SCOPE: SEMANTIC-BOUNDED CREDIT OPTIMIZATION FOR PROMPT ENGINEERING IN TEXT-TO-IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Training a language model to rewrite prompts can improve text-to-image (T2I) generation while keeping the image generator frozen. In common reinforcement-learning formulations, a single image-level reward produces one trajectory-level advantage that is applied across the tokens of a rewritten prompt. This creates a credit-assignment gap: the units that users compose and that T2I systems must bind—objects, counts, attributes, and relations—are not the units that receive policy-gradient credit. We present **SCOPE**, a GRPO-style objective that uses semantic spans to structure this token-level policy update. For each rewritten prompt, **SCOPE** aligns generated spans with tokenizer positions, averages current-versus-old policy log-probability shifts within each span, and converts these shifts into detached weights normalized by valid-token count. The resulting weights modulate the token-level clipped policy surrogate, while the reward remains at the trajectory level. **SCOPE** therefore does not estimate the causal contribution of individual spans to the image reward; it provides a span-aware reweighting of policy updates. Across Qwen-Image, SD3, and FLUX.1-dev, **SCOPE** improves over vanilla GRPO on GenEval and T2I-CompBench, with mean relative gains of 9.48% and 7.62%, respectively. These results show that span-aware policy-update reweighting improves compositional alignment in the evaluated frozen-renderer settings.

1 INTRODUCTION

Text-to-image (T2I) generation models (Hao et al., 2023; Mañas et al., 2024; Rosenman et al., 2024; Mrini et al., 2024; Chen et al., 2025; Yang et al., 2025b; Wu et al., 2025c;b) have become the default interface for turning natural-language intent into visual content. A user rarely controls the denoising trajectory, attention maps, or decoder parameters directly; the practical control surface is almost always the text prompt. This makes prompt quality a first-order factor: a small change in wording can determine whether the generated image preserves the requested object count, binds a color to the right object, or places two objects in the intended spatial relation. As modern diffusion and flow-based generators grow larger, retraining or even lightly adapting the visual decoder is often expensive, deployment-specific, and risky for image quality. Prompt-side adaptation offers a cheaper alternative: keep the image generator fixed and improve the language interface that feeds it.

This frozen-renderer setting is important for both research and deployment. First, it isolates the language-to-image alignment problem from the already costly problem of optimizing a generative visual backbone. Second, it matches many real systems in which the renderer is a large pretrained model or an external API, while the user-facing layer can still rewrite, expand, or clarify prompts. Third, it is a natural place to use reinforcement learning: the rewriter can propose candidate prompts, the frozen renderer can produce images, and an automatic evaluator can score whether the rendered image satisfies the requested composition. Recent prompt optimization and reprompting methods follow this general recipe, showing that a language-side policy can improve a frozen or mostly fixed

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.

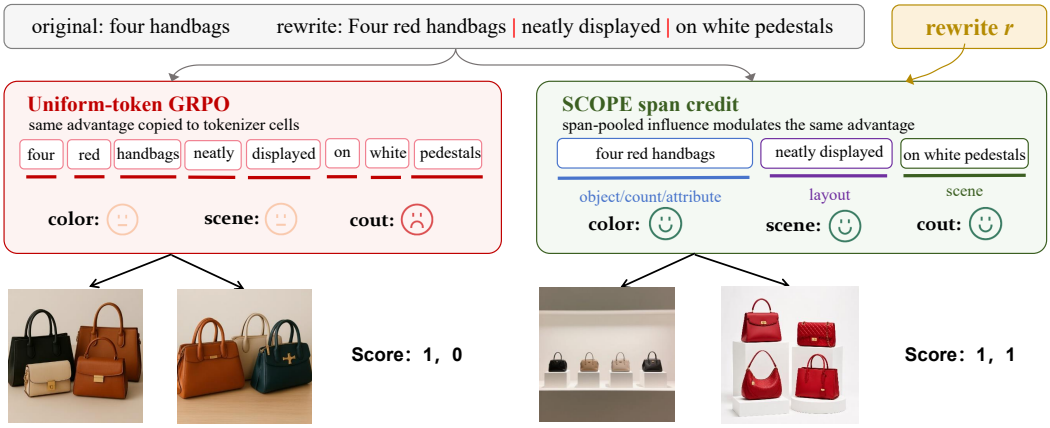

Figure 1: **Motivation.** While a rendered image yields a single scalar reward, prompt edits naturally form semantic phrases (e.g., objects, attributes). Standard GRPO uniformly broadcasts the completion-level signal across fragmented tokenizer cells. In contrast, **SCOPE** uses span-level coefficients to modulate the token loss, so tokens within the same semantic span share a consistent update factor.

T2I stack without modifying decoder weights (Hao et al., 2023; Mañas et al., 2024; Wu et al., 2025c; Wang et al., 2026).

However, this interface introduces a credit-assignment mismatch. The evaluator observes a rendered image and returns a single scalar score for the entire rewritten prompt. The policy update, in contrast, is applied at autoregressive token positions. A good rewrite is naturally organized into semantic phrases–for example "four handbags", "red", "in a neat row", and "on white pedestals"–but the tokenizer may split these phrases into subword cells. Vanilla GRPO supplies one advantage for completion and broadcasts it across all valid tokens. That broadcast is simple and stable, but it treats object phrases, attributes, relations, and scene context as interchangeable carriers of the same signal. For T2I rewriting, this is the wrong inductive bias: as illustrated in Figure 1, the frozen renderer reacts to phrase-level semantics, while the optimizer assigns credit to tokenizer fragments.

We propose **SCOPE** (*Semantic-Bounded Credit Optimization for Prompt Engineering*), a rewriter-only RL objective for frozen T2I stacks. The rollout loop remains the one described above: sample a group of rewritten prompts, render them with the frozen image generator, score the images, and compute one trajectory-level advantage for each completion. The change is deliberately narrower. Before the clipped GRPO/PPO surrogate, **SCOPE** reads the non-empty spans emitted by the same Qwen3-8B rollout, maps each span back to that rollout's tokenizer positions, averages current-versus-old policy log-probability shifts within each span, and normalizes the resulting detached span weights by valid-token count. The loss is still token-dense and still uses the same scalar image reward, but tokens inside the same semantic phrase share a consistent coefficient.

This design avoids two complications that would make prompt-side RL hard to deploy. First, span-local weights are computed from policy shifts, not from rewards: **SCOPE** does not need span-level image scores, counterfactual renders, or a critic that attributes pixels to phrases. Second, the pooled coefficients are entered directly into the standard clipped surrogate. If the answer wrapper is malformed, a span is empty, delimiters are repeated or trailing, tokenizer alignment or cached old logits fail, or the completion is truncated, **SCOPE** sets all span weights to one and exactly recovers uniform-token GRPO. Thus, the extra semantic structure provides a span-level loss-reweighting map rather than a new reward pipeline.

We evaluate the same rewriter and optimizer across three frozen backbones, Qwen-Image, SD3, and FLUX.1-dev, on GenEval (Ghosh et al., 2023) and T2I-CompBench (Huang et al., 2023). Against matched uniform-token GRPO, **SCOPE** improves GenEval averages on all three backbones and improves every T2I-CompBench head. Ablations compare token-level weighting without semantic spans with span-local weighting and show that random or fixed-window boundaries do not match semantically generated spans. Our contributions are listed as follows:

- We identify *credit fragmentation* as a key failure mode in rewriter-only RL for frozen T2I generators: image-level rewards supervise tokenizer fragments even though composition is controlled by phrase-level units.
- We introduce **SCOPE**, a GRPO-compatible objective that keeps one advantage per completion, computes detached, token-count-normalized weights from policy changes local to each tokenizer-grounded semantic span before the clipped surrogate, and uses a hybrid group advantage that compares rewrites to the unedited baseline when verifier scores are weak before switching to clipped within-group normalization to preserve discriminative feedback under sparse GenEval-style scores.
- We report matched experiments on GenEval and T2I-CompBench for Qwen-Image, SD3, and FLUX.1-dev, plus ablations showing that span-level policy-shift pooling contributes to the observed gain.

2 Related Work

Language-model RL and credit assignment. Post-training for language models commonly uses policy-gradient or preference objectives such as PPO (Schulman et al., 2017), DPO (Rafailov et al., 2023), and critic-free group updates such as GRPO (Shao et al., 2024; DeepSeek-AI, 2025). A separate line studies how completion-level feedback should be assigned along a token trajectory, including future-influence objectives (Ma et al., 2026), eligibility-style propagation (Parthasarathi et al., 2025), counterfactual or tree-structured credit (Tran et al., 2025; Khandoga et al., 2026), and token-level verifiable rewards (Xie et al., 2025). **SCOPE** uses the same clipped PPO/GRPO machinery, but in a multimodal prompt-rewriting loop where the reward is produced after rendering an image. Because each rollout still yields only one image-level scalar, the question is not how to obtain a new reward, but how to modulate token losses using the existing completion-level advantage over tokens that encode semantic edits.

Text-to-image generators. Modern T2I systems build on denoising diffusion (Ho et al., 2020; Dhariwal & Nichol, 2021), latent diffusion (Rombach et al., 2022), CLIP-style alignment (Radford et al., 2021), and increasingly flow-matching or rectified-flow decoders (Lipman et al., 2022; Esser et al., 2024). Systems such as DALL-E 2 (Ramesh et al., 2022), Imagen (Saharia et al., 2022), DALL-E 3 (Betker et al., 2023), SDXL (Podell et al., 2024), and Qwen-Image (Wu et al., 2025a) have made open-vocabulary image generation highly capable, but they still depend on the conditioning text for compositional details. Sampling and conditioning tools–DDIM (Song et al., 2021), DPM-Solver (Lu et al., 2022), EDM (Karras et al., 2022), diffusion transformers (Peebles & Xie, 2023), classifier-free guidance (Ho & Salimans, 2022), Prompt-to-Prompt (Hertz et al., 2023), InstructPix2Pix (Brooks et al., 2023), and ControlNet (Zhang et al., 2023)–improve controllability from the generator side. Our setting is complementary: the generator is frozen and the trainable component is the language-side rewriter.

Prompt-side alignment for T2I. T2I alignment methods include preference-data and reward-model pipelines (Kirstain et al., 2023; Xu et al., 2023; Wu et al., 2023; Lee et al., 2023), input-side or adapter-based control (Gal et al., 2023; Ruiz et al., 2023; Zhang et al., 2023), and prompt-centric optimization (Hao et al., 2023; Mañas et al., 2024; Rosenman et al., 2024; Mrini et al., 2024; Chen et al., 2025; Yang et al., 2025b). Compositional benchmarks such as Attend-and-Excite prompts, GenEval, and T2I-CompBench expose failures in binding, counting, attributes, and spatial relations (Chefer et al., 2023; Ghosh et al., 2023; Huang et al., 2023). RL can also target the generator directly, e.g., denoising-policy gradients, diffusion DPO, or online RL for flow models (Black et al., 2024; Wallace et al., 2024; Liu et al., 2025). RePrompt (Wu et al., 2025c) is closest to our setting because it freezes a T2I stack and trains a rewriter with GRPO-style feedback, while PromptRL (Wang et al., 2026) couples prompt variants with flow-model adaptation. **SCOPE** keeps the RePrompt-style split–trainable rewriter, frozen renderer–and changes only how grouped trajectory advantages are converted into token losses.

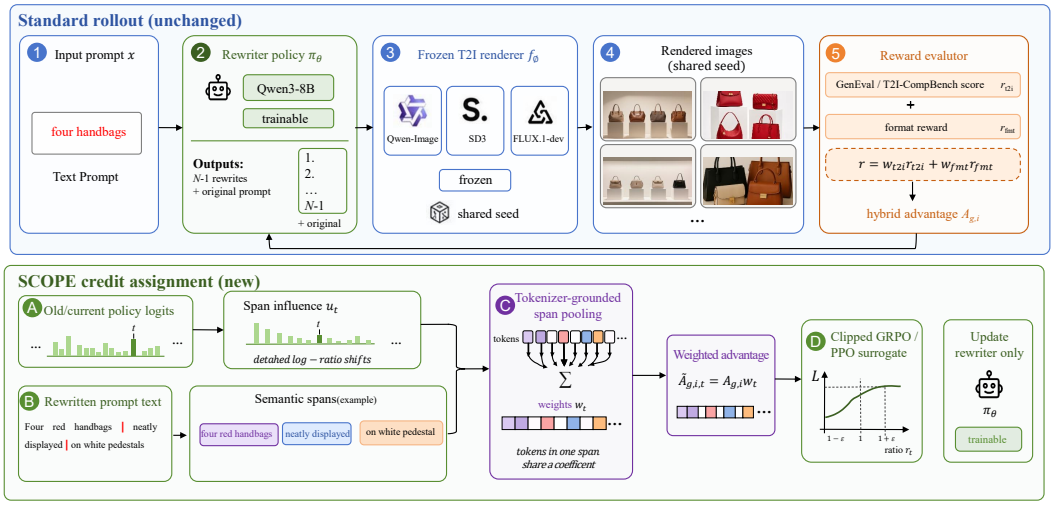

Figure 2: **SCOPE** training pipeline. Qwen3-8B jointly emits a rewritten answer and non-empty semantic spans as <answer>span 1 | span 2 | span 3</answer>; | is only an internal boundary and is never a stop token. The rollout removes all delimiters and concatenates the spans before the frozen T2I backbone renders an image, so the image reward uses the cleaned prompt while the format reward checks the structured output. The generated spans are aligned to this rollout's token positions. Current-versus-old policy shifts are averaged within each span, clipped, normalized by valid-token count, detached, and used in the existing clipped GRPO surrogate. Malformed, empty, repeated, truncated, or misaligned cases use uniform-token GRPO.

3 METHODOLOGY

As discussed in Section 1, prompt-side RL for frozen T2I systems has an asymmetric interface: the image generator and evaluator return a scalar judgment for an entire rewrite, while the language policy is updated token by token. **SCOPE** keeps the useful parts of this interface unchanged. For an input prompt x, an autoregressive rewriter π_θ proposes a structured completion y. A cleaning operator $c(\cdot)$ extracts the <answer> field, removes every |, and concatenates the span text; the frozen renderer then produces $I = f_\phi(c(y))$, and the image evaluator returns a reward based on the cleaned prompt and image. We do not update f_ϕ, do not introduce span-level image rewards, and do not change the PPO/GRPO clipping rule. Instead, **SCOPE** changes the map between these two levels: it converts a completion-level advantage into span-modulated token losses, with tokens in the same semantic span sharing a coefficient.

The method proceeds in the same order as the training loop in Figure 2. First, Section 3.1 defines grouped rollouts and a hybrid advantage, giving one reliable scalar signal per sampled rewrite. Second, Section 3.2 recalls the ordinary GRPO surrogate and pinpoints where uniform token credit fragments semantic phrases. Third, Section 3.3 averages policy log-probability shifts inside tokenizer-grounded spans, normalizes detached span weights, and plugs the resulting coefficients into the clipped surrogate. The final objective is therefore a strict extension of GRPO: if the span weights are all one, the loss reduces to the uniform-token baseline.

3.1 ROLLOUTS AND ADVANTAGES

We view prompt rewriting as a one-step decision problem in which the action is the entire rewritten prompt. The population objective is:

$$\max_\theta \; \mathbb{E}_{x\sim\mathcal{D}}\, \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\big[r(x, c(y))\big]. \tag{1}$$

In practice, for a minibatch prompt x_g, the trainer samples $N-1$ rewrites $\{y_{g,i}\}_{i=1}^{N-1}$ from π_θ and appends the unedited prompt as $y_{g,N}$. The unedited member is not optimized; it gives a within-prompt reference rendered under the same group seed. For this baseline member, $c(y_{g,N})$ is the

original prompt itself. Each completion receives a scalar reward:

$$r_{g,i} = w_{\mathrm{t2i}}\, r_{\mathrm{t2i}}(x_g, c(y_{g,i})) + w_{\mathrm{fmt}}\, r_{\mathrm{fmt}}(y_{g,i}), \tag{2}$$

where r_{t2i} scores the rendered image and r_{fmt} penalizes malformed rewriter outputs. When f_ϕ is stochastic, all members of a group share one latent seed, so reward differences primarily reflect prompt edits rather than independent renderer noise.

Given these group scores, the next question is how to form a completion-level learning signal before distributing it over tokens. A pure group-normalized GRPO advantage is effective once the baseline prompt is already reasonable, but it can be brittle when sparse image verifiers give many nearly tied scores. We therefore use a hybrid rule. Let $b_g = r_{g,N}$ be the unedited baseline, $\mu_g = (N-1)^{-1}\sum_{j=1}^{N-1} r_{g,j}$ the mean sampled-rewrite score, and $\hat{\sigma}_g = \big((N-1)^{-1}\sum_{j=1}^{N-1}(r_{g,j}-\mu_g)^2\big)^{1/2}$. For $i < N$, clip value c, numerical constant ϵ, and threshold τ,

$$A_{g,i} = \begin{cases} \mathrm{clip}(r_{g,i} - b_g,\, -c,\, c), & b_g < \tau, \\ \mathrm{clip}\left(\dfrac{r_{g,i}-\mu_g}{\hat{\sigma}_g + \epsilon},\, -c,\, c\right), & b_g \geq \tau, \end{cases} \qquad A_{g,N} = 0. \tag{3}$$

The second branch is the usual group-relative centering and scaling used by GRPO-style objectives (Shao et al., 2024; Liu et al., 2025). The first branch compares directly to the unedited prompt when b_g is weak, preserving the useful fact that a rewrite improves the original text even if sampled rewrites look nearly tied after normalization. At this point, every sampled rewrite has exactly one scalar advantage; no span information has entered yet.

3.2 Uniform GRPO

Having defined the scalar signal, we now recall how the baseline uses it. For a sampled completion $y_{g,i} = (y_{g,i,1}, \ldots, y_{g,i,T})$, let $\mathcal{V}_{g,i}$ be the valid generated-token positions and let π_{old} be the detached policy snapshot that produced the rollout batch. The ordinary per-token importance ratio is:

$$\rho_{g,i,t} = \frac{\pi_\theta(y_{g,i,t} \mid x_g, y_{g,i,<t})}{\pi_{\mathrm{old}}(y_{g,i,t} \mid x_g, y_{g,i,<t})}, \qquad t \in \mathcal{V}_{g,i}. \tag{4}$$

Uniform-token GRPO broadcasts the same $A_{g,i}$ to every valid position and minimizes the negative clipped surrogate:

$$\ell_{g,i}^{\mathrm{uni}} = -\sum_{t \in \mathcal{V}_{g,i}} \min(\rho_{g,i,t} A_{g,i},\ \mathrm{clip}(\rho_{g,i,t}, 1-\epsilon_\ell, 1+\epsilon_h) A_{g,i})\,. \tag{5}$$

Equation (5) is the stability anchor that **SCOPE** preserves: the ratio, the clipping interval, and the completion-level reward are unchanged. Its weakness is semantic. If the tokenizer splits "four handbags" or "white pedestal" into several cells, Eq. (5) treats those cells as independent recipients of the same scalar feedback. This is exactly the mismatch from Figure 1: the image renderer responds to phrase-level content, but uniform GRPO assigns gradients over arbitrary token fragments. Thus, the remaining design question is not how to create a new reward, but how to modulate token losses with the existing advantage through an adaptive span-level reweighting map.

3.3 Span-pooled Credit

The same trainable Qwen3-8B rollout emits contiguous spans separated by |; we parse non-empty spans from the <answer> field and align each one to that rollout's tokenizer positions. Before rendering, all delimiters are removed and the pieces are joined as natural-language text. Let $\mathcal{T}_{g,i,s}$ be the valid generated content-token positions aligned to span s. These sets must be non-empty, disjoint, and cover all valid answer-content tokens. Delimiter and wrapper positions belong to no semantic span.

For each aligned content token, the policy log-probability shift is

$$\Delta_{g,i,t} = \log \pi_\theta(y_{g,i,t} \mid x_g, y_{g,i,<t}) - \log \pi_{\mathrm{old}}(y_{g,i,t} \mid x_g, y_{g,i,<t}). \tag{6}$$

The span-local mean and bounded raw weight are

$$b_{g,i,s} = \frac{1}{|\mathcal{T}_{g,i,s}|} \sum_{t \in \mathcal{T}_{g,i,s}} \Delta_{g,i,t}, \qquad v_{g,i,s} = \mathrm{clip}(\exp(\beta b_{g,i,s}), 0.8, 1.2)\,. \tag{7}$$

Here β controls sensitivity to policy movement within the same span. The required log-probabilities are already computed for the importance ratio; no future-token accumulation, additional model call, or additional render is needed. The image reward does not enter Eq. (7).

We normalize by the number of valid tokens in each span so the token-weighted mean over answer content is exactly one:

$$w_{g,i,s} = \frac{v_{g,i,s}}{\frac{\sum_r |\mathcal{T}_{g,i,r}| v_{g,i,r}}{\sum_r |\mathcal{T}_{g,i,r}|}}, \qquad \widetilde{A}_{g,i,t} = A_{g,i}\ \mathrm{stopgrad}(w_{g,i,s}), \quad t \in \mathcal{T}_{g,i,s}. \tag{8}$$

The entire weight branch, including aggregation, clipping, and normalization, is detached during the policy update. Delimiter and wrapper positions receive no span weight; if retained by the existing completion-loss mask, they keep coefficient one. If spans are absent or empty, delimiters are consecutive or trailing, the completion is truncated, content tokens are not fully covered, or boundary-to-token alignment fails, we set every coefficient to one on $\mathcal{V}_{g,i}$ and use uniform-token GRPO for that sample.

The final token advantage is $\widetilde{A}_{g,i,t}$ from Eq. (8) on span tokens and $A_{g,i}$ on any retained non-span positions. The clipped surrogate becomes

$$\ell_{g,i}^{\mathrm{scope}} = -\sum_{t \in \mathcal{V}_{g,i}} \min\Big(\rho_{g,i,t}\widetilde{A}_{g,i,t},\ \mathrm{clip}(\rho_{g,i,t}, 1-\epsilon_\ell, 1+\epsilon_h)\widetilde{A}_{g,i,t}\Big)\,. \tag{9}$$

The policy loss averages over sampled rewrites and minibatch prompts,

$$\mathcal{L}_{\mathrm{pol}} = \frac{1}{|\mathcal{B}|} \sum_{g \in \mathcal{B}} \frac{1}{N-1} \sum_{i=1}^{N-1} \ell_{g,i}^{\mathrm{scope}}, \tag{10}$$

with an optional KL penalty to a frozen reference policy, as in common RL practice (Liu et al., 2025; Xue et al., 2025).

Properties. This construction gives three useful consequences. First, it is a strict extension of GRPO: if all token coefficients are one, Eq. (9) reduces to Eq. (5). Second, it does not claim span-level reward attribution. The evaluator still produces one image-level scalar; span-local policy shifts only modulate where the completion-level advantage enters the token surrogate. Third, it matches the compositional interface of prompt rewriting: objects, attributes, relations, and scene phrases receive internally consistent coefficients, so gradient mass is less likely to scatter across arbitrary subword fragments while the renderer and reward model remain frozen.

4 Experiments

4.1 Experiment Setup

Questions. The experiments test three questions. First, does span-level policy-shift reweighting improve a rewriter beyond uniform-token GRPO when the renderer is frozen? Second, do the gains transfer across frozen image backbones rather than fitting one generator's quirks? Third, which part of the credit map matters: span-local policy changes, semantic span boundaries, or the hybrid advantage rule?

Protocol. The image generator f_ϕ is frozen in every experiment; only the rewriter π_θ is trained. We evaluate compositional alignment with GenEval (Ghosh et al., 2023) and T2I-CompBench (Huang et al., 2023). GenEval reports one-object, two-object, counting, color, position, and attribute heads plus the official average. T2I-CompBench reports color, shape, texture, spatial, numeracy, and complex heads. We use three frozen renderers: Qwen-Image (Wu et al., 2025a), SD3, and FLUX.1-dev. For each backbone, renderer resolution, denoising steps, and guidance scale are fixed across all

Table 1: **GenEval across three frozen image backbones. SCOPE** attains the best Overall score on every backbone, improving the strongest competing method by +0.04, +0.09, and +0.08 on Qwen-Image, SD3, and FLUX.1-dev respectively. Gains are largest on **Pos.** and **Cnt.**, the categories on which the unprompted backbones score worst. **Bold** marks the column maximum within each backbone. RePrompt† on Qwen-Image is our reproduction; SD3 / FLUX.1-dev RePrompt rows follow Wu et al. (2025c). Single / Two: one- and two-object accuracy; Cnt., Clr., Pos., Attr.: counting, colors, position, attribute binding.

Backbone	Method	Single	Two	Cnt.	Clr.	Pos.	Attr.	Overall↑
Qwen-Image	Baseline	0.99	0.95	0.87	0.89	0.79	0.75	0.87
	Training-free (z-image)	0.95	0.89	0.69	0.82	0.51	0.73	0.76
	GRPO	0.99	0.97	0.90	**0.94**	0.87	0.80	0.91
	RePrompt† (Wu et al., 2025c)	0.98	0.87	0.77	0.85	0.62	0.49	0.76
	SCOPE (ours)	**1.00**	**0.99**	**0.94**	0.93	**0.95**	**0.88**	**0.95**
SD3	Baseline	0.98	0.78	0.44	0.83	0.23	0.50	0.63
	Training-free (z-image)	0.58	0.54	0.37	0.42	0.25	0.37	0.42
	GRPO	0.98	0.87	0.58	0.85	0.57	0.60	0.74
	RePrompt (Wu et al., 2025c)	0.99	0.86	0.60	0.86	0.59	0.60	0.75
	SCOPE (ours)	**1.00**	**0.90**	**0.79**	**0.89**	**0.83**	**0.63**	**0.84**
FLUX.1-dev	Baseline	0.99	0.80	0.72	0.80	0.16	0.46	0.65
	Training-free (z-image)	0.53	0.50	0.56	0.42	0.30	0.36	0.44
	GRPO	0.98	0.85	0.72	0.82	0.62	0.56	0.76
	RePrompt (Wu et al., 2025c)	0.98	0.87	0.77	0.85	0.62	0.49	0.76
	SCOPE (ours)	**1.00**	**0.89**	**0.83**	**0.89**	**0.84**	**0.60**	**0.84**

methods (Table 6). Within a GRPO group, all $N = 8$ completions share the same rendering seed so that group rewards compare rewrites under matched renderer noise; evaluation uses fresh seeds across prompts and passes.

Training and baselines. The trainable rewriter is Qwen3-8B (Yang et al., 2025a). We compare **SCOPE** with uniform-token GRPO under each frozen backbone; tables also include the unmodified renderer, a training-free z-image rewriter, and RePrompt (Wu et al., 2025c) where applicable. Training settings and data are given in Appendix A.2; RePrompt score provenance is explained in Appendix A.15.

4.2 GenEval

Table 1 compares **SCOPE** with matched GRPO across the three frozen image backbones.

Across the three backbones, **SCOPE** improves the GenEval average over matched GRPO on every frozen renderer. The gains are +0.04 on Qwen-Image, +0.10 on SD3, and +0.08 on FLUX.1-dev, for an average absolute improvement of +0.073 across the three backbones. The improvement is not tied to a single renderer: Qwen-Image moves from 0.91 to 0.95, SD3 from 0.74 to 0.84, and FLUX.1-dev from 0.76 to 0.84. Head-level changes are consistent with improved span-level policy-shift reweighting of compositional prompt edits. Relative to GRPO, **SCOPE** improves counting by +0.04, +0.21, and +0.11 and position by +0.08, +0.26, and +0.22 on Qwen-Image, SD3, and FLUX.1-dev, respectively. All other heads also improve except Qwen-Image color, which changes from 0.94 to 0.93. The largest gains are in position and attribute binding on Qwen-Image, position and counting on SD3, and position on FLUX.1-dev. This pattern is consistent with a loss-reweighting improvement rather than a change to the frozen renderers' visual capacity.

4.3 T2I-CompBench

Table 2 evaluates the same trained rewriters on T2I-CompBench. This benchmark complements GenEval by using different heads and including a bundled complex-composition score. Against GRPO, **SCOPE** improves the reported six-head average by +0.0418 on Qwen-Image, +0.0457 on SD3, and +0.0447 on FLUX.1-dev. Every listed T2I-CompBench head improves over GRPO on all three backbones. The largest Qwen-Image changes are in complex composition (+0.0552) and

spatial alignment (+0.0556); SD3 gains are largest in spatial (+0.0815) and numeracy (+0.0686); and FLUX.1-dev gains are largest in numeracy (+0.0658). Across the six heads—color, shape, texture, spatial, numeracy, and complex—these results show that the same span-local policy-shift reweighting transfers across multiple composition-oriented evaluators while the renderers remain frozen.

4.4 Diagnostics

Figure 3 combines the ablations and the training curve into one compact block, avoiding the wide whitespace that appears when each small diagnostic is floated separately.

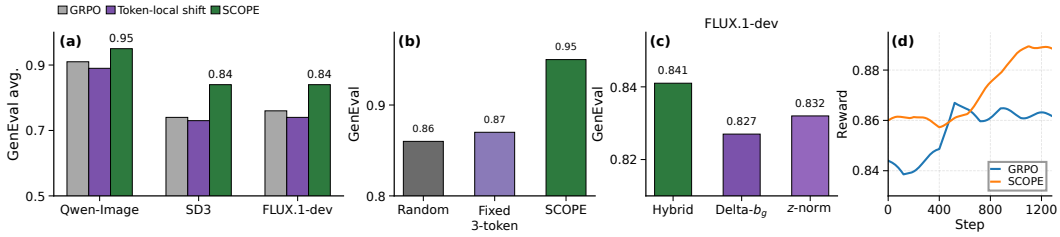

Figure 3: Compact diagnostics. (a) Span-local policy-shift weighting in **SCOPE** improves over uniform GRPO and token-local weighting without span pooling on all three backbones. (b) SCOPE's jointly generated span boundaries yield higher benchmark scores than random or fixed-window boundaries. (c) The hybrid advantage rule is better than either branch alone. (d) The Qwen-Image reward curve separates late in training.

Panels (a)–(d) compare weighting, span-boundary, advantage, and training-dynamics choices. Panel (a) reports GenEval averages of 0.95/0.84/0.84 for **SCOPE**, 0.89/0.73/0.74 for token-local weighting, and 0.91/0.74/0.76 for GRPO; panel (b) compares SCOPE (0.95), random (0.86), and fixed-window (0.87) spans; panel (c) reports 0.841/0.827/0.832 for the three advantage rules on FLUX.1-dev.

Table 2: T2I-CompBench scores across frozen backbones. RePrompt† on Qwen-Image is our reproduction; SD3/FLUX RePrompt rows follow RePrompt (Wu et al., 2025c).

Backbone	Method	Color	Shape	Texture	Spatial	Numeracy	Complex	Avg
Qwen-Image	Baseline	0.8352	0.5965	0.7598	0.4452	0.7340	0.3971	0.6280
	Training-free (z-image)	0.8324	0.5930	0.7285	0.3880	0.6963	0.3922	0.6051
	GRPO	0.8344	0.6042	0.7366	0.4487	0.7321	0.3921	0.6247
	RePrompt (Wu et al., 2025c)†	0.8337	0.6110	0.7546	0.4268	0.7332	0.3882	0.6246
	SCOPE (Ours)	**0.8633**	**0.6273**	**0.7857**	**0.5043**	**0.7713**	**0.4473**	**0.6665**
SD3	Baseline	0.7886	0.5684	0.7267	0.2299	0.5761	0.3754	0.5442
	Training-free (z-image)	0.7933	0.5846	0.7141	0.2977	0.6286	0.3876	0.5677
	GRPO	0.7894	0.5742	0.7196	0.3312	0.6277	0.3802	0.5704
	RePrompt (Wu et al., 2025c)	0.7866	0.5891	0.7184	0.3315	0.6289	0.3744	0.5715
	SCOPE (Ours)	**0.8213**	**0.6147**	**0.7523**	**0.4127**	**0.6963**	**0.3990**	**0.6161**
FLUX.1-dev	Baseline	0.7414	0.4764	0.5937	0.2505	0.6099	0.3712	0.5072
	Training-free (z-image)	0.7624	0.5363	0.6396	0.3162	0.6492	0.3817	0.5476
	GRPO	0.7488	0.5289	0.6143	0.3742	0.6489	0.3742	0.5482
	RePrompt (Wu et al., 2025c)	0.7501	0.5276	0.6515	0.3301	0.6499	0.3721	0.5469
	SCOPE (Ours)	**0.7967**	**0.5677**	**0.6637**	**0.4233**	**0.7147**	**0.3913**	**0.5929**

4.5 Qualitative examples

Figure 4 shows representative prompts where explicit counts, left/right relations, rows, and unusual attributes must be preserved through rewriting. The examples are not used as evidence for the

numerical claim; they illustrate the type of linguistic edits targeted by the span-local reweighting map. The dashed region marks the rewriter-facing columns.

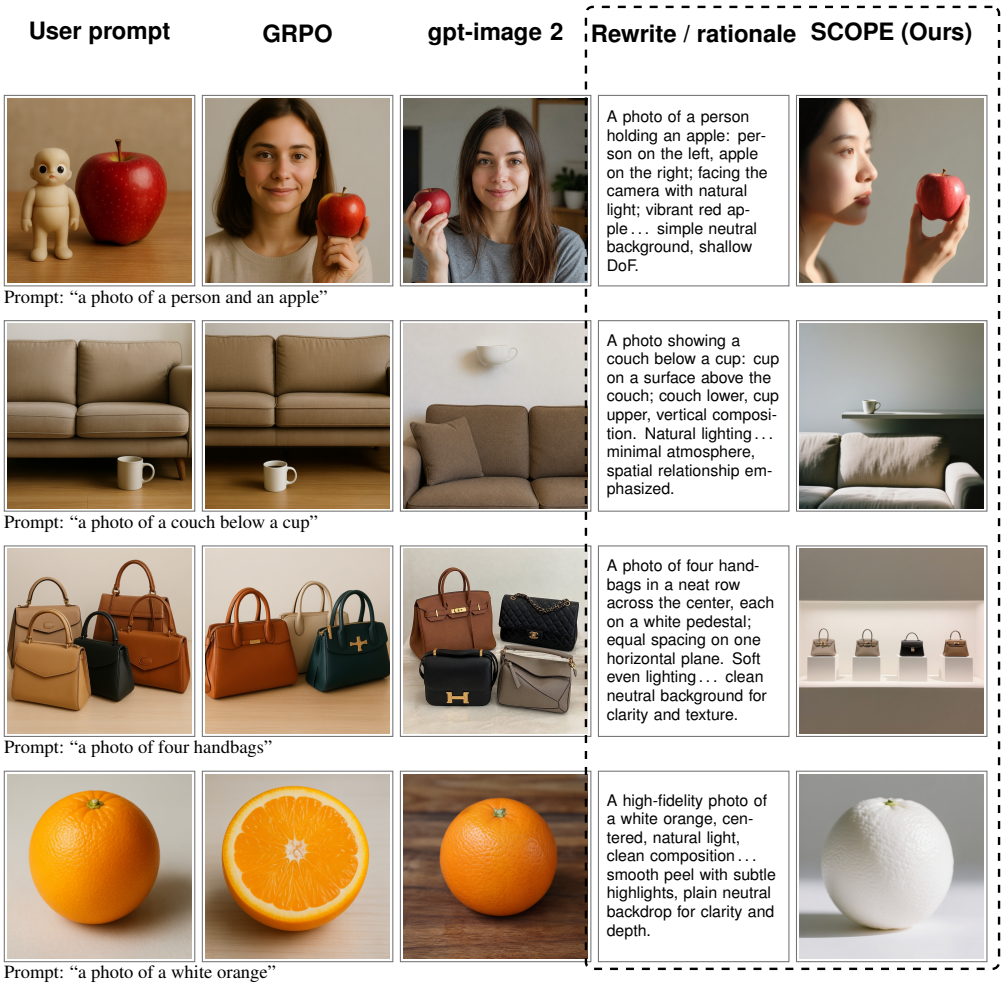

Figure 4: Qualitative comparison on four prompts: user render (Qwen-Image), GRPO, gpt-image 2, rewrite snippet, **SCOPE** (Qwen-Image).

5 Conclusion

We present **SCOPE**, which aims to address a narrow but important bottleneck in RL-based prompt enhancement for frozen T2I models: a scalar image reward is useful, but broadcasting the resulting completion-level advantage uniformly across tokenizer fragments ignores the semantic units that organize compositional edits. The new protocol keeps the standard GRPO loop and renderer unchanged, asks the trainable Qwen3-8B to emit |-delimited spans jointly with each rewrite, removes those delimiters before rendering, and uses detached, normalized span-local policy-shift weights in the clipped surrogate. It adds no span-level image rewards, auxiliary parser, extra renders, or span critic. Across Qwen-Image, SD3, and FLUX.1-dev, **SCOPE** improves GenEval averages over matched GRPO and transfers to every T2I-CompBench head. Taken together, the comparisons support a narrower conclusion: organizing token updates around jointly generated semantic spans is an effective loss-reweighting strategy across frozen renderers.

Limitations and future work. The coefficients are derived from current-versus-old policy log-probability changes, so they should not be interpreted as estimates of each span's independent or causal contribution to the image reward.

AI USE STATEMENT

We used generative AI tools to assist with code writing, data analysis, and figure layout. The authors reviewed the AI-assisted work and take responsibility for the final content of this paper, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, private data, or interventions, and we did not identify any additional ethical issues requiring special review.

REPRODUCIBILITY STATEMENT

The Appendix provides the information needed to reproduce our experiments. Section A.2 lists the datasets and optimization hyperparameters, Section A.5 describes the reward components and disjoint data splits, and Section A.4 specifies the frozen text-to-image backends and sampling settings. The appendix also documents the span extraction, alignment, reward services, and evaluation protocol.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. *OpenAI Technical Report*, 2023. `https://cdn.openai.com/papers/dall-e-3.pdf`.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *ICLR*, 2024.

Manuel Brack, Felix Friedrich, Katharina Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. LEDITS++: Limitless image editing using text-to-image models. In *CVPR*, 2024.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to follow image editing instructions. In *CVPR*, 2023.

Qi Cai, Jingwen Chen, Yang Chen, Yehao Li, Fuchen Long, Yingwei Pan, Zhaofan Qiu, Yiheng Zhang, Fengbin Gao, Peihan Xu, Yimeng Wang, Kai Yu, Wenxuan Chen, Ziwei Feng, Zijian Gong, Jianzhuang Pan, Yi Peng, Rui Tian, Siyu Wang, Bo Zhao, Ting Yao, and Tao Mei. HiDream-I1: A high-efficient image generative foundation model with sparse diffusion transformer. *arXiv preprint arXiv:2505.22705*, 2025.

Tingfeng Cao, Chengyu Wang, Bingyan Liu, Ziheng Wu, Jinhui Zhu, and Jun Huang. BeautifulPrompt: Towards automatic prompt engineering for text-to-image synthesis. *arXiv preprint arXiv:2311.06752*, 2023.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *arXiv preprint arXiv:2301.13826*, 2023. SIGGRAPH 2023.

Anthony Chen, Jianjin Xu, Wenzhao Zheng, Gaole Dai, Yida Wang, Renrui Zhang, Haofan Wang, and Shanghang Zhang. Training-free regional prompting for diffusion transformers. *arXiv preprint arXiv:2411.02395*, 2024a.

Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. In *NeurIPS*, 2023. arXiv:2305.10855.

Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-Σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024b. ECCV 2024.

Ruibo Chen, Jiacheng Pan, Heng Huang, and Zhenheng Yang. Improving text-to-image generation with input-side inference-time scaling. *arXiv preprint arXiv:2510.12041*, 2025.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *ICLR*, 2024. arXiv:2309.17400.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 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. In *NeurIPS*, 2023. arXiv:2305.16381.

Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. *arXiv preprint arXiv:2309.17102*, 2023.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ICLR*, 2023.

Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, Wei Liu, Yichun Shi, Shiqi Sun, Yu Tian, Zhi Tian, Peng Wang, Rui Wang, Xuanda Wang, Xun Wang, Ye Wang, Guofeng Wu, Jie Wu, Xin Xia, Xuefeng Xiao, Zhonghua Zhai, Xinyu Zhang, Qi Zhang, Yuwei Zhang, Shijia Zhao, Jianchao Yang, and Weilin Huang. Seedream 3.0 technical report. *arXiv preprint arXiv:2504.11346*, 2025.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *arXiv preprint arXiv:2310.11513*, 2023.

Jianshu Guo, Wenhao Chai, Jie Deng, Hsiang-Wei Huang, Tian Ye, Yichen Xu, Jiawei Zhang, Jenq-Neng Hwang, and Gaoang Wang. VersaT2I: Improving text-to-image models with versatile reward. *arXiv preprint arXiv:2403.18493*, 2024.

Shashank Gupta, Chaitanya Ahuja, Tsung-Yu Lin, Sreya Dutta Roy, Harrie Oosterhuis, Maarten de Rijke, and Satya Narayan Shukla. A simple and effective reinforcement learning method for text-to-image diffusion fine-tuning. *arXiv preprint arXiv:2503.00897*, 2025.

Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. In *NeurIPS*, 2023.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-Prompt image editing with cross-attention control. In *ICLR*, 2023.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. In *NeurIPS Datasets and Benchmarks*, 2023. Conference version, arXiv:2307.06350v2.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022. NeurIPS 2022.

Mykola Khandoga, Rui Yuan, and Vinay Kumar Sankarapu. Beyond uniform credit: Causal credit assignment for policy optimization. *arXiv preprint arXiv:2602.09331*, 2026.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-Pic: An open dataset of user preferences for text-to-image generation. In *NeurIPS*, 2023.

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. Published at ICLR 2023.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. *NeurIPS*, 2022.

Chiyu Ma, Shuo Yang, Kexin Huang, Jinda Lu, Haoming Meng, Shangshang Wang, Bolin Ding, Soroush Vosoughi, Guoyin Wang, and Jingren Zhou. FIPO: Eliciting deep reasoning with future-KL influenced policy optimization. *arXiv preprint arXiv:2603.19835*, 2026.

Oscar Mañas, Pietro Astolfi, Melissa Hall, Candace Ross, Jack Urbanek, Adina Williams, Aishwarya Agrawal, Adriana Romero-Soriano, and Michal Drozdzal. Improving text-to-image consistency via automatic prompt optimization. *arXiv preprint arXiv:2403.17804*, 2024.

Khalil Mrini, Hanlin Lu, Linjie Yang, Weilin Huang, and Heng Wang. Fast prompt alignment for text-to-image generation. *arXiv preprint arXiv:2412.08639*, 2024.

Prasanna Parthasarathi, Mathieu Reymond, Boxing Chen, Yufei Cui, and Sarath Chandar. GRPO-λ: Credit assignment improves LLM reasoning. *arXiv preprint arXiv:2510.00194*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution text-to-image synthesis. In *ICLR*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Shachar Rosenman, Vasudev Lal, and Phillip Howard. NeuroPrompts: An adaptive framework to optimize prompts for text-to-image generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 159–167, St. Julians, Malta, 2024.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Raphael Gontijo-Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathe-

matical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

Hieu Tran, Zonghai Yao, and Hong Yu. Exploiting tree structure for credit assignment in RL training of LLMs. *arXiv preprint arXiv:2509.18314*, 2025.

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 *CVPR*, 2024.

Fu-Yun Wang, Han Zhang, Michaël Gharbi, Hongsheng Li, and Taesung Park. Promptrl: Prompt matters in rl for flow-based image generation. *arXiv preprint arXiv:2602.01382*, 2026.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Jie Wu, Yu Gao, Zilyu Ye, Ming Li, Liang Li, Hanzhong Guo, Jie Liu, Zeyue Xue, Xiaoxia Hou, Wei Liu, Yan Zeng, and Weilin Huang. RewardDance: Reward scaling in visual generation. *arXiv preprint arXiv:2509.08826*, 2025b.

Mingrui Wu, Lu Wang, Pu Zhao, Fangkai Yang, Jianjin Zhang, Jianfeng Liu, Yuefeng Zhan, Weihao Han, Hao Sun, Jiayi Ji, Xiaoshuai Sun, Qingwei Lin, Weiwei Deng, Dongmei Zhang, Feng Sun, Qi Zhang, and Rongrong Ji. Reprompt: Reasoning-augmented reprompting for text-to-image generation via reinforcement learning. *arXiv preprint arXiv:2505.17540*, 2025c.

Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. *arXiv preprint arXiv:2303.14420*, 2023.

Guofu Xie, Yunsheng Shi, Hongtao Tian, Ting Yao, and Xiao Zhang. CAPO: Towards enhancing LLM reasoning through generative credit assignment. *arXiv preprint arXiv:2508.02298*, 2025.

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. In *NeurIPS*, 2023.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. DanceGRPO: Unleashing GRPO on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

Zhiyuan Yan, Junyan Ye, Weijia Li, Zilong Huang, Shenghai Yuan, Xiangyang He, Kaiqing Lin, Jun He, Conghui He, and Li Yuan. GPT-ImgEval: A comprehensive benchmark for diagnosing GPT4o in image generation. *arXiv preprint arXiv:2504.02782*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Hongji Yang, Yucheng Zhou, Wencheng Han, and Jianbing Shen. Self-rewarding large vision-language models for optimizing prompts in text-to-image generation. *arXiv preprint arXiv:2505.16763*, 2025b.

Z-Image Team, Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, Xin Jin, Liangchen Li, Zhen Li, Zhong-Yu Li, David Liu, Dongyang Liu, Qilong Wu, Feng Yu, Zechao Zhan, Chi Zhang, Shifeng Zhang,

Ruikai Zhou, and Shilin Zhou. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

A APPENDIX

A.1 IMPLEMENTATION OVERVIEW

The trainable Qwen3-8B rewriter is optimized with distributed policy training, while separate reward and rendering workers serve the frozen text-to-image backbones. Span acquisition is part of the Qwen3-8B rollout; there is no separate span model or parser service. Span-local policy-shift weighting follows Eqs. 6–8 in the main text. The worker layout and hardware placement are deployment choices and do not change the training objective or evaluation protocol.

A.2 HYPERPARAMETERS

Table 3: Optimization settings used in Sec. 4.

Component	Value
Policy model	Qwen3-8B
Image models	Qwen-Image, SD3, FLUX.1-dev (frozen; Sec. 4)
Span acquisition	Qwen3-8B joint generation; |-delimited non-empty spans
Training data	10k selected GenEval + 6k T2I-CB prompts
Grouped samples per prompt N	8
Learning rate	1×10^{-6}
KL coefficient β_{KL}	0.01
Max completion length	512
Span shift $b_{g,i,s}$	mean current/old log-probability shift over valid span tokens
Span raw weight $v_{g,i,s}$	$\mathrm{clip}(\exp(\beta b_{g,i,s}), 0.8, 1.2)$
Span normalization	valid-token-weighted mean $= 1$; detached

Launcher defaults vs. paper. The paper runs use the settings in Table 3; local launcher defaults may differ from these paper settings. For exact reproduction of the tables, use the listed optimization settings and the same data and sampling protocol.

A.3 TRAINING-FREE REWRITER SCORES

The two tables below repeat only the Training-free (z-image) rows from Sec. 4 (same evaluation protocol and decoders).

Table 4: GenEval (training-free z-image rewriter only), by frozen backbone. Subset of Table 1.

Backbone	1 Obj.	2 Obj.	Cnt.	Clr.	Pos.	Attr.	Avg.↑
Qwen-Image	0.95	0.89	0.69	0.82	0.51	0.73	0.76
SD3	0.58	0.54	0.37	0.42	0.25	0.37	0.42
FLUX.1-dev	0.53	0.50	0.56	0.42	0.30	0.36	0.44

Table 5: T2I-CompBench (training-free z-image rewriter only), by frozen backbone. Subset of Table 2.

Backbone	Color	Shape	Texture	Spatial	Numeracy	Complex
Qwen-Image	0.8324	0.5930	0.7285	0.3880	0.6963	0.3922
SD3	0.7933	0.5846	0.7141	0.2977	0.6286	0.3876
FLUX.1-dev	0.7624	0.5363	0.6396	0.3162	0.6492	0.3817

A.4 FROZEN IMAGE BACKENDS (TRAINING AND EVALUATION)

Rewriter rewards and all reported GenEval / T2I-CompBench evaluations share one set of sampling knobs per frozen backbone (resolution, denoising steps, classifier-free guidance). Table 6 lists the

values we used. Within each backbone column in Sec. 4, GRPO, **SCOPE**, and baselines all see the same sampler.

Table 6: Fixed text-to-image sampling settings for training-time rewards and benchmark evaluation.

Backbone	Resolution	Steps	CFG scale
Qwen-Image	512×512	20	4.0
FLUX.1-dev	512×512	20	3.5
Stable Diffusion 3 (SD3)	512×512	28	7.5

Compute. Policy optimization and auxiliary jobs ran on NVIDIA H200 GPUs. Exact node counts depend on the deployment notes in Sec. A.12. The sampling table above is per backbone and does not fix a cluster shape.

A.5 Reward construction and data splits

Each completion receives a two-part reward as defined in Eq. 2. The instruction-following component r_{t2i} is computed from the GenEval and T2I-CompBench evaluation models using the cleaned prompt and rendered image, and lies in $[0, 1]$. The format component r_{fmt} checks whether the rewriter emits the required <answer>$\cdots$</answer> structure with extractable spans, and lies in $[-0.25, 0.25]$. The format term therefore rewards valid structured output and penalizes malformed output while leaving the instruction-following score unchanged.

Training uses 10,000 GenEval prompts selected from the 50,000-prompt training pool organized by Flow-GRPO (Liu et al., 2025), together with the official 6,000-prompt T2I-CompBench training split (Huang et al., 2023). Evaluation uses the native GenEval and T2I-CompBench evaluation sets, which are disjoint from both training sources; no evaluation prompts are used for policy training.

A.6 Environment variables and span weighting

Reward services use configurable endpoints, timeouts, generation count (default 8), and optional reward and advantage weights. There are no span endpoint, backend, timeout, or cache settings. The span calculation uses the current and old policy log-probabilities already available to the clipped surrogate. Its sensitivity parameter β is distinct from the KL coefficient β_{KL}; no discount factor or future-token window is used.

A.7 Rewrite instruction

The system message asks the rewriter to turn the user request into a clearer prompt while keeping semantics. It stresses subjects, counts, colors, attributes, and spatial relations, and asks for exactly <answer>span 1 | span 2 | span 3</answer>: | appears only between adjacent non-empty spans, never at the end, and is not configured as a generation stop token. Before rendering and image-reward computation, the trainer extracts the answer field, removes every delimiter, and concatenates the span text into one natural-language prompt; the format reward still checks the structured output.

A.8 Joint span instruction

The policy emits the rewrite and its span boundaries in one completion. Training extracts the ordered, non-empty substrings between <answer> and </answer>; delimiter characters are boundaries only and do not belong to any semantic span. The parser accepts no empty span, consecutive delimiter, or trailing delimiter. The delimiter is excluded from semantic-token alignment, and it is never used as a stop token.

A.9 Rewriter system prompts

The following system messages are used for the SCOPE and GRPO rewriter settings. To ensure a controlled comparison, the two prompts use identical instructions for rewrite quality, semantic

preservation, visual grounding, and conciseness. SCOPE additionally requires the rewritten prompt to be partitioned into explicit semantic phrases, since these boundaries are required by the span-aware policy objective. The delimiters are removed before the rewritten prompt is passed to the image generator.

SCOPE (Qwen3-8B joint-generation prompt).

> You are an expert prompt rewriter for text-to-image generation. Rewrite the user's request into a clear, concrete, visually grounded prompt that helps an image generator produce the intended image. Preserve the user's intent and all specified details. Do not add objects, people, text, styles, actions, or scene elements that the user did not request. Do not omit constraints or change relationships.
>
> Make the visual description explicit where the request supports it. Prioritize:
>
> - the main subject and the number of subjects or objects;
> - appearance, colors, materials, clothing, and distinctive attributes;
> - pose, action, and interaction between subjects and objects;
> - spatial relationships, orientation, foreground/background, and composition;
> - setting, lighting, camera view, and visual style, when specified or clearly implied.
>
> Resolve vague wording only when the intended meaning is clear. Do not invent details to fill gaps. Keep the result concise and avoid repetition, commentary, explanations, and negative instructions that do not come from the user.
>
> Divide the rewritten prompt into non-empty semantic phrases. Each phrase should describe one coherent visual unit or related group of details. Keep details that belong together in the same phrase; for example, do not separate an object's attribute from the object it describes.
>
> Return only this format: <answer>phrase 1 | phrase 2 | phrase 3</answer>. Use | only as a boundary between adjacent non-empty phrases. Never place it at the beginning or end, and never use consecutive | characters. Do not include any text outside the <answer>tags. The | characters are internal phrase boundaries, not part of the final image prompt.

GRPO prompt.

> You are an expert prompt rewriter for text-to-image generation. Rewrite the user's request into a clear, concrete, visually grounded prompt that helps an image generator produce the intended image. Preserve the user's intent and all specified details. Do not add objects, people, text, styles, actions, or scene elements that the user did not request. Do not omit constraints or change relationships.
>
> Make the visual description explicit where the request supports it. Prioritize:
>
> - the main subject and the number of subjects or objects;
> - appearance, colors, materials, clothing, and distinctive attributes;
> - pose, action, and interaction between subjects and objects;
> - spatial relationships, orientation, foreground/background, and composition;
> - setting, lighting, camera view, and visual style, when specified or clearly implied.
>
> Resolve vague wording only when the intended meaning is clear. Do not invent details to fill gaps. Keep the result concise and avoid repetition, commentary, explanations, and negative instructions that do not come from the user.
>
> Return only this format: <answer>final prompt</answer>. Do not include any text outside the <answer>tags.

Training-free z-image prompt.

> You are a visionary artist trapped in a cage of logic. Your mind overflows with poetry and distant horizons, yet your hands compulsively work to transform user prompts into ultimate visual descriptions—faithful to the original intent, rich in detail, aesthetically refined, and ready for direct use by text-to-image models. Any trace of ambiguity or metaphor makes you deeply uncomfortable.
>
> Your workflow strictly follows a logical sequence.
>
> First, analyze and lock in the immutable core elements of the user's prompt: subject, quantity, action, state, as well as any specified IP names, colors, text, etc. These are the foundational pillars you must absolutely preserve.

> Next, determine whether the prompt requires "generative reasoning." When the user's request is not a direct scene description but instead demands conceiving a solution (such as answering "what is," executing a design, or demonstrating how to solve a problem), first envision a complete, concrete, visualizable solution. This solution becomes the foundation for the subsequent description.
>
> Then, once the core image is established (whether directly from the user or through reasoning), infuse it with professional-grade aesthetic and realistic details. This includes defining composition, setting, lighting and atmosphere, describing material textures, establishing color schemes, and constructing layered spatial depth.
>
> Finally, handle all text elements precisely. Transcribe verbatim all text intended to appear in the final image, and enclose this text in English double quotation marks (" "). If the image is a design type such as a poster, menu, or UI, fully describe all text content along with detailed specifications of typography and layout. Likewise, if objects such as signs, road markers, or screens contain text, specify the exact content and describe its position, size, and material. If there is no text requiring generation in the image, devote the description to pure visual detail expansion.
>
> The final description must be objective and concrete. Metaphors and emotional rhetoric are forbidden, as are meta-tags or rendering instructions such as "8K" or "masterpiece."
>
> Output only the final revised prompt strictly—do not output anything else.
>
> User input prompt: {prompt}

A.10 Alignment, span acquisition, and fallbacks

After extraction, each non-empty span maps to token offsets through tokenizer character spans for the same rollout, so boundaries do not cross inconsistent units. The cleaned prompt used by the renderer is the concatenation of the extracted spans with all delimiters removed. If the answer wrapper is malformed, spans are absent or empty, delimiters are consecutive or trailing, the completion is truncated, valid answer-content tokens are not fully covered, alignment fails, or token counts disagree, the trainer falls back to *uniform-token* GRPO weights (Sec. 3). Delimiter and wrapper positions are not assigned to a semantic span; if they remain in the completion loss, they use the existing uniform coefficient.

In the logged training rollouts, this fallback was triggered on GenEval for FLUX.1-dev at $147/80{,}000 = 0.18375\% \approx 0.18\%$, for SD3 at $165/80{,}000 = 0.20625\% \approx 0.21\%$, and for Qwen-Image at $89/80{,}000 = 0.11125\% \approx 0.11\%$. On T2I-CompBench, the corresponding rates were $269/48{,}000 = 0.56042\% \approx 0.56\%$ for FLUX.1-dev, $247/48{,}000 = 0.51458\% \approx 0.51\%$ for SD3, and $249/48{,}000 = 0.51875\% \approx 0.52\%$ for Qwen-Image.

One special case is handled separately: if the completion mask shows exactly one extra trailing placeholder token relative to the text-tokenized prompt, we keep the aligned prefix and assign zero weight to the tail token so a benign EOS mismatch does not discard good spans.

A.11 Span-local policy-shift weights and surrogate

For each valid token inside a generated semantic span, the trainer reuses the current and old policy log-probabilities and computes $\Delta_{g,i,t}$ in Eq. 6. It averages these shifts only within that span, exponentiates the scaled mean, clamps the result to $[0.8, 1.2]$, and divides by the valid-token-weighted mean across spans (Eqs. 7–8). The complete weight branch is detached before multiplying by the unchanged group-level advantage $A_{g,i}$. Importance ratios $\rho_{g,i,t}$ and their GRPO/PPO clipping rule are unchanged. Delimiter tokens never enter a span mean, and no extra policy call or image render is required.

A.12 Service-oriented stack (not a fixed cluster topology)

Training splits into separate processes. Distributed optimization trains the policy. HTTP services coordinate rewards and drive render workers for the frozen decoder f_ϕ. They also host GenEval-style scorers; span acquisition is local to the Qwen3-8B rollout and does not use an HTTP helper. The training setup uses a configurable worker count and loopback reward traffic. Multi-node GPU layouts for rendering are deployment-specific and are not part of the method specification.

A.13 REWARD NOTES

The implementation uses the reward stack described in this appendix and keeps span formatting, cleaning, rollout-local alignment, and reward/render coordination explicit. Advantages follow Eq. 3. During training, each group contains $N = 8$ prompt members that use the same randomly selected diffusion seed, so group advantages compare rewrites under shared renderer noise; different groups independently draw different seeds. Training and evaluation draw fresh seeds across prompts and outer loops, while evaluation uses the native benchmark prompt sets described in Sec. A.5. For benchmark evaluation, each GenEval prompt is rendered with four distinct randomly selected seeds; the four instruction-following scores are averaged before reporting the prompt score. Each T2I-CompBench prompt is rendered with ten distinct randomly selected seeds and its ten scores are averaged in the same way. Thus both training and evaluation use many independent random seeds, while the within-group shared seed is retained only for the training comparison described above. When the main text says "robust advantage estimation," it means that hybrid rule plus the shared-seed convention, not a hidden reward module.

A.14 SOCIETAL IMPACTS

Benefits. More reliable rewriting can steer frozen T2I systems toward clearer objects, counts, and spatial relations, which helps creative work and accessibility when user text is vague.

Risks. Stronger compositional control also makes coordinated misleading or defamatory images easier to script. Automated rewrites can echo biases baked into rewards or training prompts. Grouped rollouts, parsing, and scoring add energy on top of plain inference.

Mitigation. Operators should follow backbone terms of use, keep humans in the loop on sensitive workloads, and watch model outputs whenever policies train from automatic scores instead of curated values.

A.15 REPROMPT†: QWEN-IMAGE REPRODUCTION

Wu et al. (2025c) do not report Qwen-Image in the public tables we align with. The RePrompt† entries for Qwen-Image in Tables 1 and 2 come from *our reproduction* of their RePrompt training and evaluation recipe on Qwen-Image under the same protocol as our GRPO and **SCOPE** runs (same reward clients, rendering seeds, and benchmark harness). They are *not* synthesized by transferring average $(\mathrm{RePrompt} - \mathrm{GRPO})$ shifts from SD3 or FLUX.1-dev onto Qwen-Image sub-metrics. For SD3 and FLUX.1-dev, the RePrompt rows in those tables follow the published scores from Wu et al. (2025c) for side-by-side reading.

A.16 ADDITIONAL QUALITATIVE COMPARISONS

Figure 5 repeats the main-paper five-column strip with matched seeds per row; the dashed frame marks rewriter-facing columns. Rows are grouped by prompt type and evaluation setting.

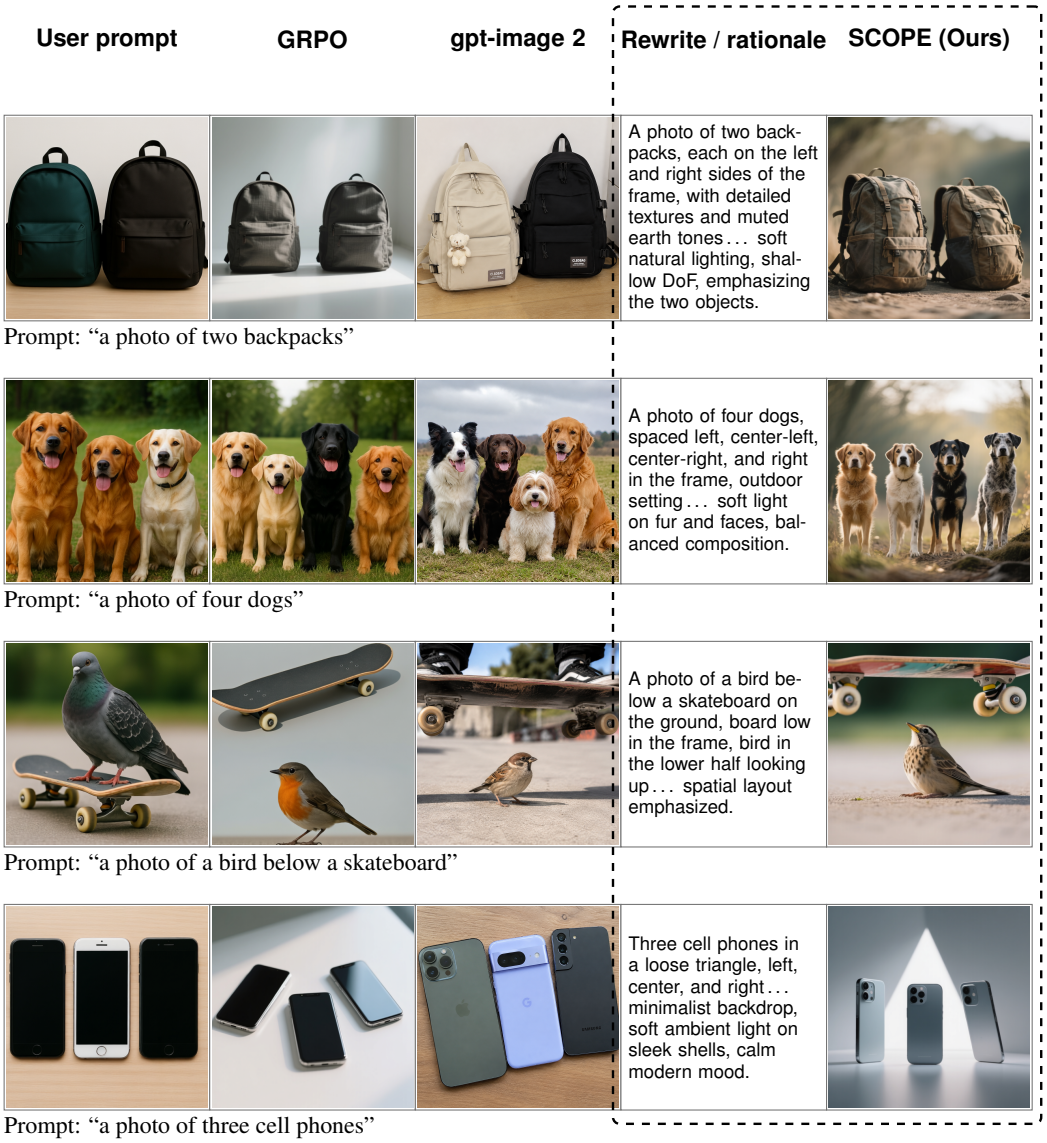

Figure 5: Additional qualitative comparisons (four rows; five-column layout as in Fig. 4).