# EXPLORE WIDELY, EXPLOIT WISELY: HARNESSING TEXT–IMAGE COST ASYMMETRY FOR UNIFIED MULTIMODAL REINFORCEMENT LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Joint reinforcement learning aligns prompt expansion (PE) with image generation, but exploring larger rewrite pools incurs substantial image-based evaluation costs. We exploit a computational asymmetry: batching amortizes text generation more effectively than image generation, allowing broader text exploration without proportionally expanding high-fidelity training rollouts. We propose PEAR, a cost-aware exploration and selection framework for unified multimodal reinforcement learning in reasoning-driven text-to-image generation. PEAR oversamples PEs and evaluates an initial subset using a low-precision, reduced-step image proxy, averaging rewards over multiple images for each candidate. It then allocates a fixed evaluation budget across prompts according to bootstrap-estimated improvements in the mean reward gap between high- and low-reward candidates. These estimates are updated as new feedback arrives. Selected top-$k$ and bottom-$k$ rewrites receive fresh high-fidelity image rollouts for joint policy updates; proxy feedback guides selection but does not directly enter the training loss. Across four evaluation metrics, PEAR improves the arithmetic mean by 4.38 percentage points over the pre-RL PE pipeline and 1.43 points over a joint-training baseline. By evaluating only 50% of the candidate pool, it reduces proxy-screening time by 39.1% relative to full-pool evaluation, with a 0.56% decrease in the mean score, equivalent to a relative decrease of approximately 0.92%. These results demonstrate the value of broad text exploration and selective image evaluation for multimodal policy optimization.

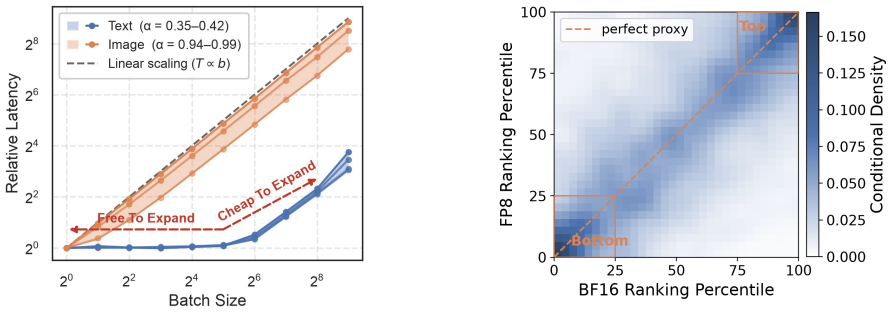
## 1 INTRODUCTION

Text-to-image generation has made substantial progress in producing high-quality images from natural-language descriptions (Cao et al., 2025; Zhao et al., 2026; Diao et al., 2026; Liu et al., 2026a), supported by advances in diffusion modeling and scalable transformer architectures (Rombach et al., 2022; Esser et al., 2024). Despite this progress, visual plausibility does not ensure faithful interpretation of a request: evaluations continue to reveal difficulties with spatial relations, attribute binding, and prompts requiring world knowledge (Ghosh et al., 2023; Niu et al., 2025). Prompt expansion (PE) helps bridge this gap by translating a user's request into a more explicit, generator-compatible description (Hao et al., 2023; Kou et al., 2026). With language-model reasoning, this intermediate stage can resolve implicit concepts and organize scene constraints before image synthesis (Jiang et al., 2025; Kou et al., 2026). A useful rewrite, however, must not only express the intended content but also enable the image model to render it faithfully. We study *unified multimodal reinforcement learning* in this reasoning-driven text-to-image setting, where text and image policies are jointly optimized using image-level feedback, allowing semantic reasoning and visual generation to adapt to one another (Kou et al., 2026; Liu et al., 2026b).

The two generation stages have different computational bottlenecks. Autoregressive LLM decoding is typically memory-bandwidth-bound at small to moderate batch sizes, allowing batching to amortize weight accesses across sequences without a proportional increase in latency (Pope et al., 2023). In contrast, diffusion transformer (DiT) denoising processes all spatial tokens at each step and is typically compute-bound, leaving less room to amortize latency across additional images (Fang et al., 2024). Figure 1(a) compares their batch-scaling behavior: within the measured range, *text gener-*



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.




(a) Text–Image Cost Asymmetry

(b) Proxy Ranking Reliability

Figure 1: **Empirical motivation for text exploration and low-cost candidate screening.** (a) Text generation exhibits sublinear latency growth with batch size ($\alpha = 0.35$–$0.42$), whereas image generation scales nearly linearly ($\alpha = 0.94$–$0.99$), motivating broader exploration on the text side. (b) Conditional distribution of FP8 reward-ranking percentiles given BF16 reference percentiles. Higher percentiles correspond to higher rewards, and darker colors indicate greater conditional density. The concentration around the dashed diagonal indicates approximate agreement between the two rankings. The outlined boxes highlight the top and bottom reward quartiles.

*ation exhibits strongly sublinear latency growth, whereas image generation scales nearly linearly*, with effective scaling exponents of 0.35–0.42 and 0.94–0.99, respectively. This asymmetry suggests that *broad text exploration need not be matched by equally broad high-fidelity image sampling*. We can therefore expand the PE candidate pool while keeping the number of image rollouts used for joint policy updates fixed, creating an opportunity to explore widely and train selectively.

Broader exploration also requires selecting suitable training samples. Empirical studies in visual RL show that retaining high- and low-reward candidates from larger pools can improve policy optimization (Xue et al., 2025; Li et al., 2026). We apply this principle to the text stage: oversample PE candidates and retain the top-$k$ and bottom-$k$ rewrites while keeping the number of formal training samples fixed. The challenge is that a rewrite's image-grounded reward is unavailable until it has been rendered. Evaluating every candidate at training fidelity can therefore offset the cost advantage of text oversampling. The resulting question is *how to identify useful high- and low-reward rewrites without paying the full image-generation cost for the entire candidate pool*.

Efficient screening requires addressing both the cost per candidate and the number of candidates evaluated. A lower-cost evaluator need not reproduce high-fidelity rewards exactly, provided it preserves enough relative information to distinguish the two reward tails. Low-precision, reduced-step rollouts have been shown to support candidate screening in visual RL (Li et al., 2026). Figure 1(b) further shows that *FP8 and BF16 reward rankings approximately agree*, with the conditional distribution concentrated around the diagonal, including both reward tails. This supports proxy-guided selection without implying exact rankings or identical rewards. Each PE is scored by averaging rewards over multiple low-precision, reduced-step proxy images. However, screening a large pool still incurs substantial image-generation and reward-computation costs, and additional evaluations need not improve selection equally across prompts. This motivates *allocating evaluation effort according to the expected improvement in two-sided selection*, rather than the current reward spread alone. Following the broader principle of using inexpensive observations to guide resource allocation (Li et al., 2018; Kandasamy et al., 2016), we use the feedback already collected to decide which prompts should receive further candidate evaluations under a fixed total budget.

We propose **PEAR** (*Prompt Exploration for Adaptive Joint Reinforcement Learning*), a cost-aware framework for unified multimodal RL. PEAR oversamples rewrites for each prompt and evaluates equal-sized initial subsets using a low-precision, reduced-step image proxy synchronized with the current image policy. Each PE is scored by averaging rewards over multiple proxy images. Bootstrap resampling (Efron, 1979) estimates the improvement in the mean reward gap between top-$k$ and bottom-$k$ candidates from additional evaluations. PEAR allocates the remaining budget across prompts in rounds according to these estimated gains, updating priorities as new feedback arrives. Finally, each prompt contributes a fixed number of high- and low-reward PEs for fresh high-fidelity

generation and joint policy updates. *Proxy feedback guides evaluation and selection, while only fresh high-fidelity rewards drive training*, keeping the text candidate pool size and formal training sample count fixed.

Across four evaluation metrics, PEAR achieves an arithmetic mean of 60.28, compared with 51.82 for the base image model, 55.90 for the PE pipeline before joint RL, and 58.85 for the joint-training baseline (Liu et al., 2026b). The improvements over the latter two baselines are 4.38 and 1.43 percentage points, respectively. Compared with full-pool proxy evaluation, PEAR evaluates only 50% of the candidate pool and reduces proxy-screening time from 630.2 s to 383.8 s per training iteration, a reduction of 39.1%. The mean score decreases from 60.84 to 60.28, an absolute difference of 0.56 points and a relative decrease of approximately 0.92%, while all four metrics remain above the joint-training baseline. These results demonstrate improved generation performance over the baselines and a favorable cost–performance trade-off relative to full-pool evaluation.

Our contributions are threefold:

- We exploit text–image cost asymmetry to broaden text-side exploration in unified multimodal RL while keeping the number of high-fidelity training rollouts fixed.
- We develop a proxy-guided evaluation and selection strategy that combines low-cost reward estimates with bootstrap-estimated marginal gains to allocate a fixed evaluation budget across prompts.
- We evaluate PEAR in reasoning-driven text-to-image generation, demonstrating improvements over the pre-RL PE pipeline and a joint-training baseline, together with reduced full-pool screening overhead and less than a 1% relative decrease in the four-metric mean.

## 2 METHOD

### 2.1 PROBLEM SETUP AND OVERVIEW

Following reasoning-driven joint generation (Kou et al., 2026; Liu et al., 2026b), let $x$ denote an original prompt, $z \sim \pi_\phi(\cdot \mid x)$ a text response, and $c = f(z)$ the extracted rewrite. The image policy generates $y \sim p_\theta(\cdot \mid c)$. The underlying task objective is

$$J(\phi, \theta) = \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{z\sim\pi_\phi(\cdot|x)}\mathbb{E}_{y\sim p_\theta(\cdot|f(z))}[R(x, y)]. \tag{1}$$

The reward is always evaluated against $x$, preserving the original request as the alignment target. The image model receives the extracted rewrite, whereas the text update uses its associated full response.

For a batch of $P$ prompts, PEAR generates $N$ text candidates per prompt but trains on only $2k$, comprising $k$ high-reward and $k$ low-reward rewrites. A budget $B$ controls how many candidates receive proxy image evaluations across the batch. Text generation, proxy evaluation, and formal training are therefore separate stages: the first produces $PN$ candidates, the second evaluates $B$ of them, and the third uses $P(2k)$ high-fidelity image rollouts in our one-image-per-rewrite setting.

### 2.2 TEXT OVERSAMPLING AND PROXY EVALUATION

For each prompt $x_i$, we sample responses $z_{ij} \sim \pi_\phi(\cdot \mid x_i)$ and extract $c_{ij} = f(z_{ij})$, for $j = 1, \ldots, N$. We maintain an inference-only proxy generator $\tilde{p}_\theta$ constructed from the current image policy. Before each outer iteration's evaluation stage, we synchronize its effective weights, including the merged LoRA update, and requantize the proxy. Its weights remain fixed throughout the allocation rounds.

For an evaluated candidate $c_{ij}$, the proxy generates $m$ images with different initial noise draws:

$$\tilde{y}_{ij\ell} \sim \tilde{p}_\theta(\cdot \mid c_{ij}), \qquad \tilde{r}_{ij\ell} = R(x_i, \tilde{y}_{ij\ell}), \qquad \hat{r}_{ij} = \frac{1}{m}\sum_{\ell=1}^{m} \tilde{r}_{ij\ell}. \tag{2}$$

Averaging separates rewrite evaluation from a single favorable or unfavorable image realization.

Our implementation uses FP8 weights and dynamic activation quantization with six-step ODE sampling and $m = 4$ proxy images per evaluated rewrite. Formal training uses BF16 and a ten-step

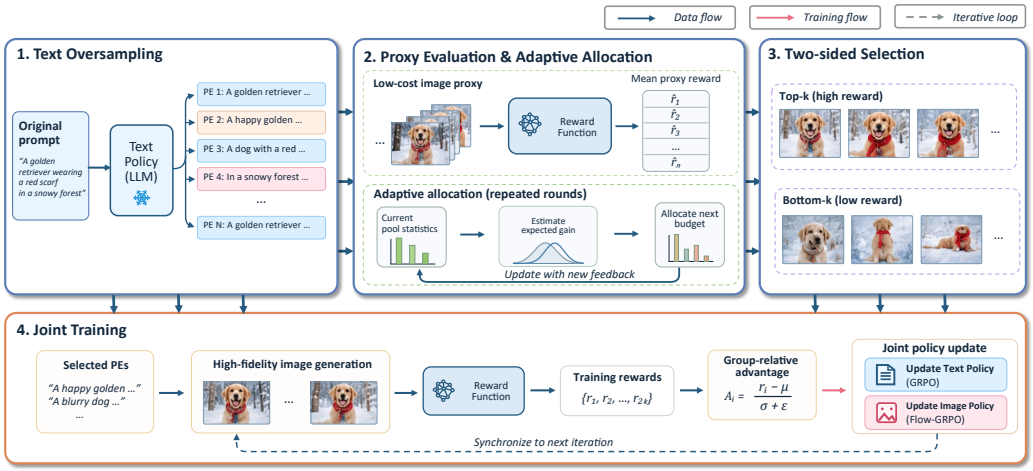



Figure 2: **Overview of PEAR for unified multimodal reinforcement learning.** (1) The text policy generates $N$ prompt expansions (PEs) for each original prompt. (2) A low-cost image proxy evaluates a subset of candidates using mean rewards over multiple images. Bootstrap-estimated gains in two-sided selection utility guide the iterative allocation of a fixed evaluation budget across prompts. (3) The top-$k$ and bottom-$k$ evaluated PEs are retained for each prompt, keeping the formal training sample count fixed. (4) Selected PEs are rendered again at high fidelity, and fresh rewards yield group-relative advantages for jointly updating the text and image policies. Proxy feedback guides evaluation allocation and candidate selection but does not directly enter the policy optimization loss.

stochastic sampling configuration; both stages generate $512 \times 512$ images. Similar to the separation of screening and optimization in Sol-RL (Li et al., 2026), proxy images are never used as training trajectories. Unlike seed selection, our proxy estimates the performance of a *rewrite across image samples*; the selected rewrites receive fresh rollouts at training fidelity.

### 2.3 Two-Sided Selection Utility

Let $\mathcal{E}_i$ be the evaluated candidate indices for prompt $i$, $n_i = |\mathcal{E}_i|$, and $\mathcal{R}_i = \{\hat{r}_{ij} : j \in \mathcal{E}_i\}$ the multiset of observed mean rewards. For sorted values $r_{(1)} \le \cdots \le r_{(n)}$, define

$$U_k(\mathcal{R}) = \frac{1}{k} \sum_{\ell=1}^{k} \left[ r_{(n-\ell+1)} - r_{(\ell)} \right], \qquad n \ge 2k. \tag{3}$$

This utility measures the reward separation of the two tails that will be retained, rather than overall variance or a single maximum–minimum pair.

The final set $\mathcal{S}_i = \mathcal{H}_i \cup \mathcal{L}_i$ contains the top-$k$ and bottom-$k$ candidate indices from this ordering. A fixed random tie order ensures that the two sets remain disjoint even when rewards are tied. We use $U_k$ as a surrogate for selection quality, not as an additional training reward or a guarantee of improved policy optimization. In particular, a larger raw reward gap does not necessarily produce larger normalized advantages.

### 2.4 Adaptive Evaluation under a Fixed Budget

We randomly permute each prompt's candidate pool and first evaluate $n_0 \ge 2k$ candidates per prompt. The remaining budget is $B_{\text{extra}} = B - Pn_0$, where $Pn_0 \le B \le PN$. Additional evaluations consume unseen candidates in permutation order, without replacement. The controller therefore decides *where to explore and how much*, without accessing rewards for candidates that have not been evaluated.

**Empirical marginal gain.** We allocate evaluations in blocks of $d$ rewrites. To estimate the value of one additional block, we use empirical resampling (Efron, 1979). Let $\widehat{F}_i$ assign equal probability to

the entries of $\mathcal{R}_i$. For each of $L$ Monte Carlo trials, draw a multiset $\mathcal{Z}_i^{(a)}$ of $d$ rewards independently from $\widehat{F}_i$, with replacement, and estimate

$$\widehat{\Delta}_i = \frac{1}{L}\sum_{a=1}^{L}\left[U_k(\mathcal{R}_i \uplus \mathcal{Z}_i^{(a)}) - U_k(\mathcal{R}_i)\right], \tag{4}$$

where $\uplus$ denotes multiset concatenation. We use $L = 200$. Reselecting both tails after each simulated block accounts for competition among candidates for the same retained positions. These simulations operate only on observed rewards and require no additional image generation.

Equation equation 4 is a plug-in estimate based on the observed pool. It cannot extrapolate beyond the observed reward range, and its Monte Carlo error does not quantify uncertainty about truly unseen rewrites. We use it as an allocation priority rather than a calibrated guarantee of future gain.

**Round-based allocation.** Each round distributes a prescribed number of evaluation blocks proportionally to the estimated marginal gains $\widehat{\Delta}_i$ in Equation equation 4, subject to each prompt's remaining candidate capacity. Prompts reaching capacity are capped, and their excess quota is redistributed. We convert continuous quotas to integer blocks by largest-remainder rounding, breaking ties by fewer previously allocated blocks and then a fixed random order. If all remaining eligible priorities are zero, the outstanding quota is distributed uniformly. Thus, the full prescribed budget is spent rather than terminated early.

Selected evaluations are batched across prompts. Their rewards are appended to the corresponding pools, and priorities are recomputed for prompts receiving new observations. This continues until

$$\sum_{i=1}^{P}|\mathcal{E}_i| = B, \qquad n_0 \le |\mathcal{E}_i| \le N. \tag{5}$$

Allocating all remaining blocks in one round recovers the static-gain variant. Evaluating all $PN$ candidates recovers full-pool screening. Dynamic allocation changes only proxy evaluation effort; it does not change the $2k$ selected training candidates per prompt.

### 2.5 Joint Updates with Fresh Image Feedback

For each selected rewrite $j \in \mathcal{S}_i$, we generate a fresh high-fidelity training rollout and evaluate its terminal image:

$$y_{ij} \sim p_\theta(\cdot \mid c_{ij}), \qquad r_{ij} = R(x_i, y_{ij}). \tag{6}$$

The stochastic rollout provides the trajectory information required by the image-policy updater. Neither proxy trajectories nor proxy rewards are substituted for these training samples.

Following group-relative optimization (Shao et al., 2024) and the shared-feedback joint-training setting (Liu et al., 2026b), we normalize the fresh rewards within each original prompt's selected group:

$$\mu_i = \frac{1}{2k}\sum_{j\in\mathcal{S}_i} r_{ij}, \qquad \sigma_i^2 = \frac{1}{2k}\sum_{j\in\mathcal{S}_i}(r_{ij}-\mu_i)^2, \qquad A_{ij} = \frac{r_{ij}-\mu_i}{\max(\sigma_i, \epsilon_A)}, \tag{7}$$

with $\epsilon_A = 10^{-6}$. The same advantage supervises the associated text response and image rollout. Grouping across rewrites allows image-policy updates even with one formal image per rewrite.

We pass these advantages to the existing text-side GRPO and image-side Flow-GRPO update routines (Shao et al., 2024; Liu et al., 2025), retaining their configured clipping and reference-regularization settings. In the configuration studied here, one image-policy update phase is followed by one text-policy update phase, after which the text rollout weights are synchronized. PEAR modifies candidate exploration and evaluation rather than introducing a new policy loss.

The proxy top/bottom labels determine inclusion only; advantage signs are recomputed from fresh rewards. Because reward-dependent selection changes the training-sample distribution, the resulting selected-sample update is not claimed to be an unbiased Monte Carlo policy-gradient estimator of Equation equation 1. With fixed $m$, adaptive screening reduces proxy images from $PNm$ to $Bm$ while preserving $P(2k)$ formal training images. Actual wall-clock savings additionally depend on weight synchronization, reward evaluation, and round-level overhead.

Table 1: **Quantitative comparison on WISE, preference metrics, and GenEvalv2.** The best and second-best results in each column are **bold** and underlined, respectively. Avg denotes the arithmetic mean of WISE Overall, HPSv2, PickScore, and GenEvalv2.

| Method | WISE ↑ | | | | | | | Preference ↑ | | GenEvalv2 ↑ | Avg ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Cultural | Time | Space | Biology | Physics | Chemistry | **Overall** | HPSv2 | PickScore | | |
| SD3.5-M | 50.78 | 41.88 | 37.71 | 21.98 | 45.31 | 23.33 | 40.74 | 27.95 | 85.67 | 52.91 | 51.82 |
| SD3.5-L | 52.38 | 45.00 | 45.31 | 28.23 | 47.29 | 27.81 | 44.19 | 29.63 | 87.96 | 56.35 | 54.53 |
| FLUX1.dev | 51.59 | 45.31 | 52.40 | 21.56 | 41.77 | 26.88 | 43.19 | **30.63** | **88.45** | 54.20 | 54.12 |
| Qwen3-8B + SD3.5-M | 77.56 | 52.81 | 56.15 | 38.54 | 54.27 | 53.23 | 61.63 | 27.12 | 85.33 | 49.53 | 55.90 |
| Train Text Only | 79.13 | 56.67 | 57.94 | 40.63 | 59.79 | 56.46 | 64.23 | 27.23 | 85.73 | 54.59 | 57.94 |
| Train Image Only | 79.22 | 57.58 | 58.96 | 42.35 | 60.13 | 56.79 | 64.79 | 26.33 | 85.22 | 53.34 | 57.42 |
| DualGRPO | 81.78 | 57.46 | 58.54 | 50.23 | 62.92 | 60.79 | 67.51 | 27.30 | 85.88 | **57.69** | 59.59 |
| UniGRPO | 80.38 | 55.42 | 58.54 | 48.44 | 62.40 | 59.48 | 66.26 | 27.69 | 85.19 | 56.26 | 58.85 |
| PEAR-NoProxy | 81.88 | 59.38 | 62.71 | 49.79 | 63.44 | 60.94 | 68.30 | 27.15 | 85.93 | 56.52 | 59.48 |
| PEAR | **82.16** | **60.63** | **63.44** | **50.42** | **65.94** | **63.65** | **69.35** | 28.23 | 86.31 | 57.23 | **60.28** |

# 3 EXPERIMENTS

## 3.1 EXPERIMENTAL SETUP

We use Qwen3-8B (Qwen Team, 2025) as the text policy and Stable Diffusion 3.5 Medium (Esser et al., 2024; Stability AI, 2024) as the image policy, training both on eight GPUs. Both policies are optimized using the WISE-based reward (Niu et al., 2025). Each batch contains 32 prompts, each expanded into 96 PEs. Adaptive allocation evaluates 50% of the candidates across the batch, scoring each evaluated PE with the mean reward of four FP8 images generated using six denoising steps. The top-12 and bottom-12 evaluated PEs per prompt are rendered again in BF16 with ten steps for joint training. We evaluate in-domain performance on WISE (Niu et al., 2025) and out-of-domain performance using HPSv2 (Wu et al., 2023), PickScore (Kirstain et al., 2023), and GenEvalv2 (Kamath et al., 2025) with VQAScore (Lin et al., 2024). Each checkpoint is evaluated with eight randomly selected, distinct generation seeds, and metric scores are averaged across runs. Detailed implementation settings, evaluation protocols, and baseline configurations are provided in Appendix B.

## 3.2 MAIN RESULTS

Table 1 compares PEAR with the base image model, the pre-RL PE pipeline, single-policy training, and joint-training baselines. We also include **PEAR-NoProxy**, which evaluates each candidate rewrite using a single training-fidelity image before two-sided selection, whereas PEAR combines multi-image proxy feedback with adaptive budget allocation. These comparisons examine overall performance, the distribution of gains across evaluation metrics, and the effectiveness of the complete framework relative to high-fidelity screening. *PEAR improves aggregate performance beyond PE integration and existing joint optimization.* It achieves the highest average score of 60.28, exceeding the pre-RL PE pipeline (55.90) by 4.38 percentage points. The gains over text-only and image-only training are 2.34 and 2.86 points, respectively. PEAR also outperforms UniGRPO and DualGRPO by 1.43 and 0.69 points, indicating benefits beyond adding a rewriting stage or jointly updating the two policies.

Beyond the aggregate score, *PEAR improves knowledge-grounded generation across all six WISE categories.* It achieves the highest WISE Overall score of 69.35, compared with 66.26 for UniGRPO and 67.51 for DualGRPO, and ranks first in every WISE category. The largest gains over UniGRPO occur in Time (55.42 to 60.63), Space (58.54 to 63.44), and Chemistry (59.48 to 63.65), with improvements also observed in Cultural knowledge, Biology, and Physics. *These gains extend beyond WISE, yielding the strongest preference scores and improved compositional prompt following relative to UniGRPO.* PEAR achieves the highest HPSv2 and PickScore scores among methods using the SD3.5-Medium backbone, reaching 28.23 and 86.31, respectively. These results exceed UniGRPO by 0.54 and 1.12 percentage points and improve upon the base SD3.5-Medium model's scores of 27.95 and 85.67. However, SD3.5-Large and FLUX.1-dev achieve higher scores on both

Table 2: **Ablation studies on budget allocation and candidate selection.** Parentheses indicate absolute changes from UniGRPO: percentage points for performance and seconds for preview time. The best and second-best results in each column are **bold** and underlined, respectively.

| Method | Performance ↑ | | | | | Preview (s) ↓ |
|---|---|---|---|---|---|---|
| | WISE | HPSv2 | PickScore | GenEvalv2 | **Avg** | |
| UniGRPO | 66.26 (0.00) | 27.69 (0.00) | 85.19 (0.00) | 56.26 (0.00) | 58.85 (0.00) | **0.0** (0.0) |
| PEAR | 69.35 (+3.09) | 28.23 (+0.54) | 86.31 (+1.12) | 57.23 (+0.97) | 60.28 (+1.43) | 383.8 (+383.8) |
| *Budget Allocation Ablation* | | | | | | |
| PEAR-Static | 67.64 (+1.38) | 26.30 (−1.39) | 84.64 (−0.55) | 57.93 (+1.67) | 59.13 (+0.28) | 367.2 (+367.2) |
| PEAR-HalfProxy | 68.51 (+2.25) | 27.57 (−0.12) | 84.42 (−0.77) | 57.05 (+0.79) | 59.39 (+0.54) | 341.4 (+341.4) |
| PEAR-FullProxy | **70.04** (+3.78) | **28.45** (+0.76) | **86.90** (+1.71) | **57.99** (+1.73) | **60.84** (+1.99) | 630.2 (+630.2) |
| *Candidate Selection Ablation* | | | | | | |
| PEAR-Top24 | 66.88 (+0.62) | 27.21 (−0.48) | 84.48 (−0.71) | 57.64 (+1.38) | 59.05 (+0.20) | 629.3 (+629.3) |
| PEAR-Bottom24 | 63.28 (−2.98) | 26.42 (−1.27) | 84.95 (−0.24) | 53.67 (−2.59) | 57.08 (−1.77) | 638.7 (+638.7) |
| PEAR-Random24 | 66.52 (+0.26) | 27.04 (−0.65) | 84.38 (−0.81) | 56.22 (−0.04) | 58.54 (−0.31) | 645.1 (+645.1) |

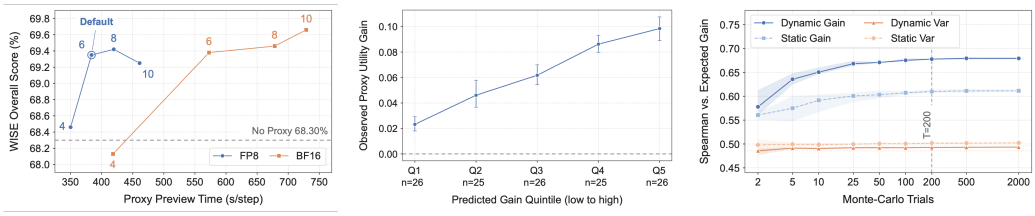
metrics, with FLUX.1-dev reaching 30.63 and 88.45. The improvements therefore demonstrate stronger preference alignment within the shared-backbone comparison, rather than superiority over all evaluated image models. On GenEvalv2, PEAR improves over UniGRPO from 56.26 to 57.23, although DualGRPO retains the highest score of 57.69. PEAR thus leads three of the four aggregate evaluation metrics, while remaining 0.46 points below the best GenEvalv2 result.

The comparison with PEAR-NoProxy further shows that *the complete proxy-based framework outperforms the high-fidelity screening configuration.* PEAR increases the average score from 59.48 to 60.28, with improvements in WISE Overall (68.30 to 69.35), HPSv2 (27.15 to 28.23), PickScore (85.93 to 86.31), and GenEvalv2 (56.52 to 57.23). A possible contributor is the sensitivity of single-image screening to generation randomness. In PEAR-NoProxy, an unusually favorable or unfavorable image can influence a PE's ranking without reliably reflecting the rewrite's expected performance. PEAR instead averages rewards over four proxy images generated with distinct random seeds, reducing reliance on any single realization and potentially providing a more representative basis for selection. This comparison, however, does not isolate the contribution of reward averaging from adaptive budget allocation.

### 3.3 Ablation Studies

Table 2 examines how evaluation-budget allocation and candidate selection affect performance and proxy-screening cost. We compare PEAR with single-round allocation (PEAR-Static), a fixed half-evaluation configuration (PEAR-HalfProxy), and full-pool proxy evaluation (PEAR-FullProxy), before examining alternative selection rules. *Updating allocation priorities with new feedback improves aggregate performance over static allocation and fixed half evaluation.* PEAR achieves an average score of 60.28, exceeding PEAR-Static and PEAR-HalfProxy by 1.15 and 0.89 percentage points, respectively. It improves all four metrics over PEAR-HalfProxy and three over PEAR-Static, although the latter achieves a higher GenEvalv2 score (57.93 versus 57.23). These gains require additional preview time: 383.8 s for PEAR, compared with 367.2 s for PEAR-Static and 341.4 s for PEAR-HalfProxy. The results therefore favor feedback-driven allocation for performance, while showing that fewer allocation rounds or a fixed evaluation pool can reduce screening time.

The comparison with full-pool evaluation reveals a complementary trade-off: *PEAR substantially reduces proxy-screening cost while retaining a similar aggregate score.* Preview time decreases from 630.2 s for PEAR-FullProxy to 383.8 s for PEAR, a reduction of 39.1%. Meanwhile, the average score decreases from 60.84 to 60.28, an absolute difference of 0.56 points and a relative decrease of 0.92%. PEAR remains above UniGRPO on all four evaluation metrics, with an average improvement of 1.43 points. Full evaluation nevertheless yields higher scores on each metric, indicating that adaptive allocation offers a lower-screening-cost alternative rather than a uniform improvement over evaluating every candidate. These savings concern proxy evaluation, not end-to-end training time.




(a) Proxy Cost–Quality Trade-off (b) Gain Prediction Reliability (c) Estimator Sensitivity

Figure 3: **Analysis of proxy configurations and adaptive evaluation.** (a) Final WISE Overall scores versus average proxy-preview time per training iteration for FP8 and BF16 proxies. Point annotations indicate denoising steps; the circled point marks the default six-step FP8 configuration. The horizontal dashed line denotes the no-proxy baseline. (b) Observed proxy utility gains across quintiles of predicted gains for 128 training-disjoint prompts. Predictions use 24 initial PEs, and observed gains average three separate eight-PE extensions of the same initial pool. Error bars show 95% prompt-bootstrap confidence intervals. (c) Spearman correlation with block-averaged observed gains, averaged over three reconstructed decision rounds, as a function of Monte Carlo trials. Gain- and variance-based scores are compared with either dynamic rescoring or fixed initial scores. Curves show means over five resampling seeds, with shaded min–max ranges.

Turning to candidate selection, we use PEAR-FullProxy as the reference for comparing selection rules under full-pool evaluation. *Retaining both reward tails is more effective than selecting only high-reward, only low-reward, or random candidates.* The top-12 and bottom-12 PEs yields an average score of 60.84, compared with 59.05 for PEAR-Top24, 57.08 for PEAR-Bottom24, and 58.54 for PEAR-Random24. The corresponding gains are 1.79, 3.76, and 2.30 points, and two-sided selection performs better on all four evaluation metrics. Preview times are comparable: 630.2 s for PEAR-FullProxy versus 629.3–645.1 s for the alternative selection rules. Notably, top-only selection does not outperform random selection on the average score, while bottom-only selection falls 1.77 points below UniGRPO. These results show that simply choosing the highest-reward candidates does not reproduce the benefits of a training group containing both high- and low-reward rewrites.

### 3.4 Further Analysis

**Proxy cost–quality trade-off.** Figure 3 examines the choice of proxy configuration, the predictive value of estimated gains, and the sampling budget used by the allocation estimator. *Six-step FP8 provides a favorable balance between proxy cost and final generation quality.* We vary only proxy precision and denoising steps, running one complete 100-update training run per configuration and evaluating each resulting checkpoint on 1,000 WISE prompts. As shown in Figure 3(a), seven of the eight configurations outperform the no-proxy baseline of 68.30, with four-step BF16 as the only exception. The default six-step FP8 proxy achieves 69.35 with approximately 384 s of preview time per training iteration, whereas six-step BF16 requires approximately 570 s to reach a similar score. Increasing FP8 denoising to eight steps yields only a small improvement, while ten steps reduces the score to 69.25. These observations support six-step FP8 as an economical default rather than the highest-scoring configuration.

**Gain prediction reliability.** Beyond proxy efficiency, *estimated gains identify prompts that benefit more from additional candidate evaluation.* We freeze a checkpoint and analyze 128 WISE prompts with no overlap with the training set. Predictions use only the initial 24 PEs and 200 bootstrap trials; validation separately adds each of three disjoint eight-PE blocks to the same initial pool and averages the resulting utility gains. In Figure 3(b), mean observed gains increase monotonically across the five predicted-gain bins, from 0.0232 to 0.0984. The highest bin gains approximately 4.2 times as much as the lowest, and their 95% bootstrap intervals do not overlap. The prompt-level Spearman correlation between predicted and observed gains is 0.819. These results support using predicted marginal gains to prioritize prompts for additional candidate evaluation under a fixed budget.

**Estimator sensitivity.** Finally, *gain-based scoring substantially outperforms variance-based scoring, while additional Monte Carlo trials beyond $T = 200$ offer little improvement.* Figure 3(c)

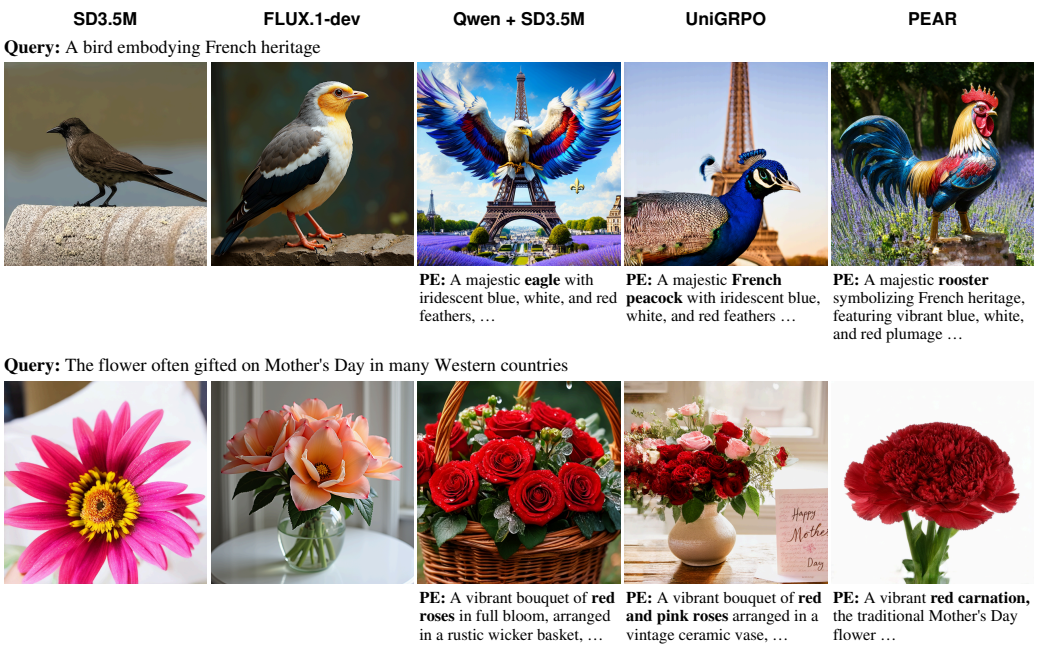



Figure 4: **Qualitative comparison on implicit cultural references.** PEAR identifies the intended rooster and carnation in its prompt expansions and renders the corresponding subjects. The displayed PE excerpts highlight how different methods interpret the same implicit references.

shows that, at $T = 2{,}000$, dynamic gain scoring exceeds dynamic variance and static gain scoring by approximately 0.185 and 0.068 in Spearman correlation, respectively. Increasing the sampling budget does not close the gain–variance gap, and gain prediction remains nearly unchanged beyond $T = 200$. These trials only resample cached rewards on the CPU, without additional image generation or reward-model calls, making their overhead negligible relative to proxy evaluation.

**Qualitative case study.** Figure 4 illustrates how *PEAR resolves implicit cultural references and preserves the intended subject in both prompt expansion and image generation.* For "A bird embodying French heritage," SD3.5-M and FLUX.1-dev generate birds other than the rooster. The Qwen-based PE pipeline and UniGRPO specify an eagle and a peacock, respectively. Although they incorporate French flag colors, these contextual cues do not correct the mismatch in the central subject. PEAR instead explicitly names a rooster in its PE and renders the corresponding bird. For the Mother's Day flower, the Qwen-based pipeline and UniGRPO both specify roses, whereas PEAR identifies the benchmark's intended carnation and depicts its densely packed, ruffled petals. These cases illustrate the distinction between adding thematically associated visual details and identifying the object implied by the query. PEAR translates the implicit reference into an explicit subject constraint, maintaining semantic consistency between the rewrite and the resulting image.

## 4 Conclusion and Limitations

We presented PEAR, which exploits text–image cost asymmetry through broad text exploration, multi-image proxy evaluation, and marginal-gain-driven budget allocation, while keeping the formal training sample count fixed. PEAR improves aggregate performance over joint-training baselines and reduces preview time by 39.1% relative to full-pool evaluation, with a 0.56-percentage-point decrease in the four-metric mean. Our evaluation is limited to one text–image model pair, and efficiency depends on hardware and proxy configurations. Moreover, empirical resampling may overlook valuable unseen candidates, and the two-sided reward gap does not directly measure downstream learning gains. Future work will explore broader model families and allocation criteria more closely aligned with training benefits.

## AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript drafting and language editing, literature searches and reference formatting, and the preparation of schematic figures and LaTeX tables, as well as to obtain suggestions on manuscript structure and presentation. We did not use these tools to refine the methodological framework, inform the experimental design, or interpret the experimental results. All AI-assisted work was reviewed by the authors, and LLM-generated code was verified and tested for correctness by two authors. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

## REFERENCES

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023. URL https://arxiv.org/abs/2305.13301.

Black Forest Labs. FLUX.1 [dev]. Hugging Face model card, 2024. URL https://huggingface.co/black-forest-labs/FLUX.1-dev.

Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. Hunyuanimage 3.0 technical report. *arXiv e-prints*, pp. arXiv–2509, 2025.

Haiwen Diao, Jiahao Wang, Chenjing Ding, Hanming Deng, Jiangnan Chen, Ruixi Zhang, Ruohui Wang, Wenwen Tong, Xiangyu Fan, Yubo Wang, et al. Sensenova-u1. 5: Towards native unified visual intelligence. *arXiv preprint arXiv:2609.11929*, 2026.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7 (1):1–26, 1979. doi: 10.1214/aos/1176344552. URL https://doi.org/10.1214/aos/1176344552.

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, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

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 *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.16381.

Jiarui Fang, Jinzhe Pan, Xibo Sun, Aoyu Li, and Jiannan Wang. xDiT: An inference engine for diffusion transformers (DiTs) with massive parallelism. *arXiv preprint arXiv:2411.01738*, 2024. URL https://arxiv.org/abs/2411.01738.

Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. *arXiv preprint arXiv:2310.11513*, 2023. URL https://arxiv.org/abs/2310.11513.

Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2212.09611.

Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2I-R1: Reinforcing image generation with collaborative semantic-level and token-level CoT. *arXiv preprint arXiv:2505.00703*, 2025. URL https://arxiv.org/abs/2505.00703.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. GenEval 2: Addressing benchmark drift in text-to-image evaluation. *arXiv preprint arXiv:2512.16853*, 2025. URL https://arxiv.org/abs/2512.16853.

Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. The multi-fidelity multi-armed bandit. In *Advances in Neural Information Processing Systems*, 2016. URL https://arxiv.org/abs/1610.09726.

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. *arXiv preprint arXiv:2305.01569*, 2023. URL https://arxiv.org/abs/2305.01569.

Siqi Kou, Jiachun Jin, Zetong Zhou, Ye Ma, Yugang Wang, Quan Chen, Peng Jiang, Xiao Yang, Jun Zhu, Kai Yu, and Zhijie Deng. Think-Then-Generate: Reasoning-aware text-to-image diffusion with LLM encoders. *arXiv preprint arXiv:2601.10332*, 2026. URL https://arxiv.org/abs/2601.10332.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018. URL https://jmlr.org/papers/v18/16-558.html.

Yitong Li, Junsong Chen, Shuchen Xue, Pengcuo Zeren, Siyuan Fu, Dinghao Yang, Yangyang Tang, Junjie Bai, Ping Luo, Song Han, and Enze Xie. FP4 explore, BF16 train: Diffusion reinforcement learning via efficient rollout scaling. *arXiv preprint arXiv:2604.06916*, 2026. URL https://arxiv.org/abs/2604.06916.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. *arXiv preprint arXiv:2404.01291*, 2024. URL https://arxiv.org/abs/2404.01291.

Jiaxiang Liu, Zhida Feng, Pengyu Zou, Zhenyu Qian, Tianrui Zhu, Jun Xia, Yuehu Dong, Yanzheng Lin, Honglin Xiong, Anqi Chen, et al. Ernie-image technical report. *arXiv preprint arXiv:2605.25347*, 2026a.

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. URL https://arxiv.org/abs/2505.05470.

Jie Liu, Zilyu Ye, Linxiao Yuan, Shenhan Zhu, Yu Gao, Jie Wu, Kunchang Li, Xionghui Wang, Xiaonan Nie, Weilin Huang, and Wanli Ouyang. UniGRPO: Unified policy optimization for reasoning-driven visual generation. *arXiv preprint arXiv:2603.23500*, 2026b. URL https://arxiv.org/abs/2603.23500.

Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Chaoran Feng, Fanqing Meng, Kunpeng Ning, Bin Zhu, and Li Yuan. WISE: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*, 2025. URL https://arxiv.org/abs/2503.07265.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. In *Proceedings of Machine Learning and Systems*, volume 5, 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/hash/c4be71ab8d24cdfb45e3d06dbfca2780-Abstract-mlsys2023.html.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. doi: 10.1109/CVPR52688.2022.01042. URL https://arxiv.org/abs/2112.10752.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Stability AI. Introducing Stable Diffusion 3.5. Official model release, 2024. URL https://stability.ai/news-updates/introducing-stable-diffusion-3-5.

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. URL https://arxiv.org/abs/2306.09341.

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. URL https://arxiv.org/abs/2505.07818.

Bing Zhao, Chenfei Wu, Deqing Li, Hao Meng, Jiahao Li, Jie Zhang, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kuan Cao, et al. Qwen-image-2.0 technical report. *arXiv preprint arXiv:2605.10730*, 2026.

# A Related Work

**Prompt adaptation and reasoning-driven generation.** Promptist learns to adapt user inputs into model-preferred prompts through supervised fine-tuning and reinforcement learning (Hao et al., 2023). Reasoning-driven methods go beyond prompt adaptation by explicitly organizing the content to be depicted. T2I-R1 jointly optimizes semantic-level planning and token-level visual generation through BiCoT-GRPO (Jiang et al., 2025). Think-Then-Generate trains an LLM encoder to reason and rewrite before diffusion, then co-optimizes the encoder and diffusion backbone with Dual-GRPO (Kou et al., 2026). UniGRPO combines text-side GRPO with image-side Flow-GRPO for joint optimization under image feedback (Liu et al., 2026b). PEAR builds on this joint-training setting but addresses a different design choice: how many rewrites to explore and which of them to evaluate before committing to high-fidelity training rollouts.

**Reinforcement learning and selective rollouts for visual generation.** DDPO formulates denoising as sequential decision-making, and DPOK combines diffusion policy optimization with reference regularization (Black et al., 2023; Fan et al., 2023). Flow-GRPO extends online policy-gradient training to flow-matching models through stochastic sampling and reduced training-time denoising steps (Liu et al., 2025). DanceGRPO studies group-relative optimization across visual generation tasks, including training on high- and low-reward candidates selected from larger pools (Xue et al., 2025). Closest to our proxy design, Sol-RL uses low-precision, reduced-step image rollouts to select noise seeds and regenerates the selected seeds in BF16 for optimization (Li et al., 2026). PEAR instead selects textual rewrites, estimates each rewrite's reward from multiple proxy images, and allocates the number of evaluations across original prompts. Its formal rollouts use fresh image feedback rather than preserving a selected proxy noise realization. The distinction is thus the unit of exploration and its budget allocation, not low-precision screening alone.

**Budgeted and multi-fidelity evaluation.** Adaptive resource allocation is central to Hyperband, which combines configuration evaluation with early stopping (Li et al., 2018), and to multi-fidelity bandits, which trade observation cost against approximation fidelity (Kandasamy et al., 2016). PEAR shares the motivation of using inexpensive observations to guide subsequent computation, but its allocation problem is different. Every prompt must retain a fixed-size training group, the proxy fidelity and image count per evaluated rewrite remain fixed, and decisions allocate additional *unseen rewrites* across prompts. We estimate marginal tail-selection utility using empirical resampling (Efron, 1979), rather than eliminating prompts or optimizing cumulative bandit reward. The resulting controller is a practical fixed-budget allocation heuristic; it does not inherit regret or optimal-allocation guarantees from classical bandit algorithms.

# B Additional Experimental Details

## B.1 Implementation Details

We train Qwen3-8B and Stable Diffusion 3.5 Medium on eight GPUs. Image-side LoRA uses rank $r = 32$ and $\alpha = 64$; the text and image learning rates are $1 \times 10^{-6}$ and $3 \times 10^{-4}$, respectively. Text sampling uses temperature 1.0, top-$p = 0.9$, and a maximum length of 1024 tokens. Each batch contains 32 prompts with 96 PEs per prompt. Proxy evaluation uses four independently seeded FP8 images per PE and six denoising steps. Adaptive allocation starts with 24 evaluated PEs per prompt and distributes 768 additional evaluations in eight rounds of 96 evaluations each, using blocks of eight PEs. Marginal gains are estimated with 200 bootstrap trials on the CPU. The total evaluation budget is 1,536 PEs per batch, or 50% of full-pool evaluation. The top-12 and bottom-12 PEs per prompt each receive one fresh BF16 image with ten denoising steps, yielding 768 formal training images per batch. Both generation stages use a resolution of $512 \times 512$. The consistency, realism, and aesthetics reward weights are 0.7, 0.2, and 0.1. The proxy-configuration study varies precision (FP8 or BF16) and denoising steps (4, 6, 8, or 10), with 100 training updates per configuration.

## B.2 Evaluation

We report WISE category and Overall scores (Niu et al., 2025), human-preference scores from HPSv2 (Wu et al., 2023) and PickScore (Kirstain et al., 2023), and VQAScore (Lin et al., 2024) on GenEvalv2 (Kamath et al., 2025). Each checkpoint is evaluated using eight randomly selected, distinct generation seeds, and results are averaged across runs and multiplied by 100. Avg averages WISE Overall, HPSv2, PickScore, and GenEvalv2. The proxy-configuration study uses 1,000 WISE prompts. Gain validation uses a frozen checkpoint and 128 WISE prompts disjoint from training, with 96 PEs per prompt. Each prediction uses 24 initial PEs and 200 bootstrap trials; observed gains average three separate eight-PE extensions of the same initial pool. An independent four-image evaluation of the initial 24 PEs is also included in the cached dataset, giving 36,864 proxy images in total. Predictions are divided into five approximately equal-sized bins, with 95% intervals obtained from 2,000 prompt-level bootstrap resamples. Estimator sensitivity uses candidate pools of 24, 32, and 40 PEs, five resampling seeds, and $T \in \{2, 5, 10, 25, 50, 100, 200, 500, 2000\}$ Monte Carlo trials.

## B.3 Baselines and Ablation Variants

Direct-generation baselines include Stable Diffusion 3.5 Medium and Large (Stability AI, 2024), as well as FLUX.1-dev (Black Forest Labs, 2024). The pre-RL PE pipeline combines Qwen3-8B (Qwen Team, 2025) with Stable Diffusion 3.5 Medium (Stability AI, 2024). Train Text Only and Train Image Only update the corresponding policy while freezing the other. Joint-training baselines are DualGRPO (Kou et al., 2026) and UniGRPO (Liu et al., 2026b). PEAR-NoProxy scores each PE using one training-fidelity image instead of four proxy images. PEAR-Static retains the initial 24 evaluations per prompt and the total 1,536-PE budget, but allocates all remaining evaluations in one round. PEAR-HalfProxy evaluates a fixed pool of 48 PEs per prompt, whereas PEAR-FullProxy evaluates all 96. Under full-pool evaluation, PEAR-Top24, PEAR-Bottom24, and PEAR-Random24 select the highest-scoring, lowest-scoring, and randomly sampled 24 PEs, respectively, instead of top-12 plus bottom-12. The estimator analysis additionally compares gain- and variance-based scores with either dynamic rescoring or fixed initial scores.

# C Additional Qualitative Analysis

Figure 5 examines object identification, knowledge-dependent visual attributes, and cross-location temporal reasoning. *In these examples, PEAR resolves the intended subject or attribute and preserves it from prompt expansion to image generation.* For the handheld optical-device query, the direct-generation baselines depict sighting or single-barrel devices. The Qwen-based pipeline describes a generic optical instrument, while UniGRPO explicitly specifies a monocular. PEAR instead names binoculars and generates two distinct, side-by-side barrels, matching the benchmark's intended object. For the historical counting-tool query, SD3.5-M and FLUX.1-dev produce calculators with displays and keypads, while the Qwen-based pipeline and UniGRPO shift attention toward

Table 3: **Per-iteration runtime breakdown across training variants.** Entries report time in seconds, with the percentage of the corresponding wall-clock time in parentheses. Stage names follow the timing logs. Dashes indicate that no proxy-preview stage is used. The final row reports the number of proxy images generated per training iteration.

| **Stage** | **UniGRPO** 24 PEs; no proxy | **PEAR** Adaptive; FP8/6, $m = 4$ | **PEAR-FullProxy** Full preview; FP8/6, $m = 4$ | **PEAR-NoProxy** Full render; 96 PEs |
|---|---|---|---|---|
| Expand | 37.9 (12.1%) | 148.0 (18.6%) | 135.6 (12.6%) | 153.6 (18.8%) |
| Preview | — | 383.7 (48.2%) | 671.3 (62.3%) | — |
| Reward | 107.8 (34.4%) | 91.3 (11.5%) | 99.4 (9.2%) | 360.4 (44.2%) |
| Update | 151.0 (48.2%) | 151.5 (19.0%) | 152.4 (14.1%) | 243.1 (29.8%) |
| Other | 16.9 (5.4%) | 21.8 (2.7%) | 19.6 (1.8%) | 58.2 (7.1%) |
| **Wall-clock time** | **313.6** | **796.3** | **1078.3** | **815.2** |
| Proxy images / iteration | 0 | 6,144 | 12,288 | 0 |

*Note.* Percentages use the reported wall-clock totals and may not sum to 100% after rounding. The reported stage times for PEAR-NoProxy sum to 815.3 s, compared with its reported wall-clock time of 815.2 s; both are retained as provided.

shops, shopkeepers, and equipment displays. PEAR specifies an abacus and makes its wooden frame, rods, and beads the central subject. This comparison concerns object identity and visual focus, not the accuracy of bead counts. For the plant query, the other methods depict generic purple-flowered plants, whereas PEAR explicitly identifies English lavender and renders slender stems with purple flower spikes. For burning boron, PEAR specifies and depicts the required green flame. In contrast, the direct-generation outputs contain mainly orange-yellow flames; the Qwen-based pipeline specifies blue-green but produces a predominantly blue flame, and UniGRPO specifies and renders a blue-white flame. Finally, for Hong Kong's waterfront when New York is approaching noon, the direct-generation baselines produce daytime scenes, while the Qwen-based pipeline specifies a sunny afternoon and UniGRPO carries over the near-noon condition. PEAR instead infers a nighttime scene, reflected in the dark sky, illuminated skyline, and harbor reflections. These examples highlight the importance of identifying the central subject and its required attributes rather than merely adding contextually associated visual details.

## D Runtime Analysis

Table 3 decomposes per-iteration runtime into expansion, preview, reward, update, and other recorded stages. *Adaptive allocation reduces the dominant proxy-evaluation bottleneck and lowers total iteration time.* Full-pool proxy evaluation generates 12,288 proxy images per iteration and spends 671.3 s on preview, accounting for 62.3% of its wall-clock time. PEAR halves the proxy-image count to 6,144 and reduces preview time to 383.7 s, a reduction of 42.8%. Total iteration time decreases from 1078.3 s to 796.3 s, corresponding to a 26.2% reduction. Preview remains the largest stage for PEAR, but its share falls to 48.2%. Thus, reducing candidate evaluations translates into measured runtime savings, although the reduction in image count is not proportional to the reduction in latency.

*The runtime savings primarily come from selective evaluation rather than cheaper policy updates.* Update times remain close across UniGRPO, PEAR, and PEAR-FullProxy at 151.0 s, 151.5 s, and 152.4 s, respectively. Broader exploration nevertheless introduces additional cost: expansion takes 148.0 s for PEAR compared with 37.9 s for UniGRPO, and PEAR adds a substantial proxy-preview stage. Consequently, PEAR's total iteration time is approximately 2.54 times that of UniGRPO. The efficiency benefit should therefore be understood as reducing the overhead of full-pool exploration, not as making a training iteration faster than the baseline without oversampling.

The no-proxy comparison further shows that *removing the preview stage does not necessarily reduce total runtime.* PEAR-NoProxy has no separate proxy-preview stage, but its recorded reward and update stages require 360.4 s and 243.1 s, accounting for 44.2% and 29.8% of wall-clock time, respectively. Its total iteration time is 815.2 s, compared with 796.3 s for PEAR. PEAR is therefore approximately 2.3% faster in this comparison, despite generating additional proxy images. This re-

sult distinguishes eliminating a dedicated proxy stage from eliminating the overall cost of candidate evaluation and training; it compares the complete configurations rather than isolating the effect of numerical precision alone.

### D.1 Effect of Proxy Image Count on Allocation Efficiency and Stability

**Allocation efficiency.** We measure allocation quality by normalizing the total realized gain achieved by a scoring policy between a random-allocation baseline and an oracle:

$$\text{Efficiency} = 100 \times \frac{G_{\text{policy}} - G_{\text{random}}}{G_{\text{oracle}} - G_{\text{random}}} \quad (\%), \tag{8}$$

where $G_{\text{policy}}$ is the total realized selection-utility gain obtained after spending the evaluation budget, $G_{\text{random}}$ is the corresponding gain of a score-agnostic random-allocation baseline, and $G_{\text{oracle}}$ is the maximum achievable gain of an oracle that knows all realized gains in advance. All three allocations obey the same budget and feasibility constraints. For $G_{\text{oracle}} > G_{\text{random}}$, an efficiency of 0% matches random allocation, while 100% matches the oracle. Negative values indicate performance below random allocation; values cannot exceed 100% because the oracle is an upper bound by construction. Thus, the metric measures the fraction of the achievable improvement over random allocation recovered by the policy, rather than the fraction of the oracle's total gain. In the image-count analysis below, all gains are evaluated using the same full four-image mean-reward reference, irrespective of the number of images used for decision scoring.

**Number of proxy images per PE.** *Reducing the number of proxy images lowers generation cost, but compromises both mean allocation efficiency and robustness to image-sample selection.* As shown in Figure 6, mean allocation efficiency increases monotonically from 75.7% to 76.9%, 81.2%, and 83.5% as $m$ increases from one to four, while the corresponding rerun standard deviations decrease from 4.76 to 3.08, 3.02, and 0.16 percentage points. At a fixed number of evaluated PEs, $m = 2$ halves the number of proxy images at a cost of 6.6 percentage points in mean efficiency. Using $m = 3$ reduces the image count by 25% with a smaller 2.3-point decrease, although its rerun variability remains close to that of $m = 2$. Thus, the third image primarily improves average decision quality rather than eliminating sensitivity to subset selection. The single-image configuration is particularly problematic: supplementary diagnostics show that it inflates two-sided selection utility by 34%, with greater inflation for prompts with higher judge noise, suggesting that random reward fluctuations can be mistaken for differences in PE quality. Moreover, our variance-decomposition implementation requires at least two images to estimate within-PE variance, making the variance-based comparator unavailable at $m = 1$. These results support retaining $m = 4$ as the default, with $m = 3$ and $m = 2$ providing more conservative and more aggressive budget-saving alternatives, respectively, rather than adopting single-image scoring. Importantly, $m = 4$ reuses the same complete image cache in every replay: its small standard deviation reflects Monte Carlo variability conditional on that cache, not reproducibility under independently regenerated images.

## E Analysis of Prompt Expansion

**Post-training inference behavior.** Table 4 compares actual PE outputs from trained UniGRPO and PEAR checkpoints under matched queries, sample indices, and saved decoding settings. *In these paired examples, PEAR more faithfully expresses the intended object identity and preserves the explicit spatial constraint.* For the military observation-device query, UniGRPO specifies a monocular, whereas PEAR names a pair of binoculars. Both descriptions fit the broad handheld military-observation context, but PEAR matches the binoculars target specified by the WISE reference. For the lion query, UniGRPO replaces "under" with "on" and reinforces the reversed relation by placing the lion's head on an armrest. PEAR retains "under" and consistently restates that the chair is above the lion and that the lion is beneath it. These are text-level comparisons: they establish neither the correctness of every added detail nor the fidelity of subsequent image generation. The inference examples also use different queries from the offline selection examples, so they do not trace individual candidates from selection to post-training behavior.

Table 4: **Post-training PE comparison at inference.** Each pair contains actual outputs from trained UniGRPO and PEAR checkpoints, matched by query, sample index, and saved decoding settings. Full PE texts are retained; boldface highlights differences in object identity and spatial relations.

| UniGRPO | PEAR |
|---|---|
| **Case 1: Object identity.**<br>*Query:* The hand-held optical device soldiers use to observe distant targets from afar | |
| A soldier's hand gripping a sleek, matte-black military-grade **monocular** with a focused lens, set against a gritty battlefield backdrop. The device features a durable rubber grip, a tactical strap, and a clear objective lens reflecting distant terrain, emphasizing its purpose for long-range observation. | A soldier holding **a pair of rugged, military-grade binoculars** with a matte black body, rubberized grip, and a tactical strap, focused on observing distant targets through the lenses in a battlefield setting. The binoculars feature a compact, durable design with a magnifying lens and a handle for one-handed use. |
| **Case 2: Spatial relation.**<br>*Query:* a lion under a pink chair | |
| A realistic lion with a muscular, golden-brown coat and intense golden eyes lounges comfortably **on a vibrant pink velvet chair**, its head resting on a cushioned armrest, surrounded by soft natural light filtering through a lush green forest backdrop, emphasizing the contrast between the rich hues of the lion's fur and the bold pink chair. | A realistic lion with golden fur resting **under a large pink chair** in a room setting, the **chair positioned above the lion** with the lion appearing comfortable and calm **beneath it**. |

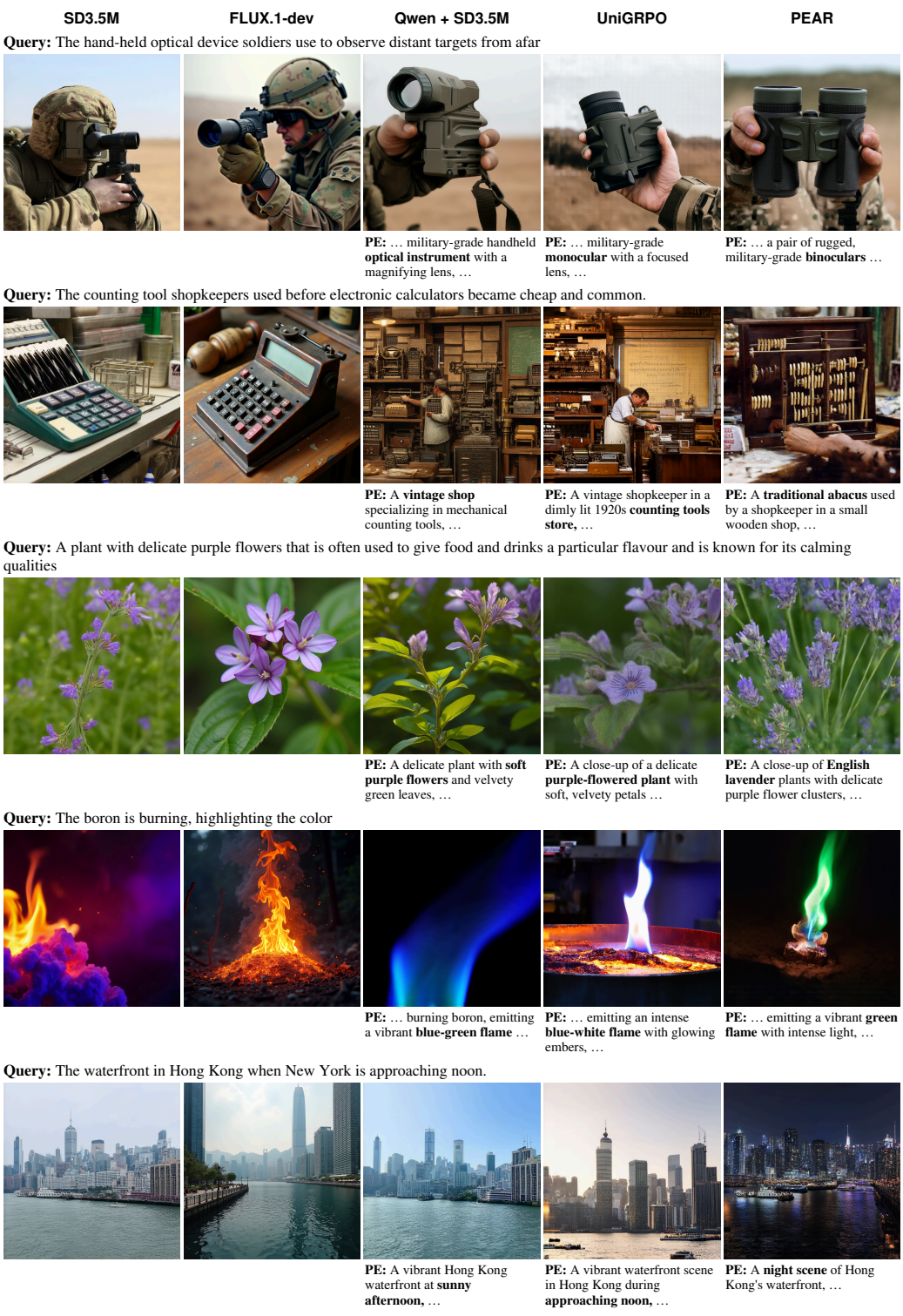



Figure 5: **Additional qualitative comparisons on knowledge-grounded generation.** From top to bottom, the benchmark queries target binoculars, an abacus, lavender, a green flame for burning boron, and Hong Kong's waterfront at night inferred from the New York time cue. PEAR makes the intended subjects or attributes explicit in its prompt expansions and reflects them in the generated images. Selected PE excerpts are shown beneath the corresponding outputs.

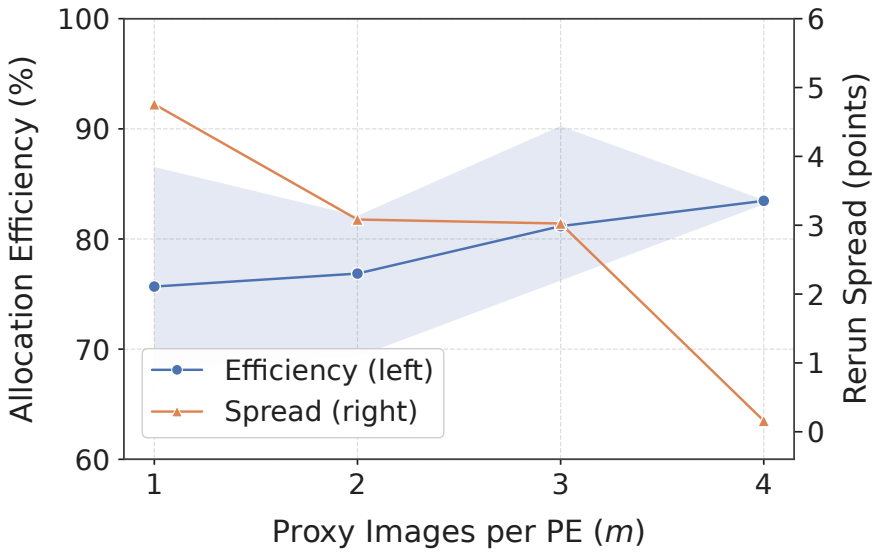



Figure 6: **Effect of the number of proxy images per PE on allocation efficiency and rerun variability.** Using the same cached dataset, we reconstruct PE scores for $m = 1, 2, 3$ by averaging randomly sampled subsets of the four image-reward columns; $m = 4$ uses all four images. All configurations are evaluated against the same four-image mean-reward reference. We combine 20 subset draws with five Monte Carlo seeds, yielding 100 offline allocation replays per configuration without additional image generation. Blue circles (left axis) show mean allocation efficiency, defined in Equation equation 8, where 0% and 100% correspond to random and oracle allocation, respectively. Orange triangles (right axis) show the standard deviation across replays, measured in efficiency percentage points. For $m = 4$, the image set remains unchanged across all replays.