result and steadier reward improvement.

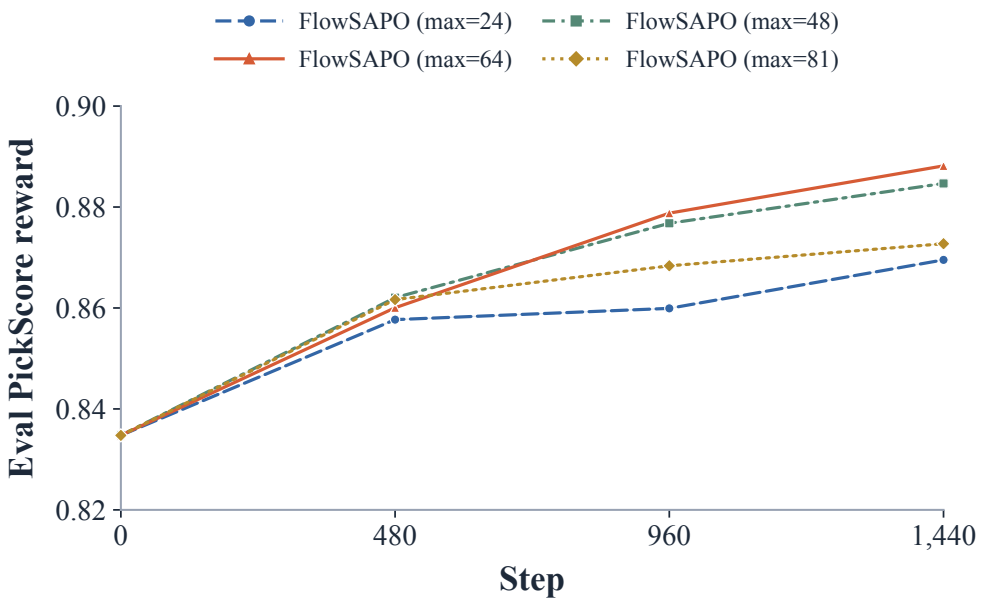

Figure 7: **Sensitivity to the region cap on SD3.5 Medium.** We vary the low noise cap S_{clean} while keeping all other training settings fixed. The vertical axis reports evaluation PickScore divided by 26. Dark curves show exponential moving averages with a half life of 40 logged steps, and faint traces show the unsmoothed values.

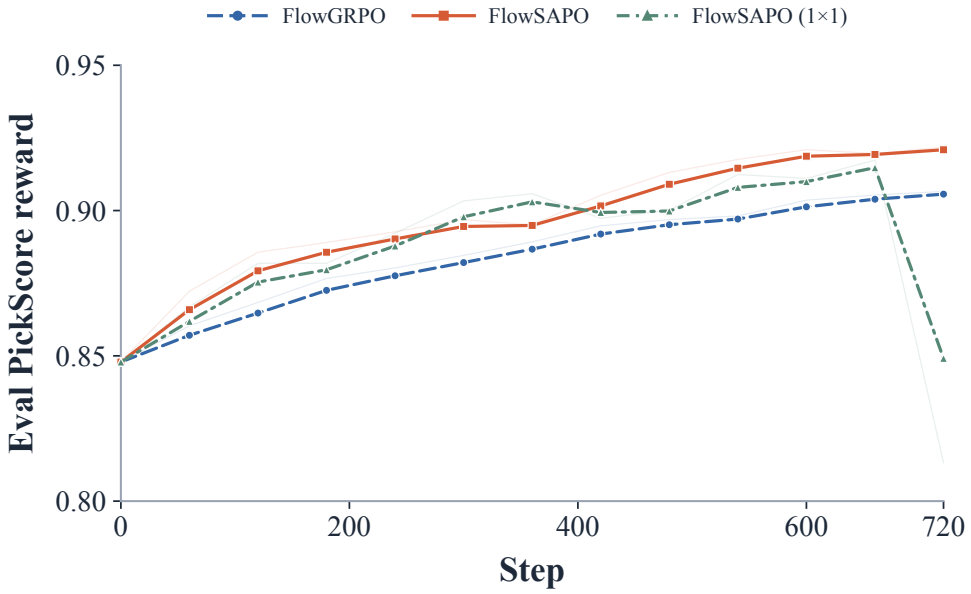

Figure 8: **PickScore evaluation during FLUX training.** We compare FlowGRPO, the 1×1 variant, and full FlowSAPO over the displayed training interval. Dark curves show exponential moving averages with a half life of 40 logged steps; faint traces show the original evaluation rewards.

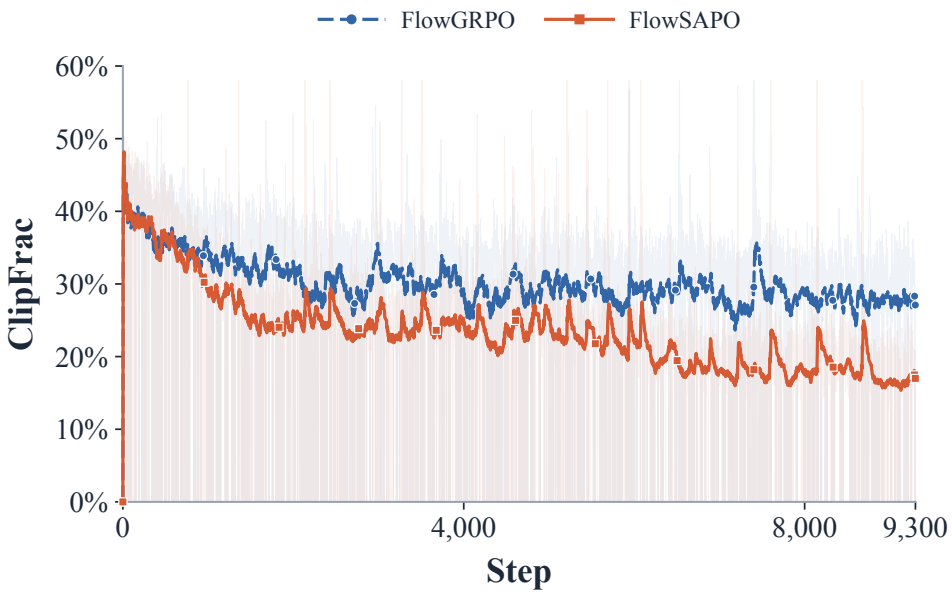

Figure 9: **Global clipping diagnostic on SD3.5 Medium.** ClipFrac is the fraction of transitions with negative advantage whose global ratio falls below $1-\epsilon$, as defined in Equation (15). Dark curves show exponential moving averages with a half life of 40 logged steps, and faint traces show the unsmoothed values.

D.5 Clipping under negative advantages

We measure the frequency with which transitions with negative advantage meet the lower clipping condition of the global ratio:

$$\text{ClipFrac}_ = \frac{\sum_{i,t} \mathbf{1}\{A_i < 0\}\mathbf{1}\{r_{i,t}^{\text{geo}} < 1-\epsilon\}}{\sum_{i,t} \mathbf{1}\{A_i < 0\}}, \tag{15}$$

where r^{geo} is the statistic in Equation (6). We use the same diagnostic threshold ϵ for FlowGRPO and FlowSAPO.

After the initial stage, FlowSAPO maintains a lower fraction than FlowGRPO in Figure 9. Fewer sampled transitions therefore meet the condition that would suppress their entire contribution under a global clipping decision. This is evidence that the two training procedures encounter the global clipping condition at different frequencies. The statistic measures exposure to the global gate; the gradients retained by FlowSAPO itself are determined by the local and regional objectives in Equation (14).

E Supplementary Image Comparisons

In Figure 10, FlowSAPO retains the Lego form with detailed surroundings and places the sheep to the right of the wine glass, while the other aligned models reverse the spatial relation. These examples extend Figure 3 to style and spatial composition. The sign example also shows a remaining text rendering error: FlowSAPO writes “Brarin” in place of “Brain,” while the 1×1 variant replaces that word with an illustration. This example therefore concerns visual composition rather than exact transcription of the requested text.



Figure 10: **Additional qualitative comparisons on FLUX.1 [dev].** Prompts and methods are labeled in the figure. Aligned models are optimized with PickScore.

F Adaptive Region Visualizations on FLUX

Figures 11 and 12 visualize the adaptive regions constructed by FlowSAPO during FLUX sampling. Each row shows six regional views of the same intermediate image, with time advancing downwards. Spatial seeds are selected using subject attention and spatial coverage, and regions are rebuilt independently at each displayed step following Appendix I. The selected seed locations can therefore change between rows.

For display, we apply Gaussian smoothing to each binary region mask and upsample it with bicubic interpolation. The mask is overlaid on the RGB image decoded after the corresponding denoising update; the attention used to construct the region is collected before that update. This rendering does not change the stored region membership.

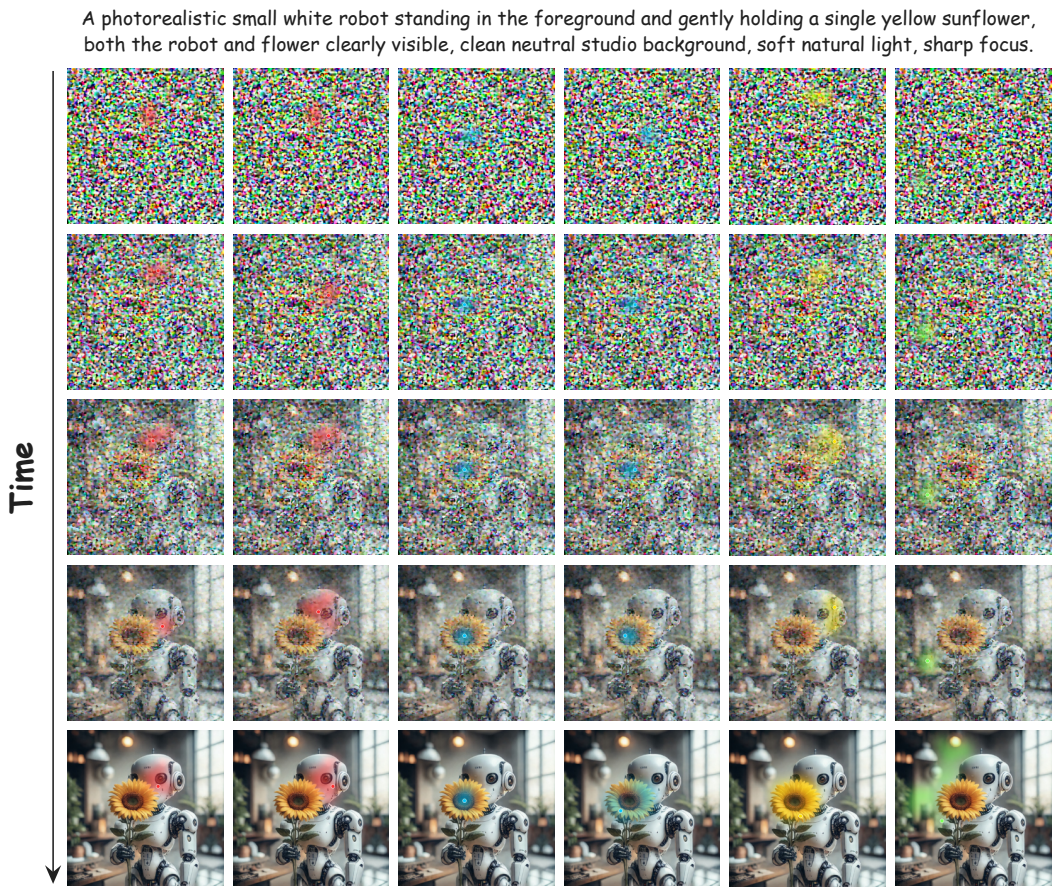

Figure 11: **Adaptive regions during FLUX sampling: robot and sunflower.** The first two columns show regions selected for the robot (red), followed by two regions for the sunflower (cyan), an additional foreground region (yellow), and a context region (green). Rows show five stages of the same denoising trajectory, ordered from top to bottom. Dots mark the spatial seeds. The colored fields depict smoothed region masks.

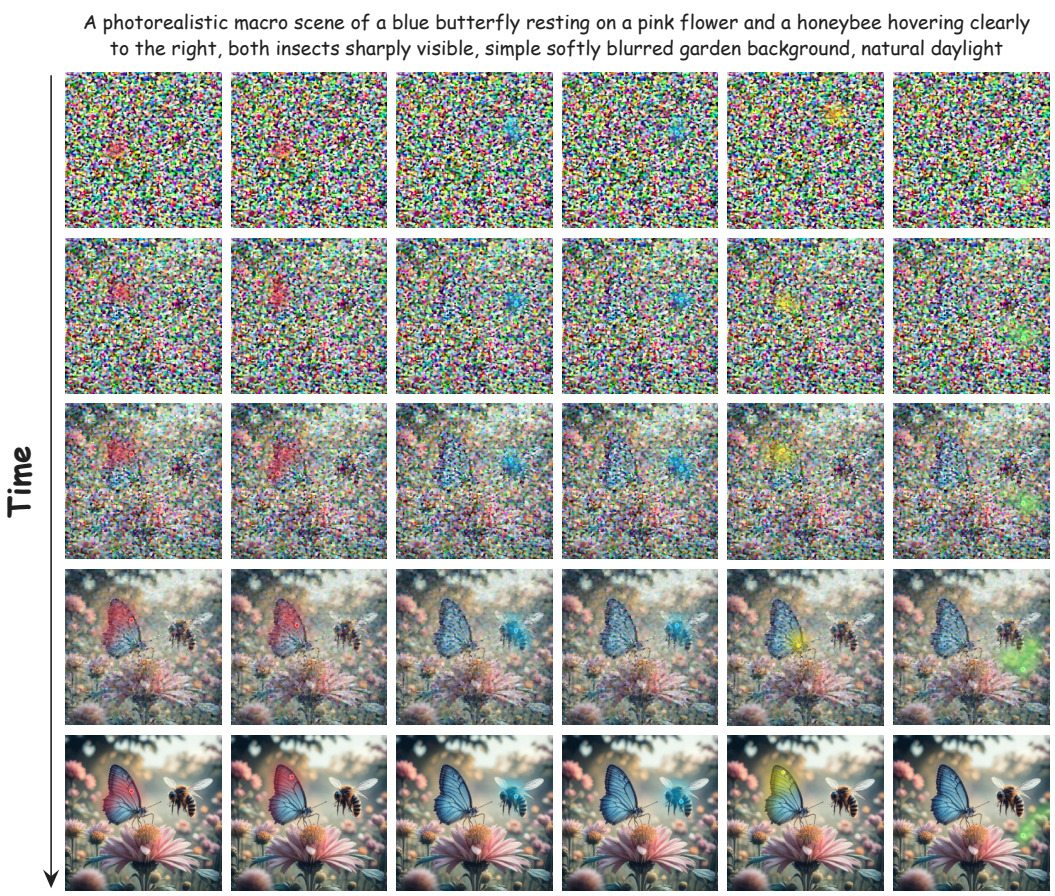

Figure 12: **Adaptive regions during FLUX sampling: butterfly and bee.** The first two columns show regions selected for the butterfly (red), followed by two regions for the bee (cyan), an additional foreground region (yellow), and a context region (green). Time advances from top to bottom across five sampling stages. Dots identify the spatial seeds, whose locations can change between rows.

G SUPPLEMENTARY VIDEO COMPARISONS

We compare FlowGRPO and FlowSAPO on Wan2.1-T2V 1.3B after optimization with VideoReward. Figures 13 and 14 present first frames for four prompts and two seed settings per prompt. Figure 15 shows six frames ordered in time, complementing the appearance comparisons with a view of how the generated sequence evolves.



Figure 13: **First-frame comparison after VideoReward optimization: boat and car scenes.** The left and right columns show FlowGRPO and FlowSAPO, respectively, using Wan2.1-T2V 1.3B optimized with VideoReward. Each prompt block presents the first frame of the videos generated with Seed 0 and Seed 1: (a) a boat slowing down to stop; (b) a car moving slowly along an empty street on a rainy evening. These frames compare visual appearance and scene content at the start of each video.

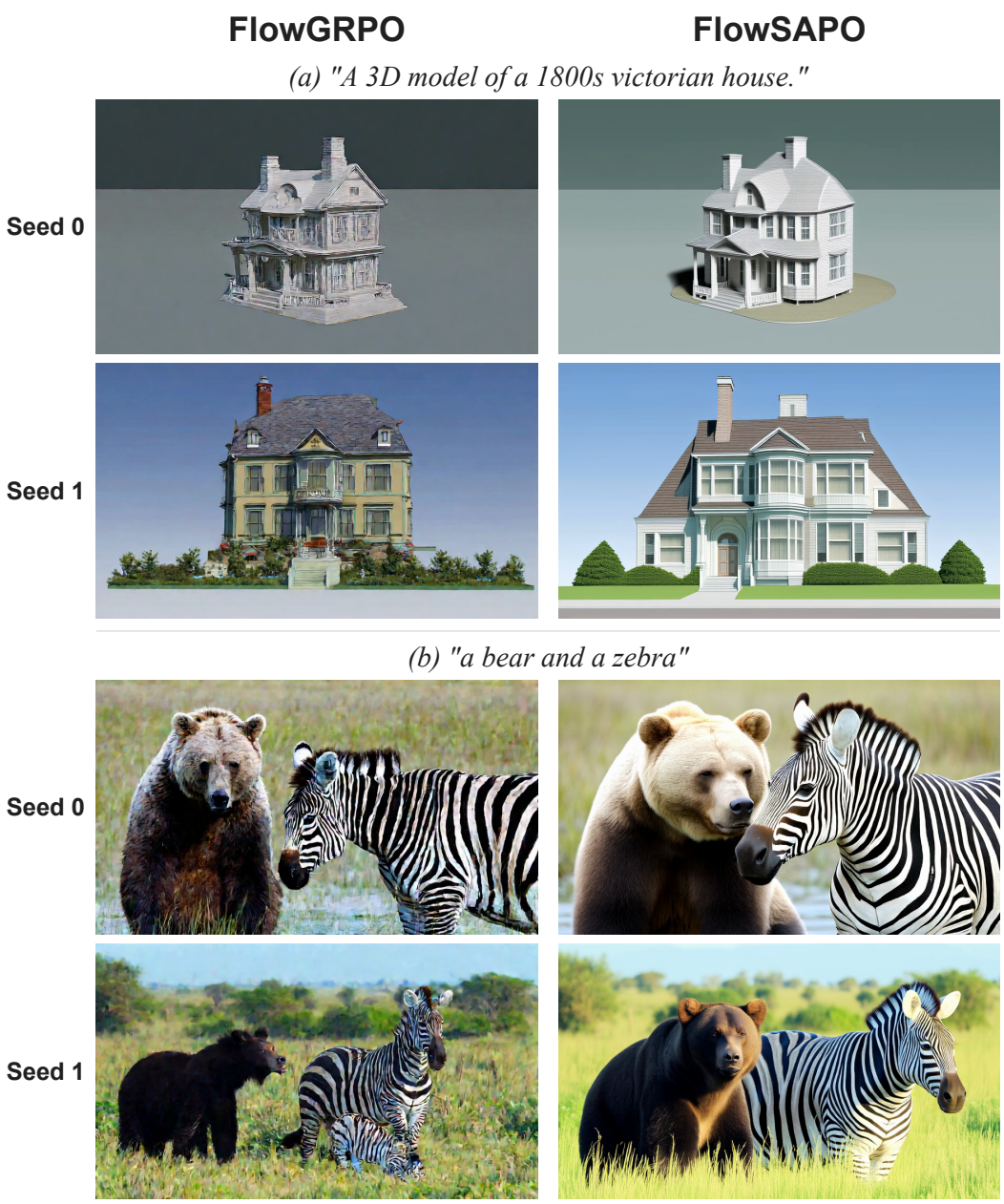

Figure 14: **First-frame comparison after VideoReward optimization: architecture and animals.** The left and right columns show FlowGRPO and FlowSAPO, respectively, using Wan2.1-T2V 1.3B optimized with VideoReward. Each prompt block presents the first frame of the videos generated with Seed 0 and Seed 1: (a) a 3D model of an 1800s Victorian house; (b) a bear and a zebra. The first frames show architectural structure, subject appearance, and scene composition.



Figure 15: **Temporal comparison after VideoReward optimization.** The left and right columns show FlowGRPO and FlowSAPO, respectively, using Wan2.1-T2V 1.3B optimized with VideoReward. Six sampled frames depict a man lowering a telephone handset. Time progresses from top to bottom, as indicated by the arrow, allowing comparison of the changes in pose, hand position, and subject appearance across the sequence.

H Analysis of Global, Positionwise, and Regional Clipping

We analyze one replayed transition at a fixed stored state. Throughout, the current and rollout policies share the diagonal covariance of Equation (1), the advantage A is fixed, and the stored cover is part of the rollout. We examine what the global statistic reveals about a policy change, what a local factor retains, and what regional clipping adds to positionwise clipping. Table 8 lists the results; after each one, a paragraph states what it shows and, where a stronger reading is tempting, what it does not. Numerical examples use the latent geometry of SD3.5 Medium at 512^2, with $C = 16$ channels and $N_f = 4096$ latent positions. They illustrate the Gaussian model and are not measurements of training trajectories. Proofs are in Appendix K.

Table 8: Results of the analysis and their implications for policy optimization.

Result	What it shows
Proposition 1	The global weight depends on the policy change only through its total; a local factor depends only on the change at its own position.
Proposition 2	Opposing local log ratio changes cancel in the global statistic; the cancellation index is about 94% in the illustrative homogeneous Gaussian case.
Corollary 1	Global clipping decides by the average divergence alone and one-sidedly; positionwise decisions differ across positions.
Theorem 1	A local factor is the best approximation of the joint ratio that sees one position; the 1×1 objective still drops cross-position terms.
Proposition 3	A regional score has a variance bound proportional to area; its actual variance also depends on coverage.
Proposition 4	At rollout, the regional surrogate gradient is proportional to the local one; during replay, both regional weights and shared clipping gates matter.
Proposition 5	Without clipping, the grouping still changes the objective at second order.
Proposition 6, Corollary 2	The builder returns a deterministic cover of connected regions that includes every cell and survives small attention changes.

H.1 The global statistic sees only the total change

Write K_p for the KL divergence at latent position p from the rollout-policy action factor to the current-policy one, $K = \sum_p K_p$ for the whole transition, and $D = CN_f$ for its number of coordinates.

Proposition 1 (Only the total change reaches the global weight)**.** *For actions drawn from the rollout policy at a fixed stored state,*

$$\log r^{\mathrm{joint}} \sim \mathcal{N}(-K,\, 2K), \qquad \log r^{\mathrm{geo}} \sim \mathcal{N}\big(-\tfrac{K}{D},\, \tfrac{2K}{D^2}\big)\,, \qquad \delta_p \sim \mathcal{N}(-K_p,\, 2K_p), \tag{16}$$

and the δ_p are independent across positions. The joint ratio and each local factor are density ratios and have mean one; the global weight has mean $\exp(-K/D + K/D^2) < 1$ whenever $K > 0$ and $D > 1$.

What it shows. The three statistics describe the same policy change at different resolutions. The joint ratio can be highly dispersed: at a divergence of $k = 0.005$ per position, about two thirds of the local factors stay inside a 10% band, whereas the joint weight for the SD3.5 geometry lies in that band with probability below 10^{-4}. Its log has mean -20.5 and standard deviation 6.4. The global log statistic divides this total by D, giving mean $-K/D$ and standard deviation $\sqrt{2K}/D$. Two changes with the same total divergence but different spatial patterns therefore give the same distribution of the global weight. In contrast, each local log ratio has standard deviation $\sqrt{2K_p}$ and retains the dependence on its own position. This identifies a loss of spatial resolution in the global statistic, rather than establishing that averaging is always harmful.

Proposition 2 (Opposing local changes cancel in the global statistic)**.** *Let $K_p = k > 0$ at every position. As $N_f \to \infty$, the cancellation index $1 - |\sum_p \delta_p| / \sum_p |\delta_p|$ converges in probability to $1 - k/m(k) = 1 - \frac{1}{2}\sqrt{\pi k} + O(k^{3/2})$, where $m(k) = \mathbb{E}|\delta_p|$, and $\max_p \delta_p / |\log r^{\mathrm{geo}}| = C\sqrt{4 \ln N_f / k}\,(1 + o_p(1))$.*

What it shows. The cancellation index quantifies how much the signed sum differs from the sum of absolute local log ratio changes. In the illustrative homogeneous case with $k = 0.005$, it approaches

0.937, while the global weight concentrates near 0.9997. The index approaches one as k decreases because the random local fluctuations scale as $\sqrt{k}$ while their common mean scales as k. Thus the quoted 94% describes cancellation in this Gaussian example, not the fraction of useful optimization signal lost in training. Averaging can also suppress noise; the empirical comparisons are needed to assess the effect of local measurement on optimization.

Corollary 1 (Global clipping decides by the average divergence alone)**.** *Under the assumptions of Proposition 2, let $0 < \epsilon_g, \epsilon_f < 1$ define global and positionwise clipping bands. (i) The magnitude of the mean of $\log r^{\mathrm{geo}}$, k/C, exceeds its standard deviation, $\sqrt{2k/N_f}/C$, by the factor $\sqrt{kN_f/2}$. As $N_f \to \infty$, the probability that the global weight leaves its band tends to*

$$\mathbf{1}\{e^{-k/C} < 1 - \epsilon_g\} + \tfrac{1}{2}\mathbf{1}\{e^{-k/C} = 1 - \epsilon_g\}. \tag{17}$$

(ii) The local factor at p leaves its band with probability $\Phi\big(\frac{\log(1-\epsilon_f)+k}{\sqrt{2k}}\big) + 1 - \Phi\big(\frac{\log(1+\epsilon_f)+k}{\sqrt{2k}}\big)$, strictly between 0 and 1 for every $k > 0$.

What it shows. Away from the boundary case in Equation (17), the global decision becomes determined by the average divergence as the grid grows. In the illustrative SD3.5 case with $k = 0.005$ and the baseline width $\epsilon_g = 10^{-4}$, the global weight falls below its band with probability 0.985. Under the fixed negative advantage assumed here, this event removes the gradient of the global clipped surrogate. GRPO-Guard reports a related shift of the ratio below one in flow matching GRPO (Wang et al., 2026); Proposition 1 describes it under a shared covariance Gaussian transition. The concentration increases with grid size, while at the same local divergence a positionwise factor leaves its 10% band with probability 0.32. Local clipping therefore retains decisions that vary across positions even when the global statistic is strongly concentrated.

What it does not show. The corollary fixes one divergence for every position and every sample. When the divergence varies, the global decision still depends on each sample only through its average divergence; it does not become a function of where the change happened.

Theorem 1 (A local factor is the best one-position approximation of the joint ratio)**.** *Let $a = (a_1, \ldots, a_N)$ be a disjoint conditional action partition at a fixed state s, with rollout density $p_0(a \mid s) = \prod_n p_{0,n}(a_n \mid s)$ and current density $p(a \mid s) = \prod_n p_n(a_n \mid s)$. Assume $p_n(\cdot \mid s) \ll p_{0,n}(\cdot \mid s)$ for every n. Write $r_n = p_n/p_{0,n}$, $r_B = \prod_{n \in B} r_n$ for a set B of factors, and $r_{\mathrm{joint}} = \prod_n r_n$, and assume $r_{\mathrm{joint}} \in L^2(p_0(\cdot \mid s))$. (i) For every factor p,*

$$r_p(a_p \mid s) = \mathbb{E}_{p_0}[r_{\mathrm{joint}}(a \mid s) \mid s, a_p]\,, \tag{18}$$

so r_p uniquely minimizes the conditional mean-squared error to the joint ratio among functions of (s, a_p). (ii) For sets of factors $B \subseteq B'$, $\mathrm{Var}_{p_0}(r_B \mid s) \le \mathrm{Var}_{p_0}(r_{B'} \mid s)$.

What it shows. The local factor is not an approximation chosen by FlowSAPO: it is what the joint ratio predicts once only the action at one position is known, and no other function of that action predicts it better in mean square. Part (ii) is the general form of the growth of $\mathrm{Var}(r) = e^{2K} - 1$ with K: an exact ratio over a larger unit is never less variable. Neither part needs the Gaussian form; both hold for product densities under the stated absolute continuity and square integrability assumptions.

What it does not show. A sum of local PPO terms is not the joint objective. With $\eta_p = r_p - 1$,

$$\mathbb{E}_{p_0}[A(r_{\mathrm{joint}} - 1)] = \sum_{\emptyset \neq S \subseteq \{1,\ldots,N\}} \mathbb{E}_{p_0}\left[A \prod_{p \in S} \eta_p\right], \tag{19}$$

and the $1{\times}1$ objective keeps the terms with one position and drops the products over two or more positions, each at least second order in the per-position changes. FlowSAPO claims an exact local measure of policy change, not a reconstruction of the joint reward-weighted objective.

H.2 Regional weights and shared clipping decisions

The stored cover is deterministic given the attention features and the area cap (Proposition 6, Appendix I). The analysis uses only that every cell belongs to at least its own seeded region, so $n(q) \ge 1$, and that regions may overlap. For region j define $a_{jq} = \mathbf{1}\{q \in \mathcal{R}_j\}/n(q)$, so that $s_j = \sum_q a_{jq} u(q)$, and let K_q be the KL divergence of the action block of cell q.

Proposition 3 (A regional score's spread grows at most as the square root of its area)**.** *Under the shared-covariance transition and before the numerical guard,*

$$s_j \sim \mathcal{N}\Big(-\sum_q a_{jq}K_q,\ 2\sum_q a_{jq}^2 K_q\Big), \qquad \mathbb{E}_{\text{old}}[e^{s_j}] = \exp\Big[-\sum_q a_{jq}(1-a_{jq})K_q\Big] \le 1, \quad (20)$$

with equality only when no contributing cell with nonzero divergence is shared. If $K_p = k$ at every position, the standard deviation of s_j is at most $\sqrt{2k\,a_j}$, where $a_j = b\,|\mathcal{R}_j|$ is the latent area of the region, with equality when no cell of $\mathcal{R}_j$ is shared.

What it shows. The variance bound provides a scale reference for Equation (12). With $\nu = 0.5$, the width is proportional to the upper bound on the standard deviation. The actual variance need not grow with area: for a regular cover with m cells per region and coverage $n(q) = m$, it equals $2bk/m$. The training rule $\epsilon_f a_j^{0.7}$ is wider than the square root reference by a factor $a_j^{0.2}$. With $\epsilon_f = 0.1$, its upper log boundary is about 1.2 to 1.4 times $\sqrt{a_j}\log(1+\epsilon_f)$ over the areas considered in Appendix K. This is a comparison of boundaries, not of clipping probabilities. The expectation in Equation (20) also confirms that overlapping regions generally produce weights with mean below one, rather than density ratios.

What it does not show. The proposition does not select the exponent; $\nu = 0.7$ remains a design choice. The comparison of boundary widths does not establish that large regions are clipped less often, since clipping also depends on the distribution of regional scores and the signs of the advantages. For either exponent, the lower boundary becomes inactive once $\epsilon_j \ge 1$, because the regional weight is positive. With $\epsilon_f = 0.1$ and $\nu = 0.7$, this occurs from 27 latent positions onward, as detailed in Appendix K.

To see how the cover enters the update, write $\mathcal{J}_{\text{geo}}$, $\mathcal{J}_{\text{fine}}$, and $\mathcal{J}_{\text{region}}$ for the unclipped, unguarded versions of the three objectives: each averages $\mathbb{E}_{\text{old}}[A\,r]$ over its own units, one weight per transition, N_f local factors, or N_a regional weights. Away from clip and guard boundaries, let $I_j = \mathbf{1}\{A > 0, \bar{w}_j < 1+\epsilon_j\} + \mathbf{1}\{A < 0, \bar{w}_j > 1-\epsilon_j\}$ indicate that the clip of region j is open, and $\gamma_j = -A\bar{w}_j I_j \mathbf{1}\{|s_j| < 20\}$ its signed gradient factor.

Proposition 4 (Gradient at rollout and regional coupling during replay)**.** *(i) At $\theta = \theta_{\text{old}}$, for any stored cover,*

$$\nabla\mathcal{J}_{\text{fine}} = C\,\nabla\mathcal{J}_{\text{geo}}, \qquad \nabla\mathcal{J}_{\text{region}} = b\,\nabla\mathcal{J}_{\text{fine}}, \qquad b = N_f/N_a, \quad (21)$$

all three are multiples of the policy gradient $\mathbb{E}_{\text{old}}[A\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)]$, and the same holds for the clipped, guarded losses. (ii) Away from the boundaries, the derivative of the per-transition regional loss with respect to the cell score $u(q)$ is

$$\frac{\partial \mathcal{L}_{\text{region}}^{(i,t)}}{\partial u(q)} = \frac{1}{N_a n(q)} \sum_{j:\, q\in\mathcal{R}_j} \gamma_j. \quad (22)$$

Part (i) rests on one identity. Because each cell is divided by its coverage,

$$\sum_j s_j = \sum_j \sum_{q\in\mathcal{R}_j} \frac{u(q)}{n(q)} = \sum_q u(q) = \sum_p \delta_p, \quad (23)$$

so at the rollout policy the regional objective differentiates to the same sum of local scores as the 1×1 objective, averaged over N_a regions instead of N_f positions.

What it shows. At the rollout policy, the mixed surrogate gradient equals the local surrogate gradient multiplied by $(1-\lambda)+\lambda b$, independently of the cover. During replay, the cover affects both the weights $\bar{w}_j$ and the shared clipping gates I_j in part (ii). Even when every gate remains open, the exponential weight couples each cell to the other members of its region. Once a regional gate closes, it suppresses that region's term for all of its members, while overlapping regions may still contribute. Neither effect requires a spatial advantage or a regional density ratio.

What it does not show. Part (i) is an identity of the loss gradient, not of the parameter update: AdamW divides each coordinate by a running estimate of its gradient magnitude, so a gradient scaled by $(1-\lambda)+\lambda b$ does not produce a step scaled by the same factor.

Proposition 5 (The grouping matters at second order even without clipping)**.** *For a frozen cover and a fixed advantage A, let $F(\theta) = AN_a^{-1} \sum_j e^{s_j(\theta)}$ be the unclipped, unguarded regional objective of one transition, and $\theta_0 = \theta_{\text{old}}$. Then*

$$\nabla F(\theta_0) = \frac{A}{N_a} \nabla \sum_p \delta_p(\theta_0), \tag{24}$$

$$\nabla^2 F(\theta_0) = \frac{A}{N_a} \Big[\nabla^2 \sum_p \delta_p(\theta_0) + \sum_j \nabla s_j(\theta_0) \, \nabla s_j(\theta_0)^\top \Big]. \tag{25}$$

The gradient and the first term of the Hessian do not depend on the cover; the outer-product sum does.

What it shows. Two covers give the same surrogate gradient at the rollout policy but can have different curvature before any clipping occurs. To make this dependence explicit, treat the cell scores $u(q)$ as independent variables and define $W_{jq} = \mathbf{1}\{q \in \mathcal{R}_j\}/n(q)$. For a fixed advantage and a frozen cover, the unclipped, unguarded regional objective is $\mathcal{J}_{\mathcal{R}}(u) = AN_a^{-1} \sum_j \exp((Wu)_j)$. Inverse coverage gives $W^\top \mathbf{1} = \mathbf{1}$, and hence

$$\nabla_u \mathcal{J}_{\mathcal{R}}(0) = \frac{A}{N_a} \mathbf{1}, \qquad \nabla_u^2 \mathcal{J}_{\mathcal{R}}(0) = \frac{A}{N_a} W^\top W. \tag{26}$$

The entries of $W^\top W$ have a direct interpretation:

$$(W^\top W)_{qq'} = \frac{|\{j : q, q' \in \mathcal{R}_j\}|}{n(q) n(q')}, \qquad (W^\top W)_{qq} = \frac{1}{n(q)}. \tag{27}$$

Thus, shared region membership determines the mixed second derivatives, while coverage also affects the diagonal terms. These derivatives are with respect to the cell scores; Equation (25) includes their dependence on the model parameters through the chain rule.

The cap controls this structure by limiting the membership of each regional term. Changing it preserves one seeded region per cell and therefore leaves N_a unchanged. On the same fixed rollout, the cap introduces no additional scaling of the initial surrogate gradient; its effect appears through regional weighting and clipping during replay. This explains why the cap can affect optimization even though the gradient at the rollout point is independent of the cover. The identity does not equate the parameter updates of separate training runs, whose sampled states, advantages, and optimizer histories can differ.

I Adaptive Region Construction

This appendix gives the complete construction summarized in Section 3.2. Algorithm 1 constructs the cover for one denoising step, and Algorithm 2 specifies the growth of each seeded region. We detail the acceptance test and area schedule below, followed by the properties of the resulting cover in Appendix I.1.

Choice of grouping features. A region in FlowSAPO determines which local log-ratios are aggregated before exponentiation and which local policy changes interact through a common regional surrogate. The grouping signal should therefore reflect associations relevant to the current generator and prompt, rather than merely generic visual similarity. DINO and ResNet features provide useful perceptual representations, while SAM provides object-oriented masks, and all could serve as alternative grouping signals after appropriate adaptation. In their standard form, however, these representations are not conditioned on how the current generator associates latent positions with prompt tokens at a particular denoising state. Using them would additionally require choosing an image estimate at each step and mapping the resulting features or masks back to the latent grid. We instead use text attention from the conditional rollout branch. It is available at the stored state and timestep, is directly indexed by prompt tokens, and has a spatial correspondence to the generator's latent representation determined by the backbone. This avoids introducing a separate external perception model, although attention capture and region construction still incur computational overhead. Our choice should therefore be understood as an objective-aligned design choice rather than a claim that text attention is universally superior to alternative visual features.

DINO features contain semantic structure (Caron et al., 2021), and ResNet features provide representations learned for visual recognition (He et al., 2016). In their standard form, however, these image features do not describe how the current generator associates latent positions with the input prompt. SAM provides promptable segmentation (Kirillov et al., 2023), but its masks likewise do not directly express this association or determine the aggregation size of the policy objective. Using these models would require choosing which image estimate to process at each denoising step and mapping the resulting features or masks to the latent grid. They could provide alternative grouping signals after such adaptation. We use text attention from the conditional rollout branch, whose spatial associations with prompt tokens have also been used to guide diffusion editing (Hertz et al., 2022). These features are available for the sampled state and timestep, with a correspondence to latent positions determined by the backbone. This avoids a separate external feature extractor, although attention capture and region construction still incur overhead.

Text attention supplies the association signal, while the growth rule enforces connectivity, bounds region size, and covers every cell through overlapping regions. These structural properties follow from the builder described below. The fixed patch experiments in Appendix D.1 show that enlarging patches or mixing fixed sizes does not reliably improve local optimization, and Appendix D.3 shows that increasing the area cap does not improve performance monotonically. Together, these observations support controlling both region membership and extent. The visualizations in Appendix F illustrate how the resulting regions vary during denoising. We use attention to associate positions for optimization, without treating it as an object annotation or a source of local reward. A direct comparison with external grouping features would require evaluating them under comparable region sizes and coverage; the present ablations assess the proposed construction and its parameters.

Growing a region from every cell. Using the stored attention features $f(q)$, we grow regions by a rule with three parts. Every cell seeds one region, which grows over its eight connected frontier by cosine similarity to the seed's feature, $\kappa_j(q) = \langle f(j), f(q)\rangle / (\|f(j)\|\,\|f(q)\|)$. First, similarity is always measured to the seed, never to the last accepted cell, so a region cannot drift along a chain of locally similar cells. Second, a frontier cell is accepted when its similarity reaches the mean similarity of the cells already accepted together with the 3×3 neighbourhood of the accepted cell it touches; this contextual threshold replaces a global constant, and after each acceptance, always of the most similar admissible cell, the thresholds are recomputed. Third, growth stops when no frontier cell is admissible or the region reaches the area cap. For video, the rule runs within each latent frame, so regions do not cross frames. Algorithm 1 summarizes the construction, with $W_3(u)$ denoting the 3×3 window centred at u. Algorithm 2 specifies the acceptance test and the order for resolving ties in full.

Algorithm 1 Adaptive regions for one stored denoising step, built at rollout without gradient.

Input: text attention of the conditional branch at step t; area cap $S_t^{\max}$ ▷ Eq. (29)
$f(q) \leftarrow$ attention of cell q to the prompt tokens, averaged over heads and layers, padded tokens zeroed
for each cell $j \in \Omega_a$ **do** ▷ every cell seeds one region
  $\kappa(q) \leftarrow \cos\big(f(j), f(q)\big)$ for every cell q; $\mathcal{R}_j \leftarrow \{j\}$
  while $|\mathcal{R}_j| < S_t^{\max}$ **do**
    $\mathcal{E} \leftarrow$ 8-neighbours $v \notin \mathcal{R}_j$ of cells $u \in \mathcal{R}_j$ with $\kappa(v) \geq \vartheta(u)$ ▷ $\vartheta(u)$: mean κ over $\mathcal{R}_j \cup W_3(u)$
    if $\mathcal{E} = \emptyset$ **then stop**
    add to $\mathcal{R}_j$ the cell of $\mathcal{E}$ with the largest κ ▷ ties by row-major index
  end while
end for
$n(q) \leftarrow$ number of regions $\mathcal{R}_j$ that contain q, for every cell q
Output: regions $\{\mathcal{R}_j\}_{j\in\Omega_a}$ and coverage n, stored with the transition and frozen during replay

Algorithm 2 Growth of the region seeded at cell j for one attention feature map.

1. Normalize each cell feature and compute the cosine scores $\kappa_j(q) = \langle f(j), f(q) \rangle$ to the seed; set $\kappa_j(j) = 1$.
2. Start from the accepted set $\mathcal{S}_j = \{j\}$ and the ordered region $\mathcal{R}_j = [j]$.
3. While $|\mathcal{R}_j| < S_t^{\max}$, list every eight-connected edge (c, q) from a cell $c \in \mathcal{R}_j$ to a cell q not yet accepted, sorted by $(-\kappa_j(q), q, c)$.
4. For each edge in that order, let $E_c = \mathcal{N}_9(c) \setminus \mathcal{S}_j$ be the unaccepted cells of the 3×3 neighbourhood of c, and compute
$$\vartheta_{j,c} = \frac{\sum_{x \in \mathcal{R}_j} \kappa_j(x) + \sum_{x \in E_c} \kappa_j(x)}{|\mathcal{R}_j| + |E_c|}.$$
Accept the first q with $\kappa_j(q) \geq \vartheta_{j,c}$, add it to $\mathcal{S}_j$ and $\mathcal{R}_j$, and return to step 3.
5. Stop when no edge passes or the cap is reached, and store $\mathcal{R}_j$ with the rollout.
6. After all seeds, set the coverage $n(q)$ of each cell to the number of stored regions that contain it.

Acceptance test. A cell joins a region when it is at least as similar to the seed as its context. Because a candidate q belongs to E_c, the test in step 4 can be rewritten as

$$\kappa_j(q) \geq \frac{\sum_{x \in \mathcal{R}_j} \kappa_j(x) + \sum_{x \in E_c \setminus \{q\}} \kappa_j(x)}{|\mathcal{R}_j| + |E_c| - 1}, \tag{28}$$

so q is compared with the average of the region so far and of the parent's other unaccepted neighbours, not with a fixed cosine cutoff. Every comparison is made to the seed $f(j)$; the parent only chooses which neighbourhood is consulted, so a region cannot drift along a chain of locally similar cells.

Area cap. The cap limits the number of attention cells in a single regional term. For fixed attention features, reducing the cap stops the same growth sequence earlier, yielding a prefix of the region obtained with a larger cap (Proposition 6). We allow smaller regions at high noise and larger regions as noise decreases. The cap interpolates between S_{noisy} and S_{clean} cells as the normalized noise level $\rho_{i,t} \in [0, 1]$ decreases from one:

$$S_{i,t}^{\max} = \text{round}(S_{\text{noisy}} \rho_{i,t} + S_{\text{clean}}(1 - \rho_{i,t})) \,. \tag{29}$$

The configured endpoints are given in Appendix B.1. The noise level is normalized using the scheduler's sigma sequence, including its terminal entry when present. If this sequence is unavailable or constant, we use $\rho = 1 - j/(N-1)$, where j is the sampling index starting at zero and N is the number of sampling steps; for $N = 1$, we use $\rho = 1$. The schedule specifies an upper bound: growth can stop before reaching the cap, and the last sampled step need not attain the low noise endpoint. Attention changes across denoising steps, so the realized regions need not grow monotonically or remain nested.

Storing the cover. The cover is stored with each rollout transition as index lists and per cell coverage counts and is never recomputed from the current policy. Gradients do not pass through attention capture or region construction. Storing at most S_{clean} cells per seed costs $O(S_{\text{clean}} T N_a)$ indices per sample. Replay adds region gathers and reductions without an attention capture or region construction graph.

I.1 Properties of the builder

Proposition 6 (The builder returns a deterministic cover of connected regions)**.** *Assume every cell feature is nonzero and $S^{\max} \geq 1$. The fixed-seed builder has the following properties:*

1. *each $\mathcal{R}_j$ contains its seed, is eight-connected, and satisfies $1 \leq |\mathcal{R}_j| \leq S^{\max}$;*
2. *the seeded regions cover the grid, so $n(q) \geq 1$ for every cell;*
3. *the similarity and tie-breaking rules determine one ordered cover without randomness;*
4. *rescaling each cell's feature vector by a positive number, or applying one orthogonal transformation to all of them, leaves the cover unchanged; and*

5. *for a fixed feature field, the region produced with a smaller area cap is a prefix of the region produced with a larger cap.*

Corollary 2 (Small attention changes leave the cover unchanged)**.** *If every similarity ordering and eligibility comparison visited by the builder has a nonzero margin, then the ordered cover is unchanged in a neighborhood of the normalized attention feature field.*

What it shows. Properties 1 and 2 are what the analysis needs: each region is one connected piece of the grid, and every cell has coverage at least one, so inverse coverage is defined. Property 3 makes the stored cover reproducible from the stored attention. Property 4 says that a cell is grouped by the pattern of its attention over the prompt tokens, not by its total amount of attention. Property 5 says that the noise-dependent area cap only stops growth earlier: at a noisier step, a region is the beginning of the region that the same attention would produce under a larger cap. The corollary says that the cover does not change under small perturbations of the attention unless one of the builder's comparisons is tied.

What it does not show. None of these properties says that a region matches an object; they hold for any attention field. Tie breaking fixes the output at a tie but does not make it stable there, and the prefix property compares caps under one feature field, so regions at different denoising steps need not be nested.

Proof. The seed initializes its own region, and every later cell is added through an eight-connected edge from an accepted parent, so induction gives seed inclusion, connectivity, and the area bound. Since every cell also acts as a seed, the union covers the grid and every multiplicity is positive. For fixed cosine scores, the eligibility test and the lexicographic edge order select at most one next edge, which proves determinism. Positive cellwise rescaling and a common orthogonal transformation preserve all cosine scores, hence every threshold, edge order, and decision. For a fixed feature field, the area cap only terminates an otherwise identical sequence of growth decisions, so the smaller-cap region is a prefix of the larger-cap region. Finally, the builder performs finitely many comparisons of continuous cosine scores and finite averages; when every visited comparison has a nonzero margin, the smallest margin is positive and all decisions persist under a sufficiently small perturbation.

J Implementation Details

This appendix details the ratio computation, reference regularizer, training procedure, and latent geometry used by FlowSAPO. All training configurations and evaluation protocols are collected in Appendix B.

J.1 Computing the ratios

The coordinate change of Equation (2) is computed as

$$\Delta\ell_{i,t,d} = \frac{-(a_{i,t,d} - \mu_{\theta,i,t,d})^2 + (a_{i,t,d} - \mu_{\mathrm{old},i,t,d})^2}{2\sigma_{\mathrm{tr},i,t}^2}, \tag{30}$$

with the stored action, the rollout-policy mean, and the variance held fixed, so that gradients flow only through μ_θ. Our FlowGRPO and FlowSAPO variants use the SDE sampler. Its transition variance is $\sigma_{\mathrm{tr},i,t}^2 = \mathrm{std}_{i,t}^2(-\Delta\sigma_t)$, including the scheduler timestep factor. Here μ_θ is the sampler transition mean rather than the raw denoiser output. The Gaussian normalization constants cancel because both policies share the stored variance, and the computation is carried out in FP32. Both branches pass the exponent through a numerical guard,

$$G_M(x) = \exp(\mathrm{clip}(x, -M, M)), \qquad M = 20, \tag{31}$$

applied to $\delta_{i,t,p}$ in the local branch and to $s_{i,t,j}$ in the regional branch. The guard prevents overflow; the exact density-ratio statements refer to the exponentials before it.

J.2 Reference regularizer

For SD3.5 Medium, disabling the LoRA adapter gives the frozen reference mean $\mu_{\mathrm{ref},t}$, and the regularizer is

$$\mathcal{L}_{\mathrm{ref}} = \mathrm{mean}_{i,t,d} \frac{(\mu_{\theta,i,t,d} - \mu_{\mathrm{ref},i,t,d})^2}{2\,\mathrm{std}_{i,t}^2}, \tag{32}$$

with task specific coefficients given in Appendix B.1. Here $\mathrm{std}_{i,t}$ is the sampler's noise coefficient before the timestep factor $-\Delta\sigma_t$ that enters the transition variance, so Equation (32) is a mean squared proxy rather than an exact transition KL. The Wan implementation includes the sampler's returned timestep factor in the denominator. In each case, the regularizer compares the current policy with the frozen reference, whereas the policy ratios compare the current and rollout policies.

J.3 Training procedure

Algorithm 3 gives the complete training procedure, incorporating the region construction in Algorithm 1.

Algorithm 3 FlowSAPO training procedure.

1. Roll out the stochastic sampler with the rollout policy and store, for every sampled step, the state, the action, the timestep, the rollout-policy mean, and the transition variance; record the text attention of the conditional branch at every sampled step.
2. Form one feature per attention cell from attention to prompt tokens. For SD3.5 Medium and FLUX, retain the text columns of joint attention normalized over image and text keys. For Wan, use cross attention normalized over text keys. Average over heads and the selected layers, then set padded prompt entries to zero without renormalizing.
3. Build the region of every seed (Algorithm 2), compute the coverage, and store both with the rollout.
4. Score the samples, compute the advantages of Equation (4) over the rollout pool, and drop samples with zero advantage from every stored tensor alike.
5. At each replay epoch, shuffle all stored tensors with one common permutation and split them into the same microbatches; the regions are never recomputed.
6. For each trained step, recompute μ_θ, compute the coordinate changes once, and form from them both the local log ratios and the cell scores.
7. Apply inverse coverage and the area dependent width, evaluate the guarded PPO loss of both branches, mix them with the configured weights $1 - \lambda$ and λ (Appendix B.1), and add the reference regularizer.
8. Accumulate gradients over the configured samples and steps; at each optimizer step, apply the gradient-norm cap, update with AdamW, and update the EMA.

To keep equal batch sizes across workers, a few zero-advantage samples may be reinstated after step 4; they contribute nothing to either policy branch.

J.4 Latent and attention grids

For both image backbones, a 512×512 training image has a latent grid of $16 \times 64 \times 64$, so $C = 16$ and $N_f = 4096$. Each attention cell covers 2×2 latent positions, giving $b = 4$ and $N_a = 1024$. FLUX packs these blocks into tokens with 64 channels for the transformer; we unpack the action and both transition means before computing the positionwise ratios, so the local factor still contains the 16 channels of one latent position.

For Wan, 33 training frames at 240×416 give a latent tensor of $16 \times 9 \times 30 \times 52$. The transformer uses patches of $1 \times 2 \times 2$, giving an attention grid of $9 \times 15 \times 26$, with $N_f = 14040$, $N_a = 3510$, and $b = 4$. Region construction runs independently within each latent frame. These are training geometries; evaluation uses the resolutions and frame counts specified in Appendix B.3.

K PROOFS

This appendix proves the results of Appendix H in the order in which they appear there. Each subsection first says in one sentence what it proves and then gives the proof; where Appendix H quotes a number for SD3.5, the subsection also shows how it is computed, with $k = 0.005$, $C = 16$, $N_f = 4096$, and $\epsilon_f = 0.1$. The properties of the region builder are proved next to its construction (Appendix I.1).

K.1 PROPOSITION 1: ONLY THE TOTAL CHANGE REACHES THE GLOBAL WEIGHT

What is proved. In log space, the joint ratio, the global statistic, and every local factor are Gaussian, and the global statistic depends on the per-position divergences only through their total.

Proof. Let $\Delta = \mu - \mu_0$ and write $a = \mu_0 + \sigma\xi$ with $\xi \sim \mathcal{N}(0, I)$. Then

$$\log r(a) = -\frac{\|a-\mu\|^2 - \|a-\mu_0\|^2}{2\sigma^2} = \frac{\Delta^\top \xi}{\sigma} - \frac{\|\Delta\|^2}{2\sigma^2}, \tag{33}$$

which is Gaussian with mean $-K$ and variance $2K$ because $K = \|\Delta\|^2/(2\sigma^2)$. Applied to the complete transition this is the first law in Equation (16); applied to the action factor at position p, with K_p in place of K, it is the third, and the factors are independent because the covariance is diagonal. Hence $\sum_p \delta_p \sim \mathcal{N}(-K, 2K)$ and $\log r^{\mathrm{geo}} = D^{-1}\sum_p \delta_p \sim \mathcal{N}(-K/D, 2K/D^2)$, whose law involves $(K_p)_p$ only through K. The Gaussian moment-generating function gives $\mathbb{E}[r^q] = \exp((q^2-q)K)$: $q = 1$ gives mean one for the joint ratio and for every local factor, $q = 2$ gives $\mathrm{Var}(r^{\mathrm{joint}}) = e^{2K} - 1$, and $q = 1/D$ gives $\mathbb{E}[r^{\mathrm{geo}}] = \exp(-K/D + K/D^2)$, which is below one whenever $K > 0$ because $D > 1$.

In the numerical example, $K = kN_f = 20.5$, so $\log r^{\mathrm{joint}}$ has mean -20.5 and standard deviation 6.4 and the joint weight lies in $[0.9, 1.1]$ with probability 7.5×10^{-5}, whereas a local factor lies in the same band with probability 0.68.

K.2 PROPOSITION 2 AND COROLLARY 1: CANCELLATION AND ONE DECISION PER DIVERGENCE

What is proved. With the same divergence at every position, the local changes largely cancel in the global statistic, the largest local change exceeds the global one by a factor that grows as the step shrinks, and, away from the boundary case, the global clipping decision concentrates as the grid grows while positionwise decisions keep varying.

Proof. Let $K_p = k$ for all p. By Proposition 1 the δ_p are independent and identically distributed as $\mathcal{N}(-k, 2k)$, so $\Pr(\delta_p > 0) = 1 - \Phi(\sqrt{k/2})$, which is 0.48 at $k = 0.005$: increases and decreases are almost equally frequent. By the strong law of large numbers, $N_f^{-1}\sum_p \delta_p \to -k$ and $N_f^{-1}\sum_p |\delta_p| \to m(k)$ almost surely, so the cancellation index converges to $1 - k/m(k)$. For $X \sim \mathcal{N}(\mu, s^2)$, $\mathbb{E}|X| = s\sqrt{2/\pi}\, e^{-\mu^2/(2s^2)} + \mu\big(1 - 2\Phi(-\mu/s)\big)$; with $\mu = -k$ and $s^2 = 2k$ this is $m(k) = 2\sqrt{k/\pi}\, e^{-k/4} + k\big(2\Phi(\sqrt{k/2}) - 1\big)$. Expanding $e^{-k/4} = 1 - k/4 + O(k^2)$ and $2\Phi(\sqrt{k/2}) - 1 = \sqrt{k/\pi} + O(k^{3/2})$ gives $m(k) = 2\sqrt{k/\pi}\,(1 + k/4 + O(k^2))$, hence $1 - k/m(k) = 1 - \frac{1}{2}\sqrt{\pi k} + O(k^{3/2})$; at $k = 0.005$ the exact value and the expansion both give 0.937.

For the maximum, the variables $(\delta_p + k)/\sqrt{2k}$ are independent standard normals, and the maximum of N such variables divided by $\sqrt{2\ln N}$ converges to one in probability. Therefore $\max_p \delta_p = -k + \sqrt{4k \ln N_f}\,(1 + o_p(1))$, while $\log r^{\mathrm{geo}} \to -k/C$ in probability because its standard deviation $\sqrt{2k/N_f}/C$ vanishes; dividing the two gives the stated ratio. The finite grid values need not closely match this asymptotic maximum, so the asymptotic expression is used only to describe its scaling.

For the corollary, part (i) reads the mean $-k/C$ and the standard deviation $\sqrt{2k/N_f}/C$ of $\log r^{\mathrm{geo}}$ off Proposition 1 with $K = kN_f$ and $D = CN_f$; their ratio is $\sqrt{kN_f/2}$. Since $\log r^{\mathrm{geo}} \to -k/C$ in probability, the exit probability converges to $\mathbf{1}\{e^{-k/C} < 1 - \epsilon_g\}$ whenever $e^{-k/C} \neq 1 - \epsilon_g$; the upper boundary is never reached in the limit because $e^{-k/C} < 1$. At equality with the lower

boundary, the Gaussian log weight is centered exactly on that boundary for every N_f, giving a lower tail probability of $1/2$, while the upper tail vanishes. This proves Equation (17). At $\epsilon_g = 10^{-4}$ and $N_f = 4096$, the global weight lies below $1 - \epsilon_g$ with probability $\Phi(2.18) = 0.985$; on a grid sixty-four times larger the standard deviation is eight times smaller and the probability is one to within 10^{-60}. Part (ii) is the Gaussian probability that $\delta_p \notin [\log(1-\epsilon_f), \log(1+\epsilon_f)]$; both tails are positive for every $k > 0$, and at $k = 0.005$ and $\epsilon_f = 0.1$ their sum is 0.32.

K.3 Theorem 1: the local factor as the best one-position approximation

What is proved. Under the assumptions of Theorem 1, a local factor is the conditional expectation of the joint ratio given the action at its own position, and an exact ratio over a larger set of factors is never less variable.

Proof. Fix a state s outside a p_0-null set. Absolute continuity gives $\mathbb{E}_{p_{0,n}}[r_n \mid s] = 1$ for every factor. Since p_0 and p factorize over the partition,

$$\mathbb{E}_{p_0}[r_{\text{joint}} \mid s, a_p] = r_p(a_p \mid s) \prod_{n \neq p} \mathbb{E}_{p_{0,n}}[r_n \mid s] = r_p(a_p \mid s). \tag{34}$$

Conditional expectation is the orthogonal projection of $L^2(p_0(\cdot \mid s))$ onto the $\sigma(s, a_p)$-measurable functions, which proves the optimality and almost-sure uniqueness in part (i). For part (ii), the same computation gives $\mathbb{E}_{p_0}[r_{B'} \mid s, a_B] = r_B$, and the law of total variance gives

$$\mathrm{Var}_{p_0}(r_{B'} \mid s) = \mathrm{Var}_{p_0}(r_B \mid s) + \mathbb{E}_{p_0}\big[\mathrm{Var}_{p_0}(r_{B'} \mid s, a_B) \mid s\big] \geq \mathrm{Var}_{p_0}(r_B \mid s). \tag{35}$$

Expanding the finite product $r_{\text{joint}} = \prod_p (1 + \eta_p)$, multiplying by A, and taking expectations gives Equation (19).

K.4 Proposition 3: the spread of a regional score and the clip width

What is proved. A regional score is Gaussian, its exponential has mean at most one, and its standard deviation grows at most as the square root of the region's area; the subsection also quantifies the clipping boundaries for the training exponent $\nu = 0.7$.

Proof. Each $u(q)$ is the log density ratio of a disjoint Gaussian action block, so $u(q) \sim \mathcal{N}(-K_q, 2K_q)$ under the rollout distribution, independently across cells at the stored state. Since $s_j = \sum_q a_{jq} u(q)$ is a fixed linear combination, it is Gaussian with the mean and variance in Equation (20), and the moment-generating function at one gives the mean of e^{s_j}. Because $0 < a_{jq} \leq 1$ for every cell of the region, each term $a_{jq}(1 - a_{jq})K_q$ is nonnegative, and all of them vanish exactly when $a_{jq} = 1$ wherever $K_q > 0$. If $K_p = k$ at every position, then $K_q = bk$ for every cell and $\mathrm{Var}(s_j) = 2bk \sum_{q \in \mathcal{R}_j} n(q)^{-2} \leq 2bk\,|\mathcal{R}_j| = 2k\,a_j$, with equality exactly when every cell of $\mathcal{R}_j$ has coverage one.

The comparison with the positionwise band uses the ratio $\log(1 + \epsilon_f a^\nu)/(\sqrt{a}\,\log(1+\epsilon_f))$ for the upper side and $\log(1 - \epsilon_f a^\nu)/(\sqrt{a}\,\log(1-\epsilon_f))$ for the lower side when $\epsilon_f a^\nu < 1$. For the reference exponent $\nu = \frac{1}{2}$, these ratios are close to one only while the clipping widths are small. With $\epsilon_f = 0.1$ and $\nu = 0.7$ the upper ratio is 1.23, 1.39, 1.37, and 1.16 at $a = 4$, 16, 64, and 256 latent positions; the lower ratio is 1.45 at $a = 4$ and 2.83 at $a = 16$, and from $a = 27$ on $\epsilon_j \geq 1$, so the lower boundary $1 - \epsilon_j \leq 0$ can no longer be crossed by the positive weight. For $b = 4$, the configured maximum area is $a_{\max} = bS_{\text{clean}} = 256$, giving a clipping width of 4.85. Outside $|s_j| < 20$ the numerical guard has derivative zero, so a region whose score leaves this interval stops contributing a gradient whatever its clip.

K.5 Proposition 4: gradient at rollout and regional coupling

What is proved. At the rollout policy, all three objectives have gradients proportional to one policy gradient, whatever the cover; away from the boundaries, each cell receives the exponential weights and clipping gates of the regions that contain it, divided by its coverage.

Proof of part (i). The three unclipped, unguarded surrogates are

$$\mathcal{J}_{\mathrm{geo}} = \frac{1}{T}\sum_{t}\mathbb{E}_{\mathrm{old}}\big[A\, r_t^{\mathrm{geo}}\big], \qquad \mathcal{J}_{\mathrm{fine}} = \frac{1}{TN_f}\sum_{t,p}\mathbb{E}_{\mathrm{old}}\big[A\, r_{t,p}^{\mathrm{fine}}\big],$$
$$\mathcal{J}_{\mathrm{region}} = \frac{1}{TN_a}\sum_{t,j}\mathbb{E}_{\mathrm{old}}\big[A\, e^{s_{t,j}}\big]. \tag{36}$$

At $\theta = \theta_{\mathrm{old}}$ every weight equals one, so its gradient equals the gradient of its logarithm. With $g_t = \nabla_\theta \log \pi_\theta(a_t \mid s_t) = \sum_d \nabla \Delta\ell_{t,d}$, the logarithm of the global weight has gradient g_t/D, the local log ratios have gradients summing to $\sum_p \nabla\delta_{t,p} = g_t$, and the regional scores have gradients summing to $\sum_j \nabla s_{t,j} = g_t$ by Equation (23), which holds whenever the stored coverage equals the actual multiplicity. Hence $\nabla\mathcal{J}_{\mathrm{geo}} = G/(TD)$, $\nabla\mathcal{J}_{\mathrm{fine}} = G/(TN_f)$, and $\nabla\mathcal{J}_{\mathrm{region}} = G/(TN_a)$ with $G = \mathbb{E}_{\mathrm{old}}[A\sum_t g_t]$, which is Equation (21) because $D/N_f = C$ and $N_f/N_a = b$. At the rollout policy every weight lies strictly inside every band of positive width and inside the guard, where the PPO loss has derivative $-A$ in its weight and the guard has derivative one, so the clipped, guarded losses have the same gradients up to the sign of the PPO loss.

Proof of part (ii). Away from the stated boundaries, the derivative of the signed PPO loss in Equation (5) with respect to its weight is $-AI_j$, the guarded weight has derivative $\bar{w}_j$ for $|s_j| < 20$ and zero outside this interval, and Equation (10) gives $\partial s_j/\partial u(q) = \mathbf{1}\{q \in \mathcal{R}_j\}/n(q)$. The chain rule gives $\mathbf{1}\{q \in \mathcal{R}_j\}\gamma_j/n(q)$ for each region, and averaging over the N_a regions gives Equation (22).

K.6 Proposition 5: the grouping at second order

What is proved. The unclipped regional objective has the same gradient under every cover at the rollout policy, but its Hessian has a term that depends on the cover.

Proof. The chain rule gives $\nabla F = AN_a^{-1}\sum_j e^{s_j}\nabla s_j$ and $\nabla^2 F = AN_a^{-1}\sum_j e^{s_j}(\nabla^2 s_j + \nabla s_j \nabla s_j^\top)$. At rollout, $s_j = 0$. Differentiating Equation (23) gives $\sum_j \nabla s_j = \nabla\sum_p \delta_p$ and $\sum_j \nabla^2 s_j = \nabla^2\sum_p \delta_p$. Substitution yields Equations (24) and (25). The outer product sum still depends on the cover: regional score derivatives $(x, 0)$ and $(x/2, x/2)$ along a direction v have equal sums but contribute x^2 and $x^2/2$ to its quadratic form.