REALLOCATE, RESCALE, RETAIN: TEST-TIME LATENT CONTROL FOR IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Optimizing continuous text and image latents at test time can improve image generation in unified multimodal large language models without updating pretrained parameters. Yet these latent states evolve during optimization, while the policy controlling their refinement often remains fixed. This mismatch can waste updates on locally saturated positions, ignore differences in update reliability, and discard better intermediate images by returning the final iterate. We introduce **TRILO** (Triple-Control Latent Optimization), a state-dependent method that controls *where* to refine, *how* to scale update signals, and *which* evaluated image to return. *Reallocate* uses predictive entropy to shift a fixed-size text-latent refinement window away from local saturation. *Rescale* adjusts update signals within the selected window, applying more conservative scaling at less confident positions. *Retain* returns the highest-reward evaluated image if the stopping criterion remains unmet after budget exhaustion, without additional generation or reward evaluation. Together, these controls adapt how a fixed optimization budget is deployed rather than increasing it. Local theoretical analysis and controlled diagnostics support these design principles. Experiments on GenEval, T2I-CompBench, and WISE demonstrate consistent improvements in compositional and knowledge-intensive generation on Janus-Pro-1B and Janus-Pro-7B. With benchmark evaluators as online rewards and matched maximum update budgets, TRILO achieves higher scores and 12.8–24.0% lower measured end-to-end runtime than a fixed-policy baseline across all three benchmarks on the 7B model.

1 INTRODUCTION

High visual fidelity does not guarantee faithful instruction following. Despite substantial progress in diffusion and autoregressive image generation (Rombach et al., 2022; Sun et al., 2024; Esser et al., 2024), generated images can still violate compositional relations, numerical constraints, or knowledge-dependent requirements (Huang et al., 2023; Niu et al., 2026). Motivated by advances in reasoning with large language models (Wei et al., 2022; OpenAI et al., 2024; DeepSeek-AI et al., 2025), recent work introduces planning, reflection, and iterative refinement into image generation (Guo et al., 2025; Jiang et al., 2025; Zhang et al., 2025; Li et al., 2025b; Zhuo et al., 2025). Unified multimodal large language models, such as Janus (Chen et al., 2025), offer a complementary route through their continuous text and image hidden states. MILR (Mi et al., 2026) demonstrates that these latents can be optimized using reward feedback at test time while all pretrained parameters remain frozen. This makes the hidden states of an individual generation, rather than the shared model weights, the target of adaptation. However, identifying *what can be optimized* leaves a separate question unanswered: *how should that optimization be controlled as inference proceeds?*

A fixed model does not imply fixed refinement needs. Within a single optimization trajectory, some latent positions become locally saturated, update signals differ in reliability, and a later decoded image need not improve upon an earlier one. Yet the baseline policy in MILR uses a fixed refinement support, uniform multiplicative scaling of the selected update signals, and terminal-state fallback when the stopping criterion remains unmet. These choices can repeatedly allocate updates to weakly responsive positions, overlook differences in signal reliability, and discard higher-reward images already generated. Simply continuing the same updates does not address this mismatch between evolving latent states and fixed control decisions.

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.

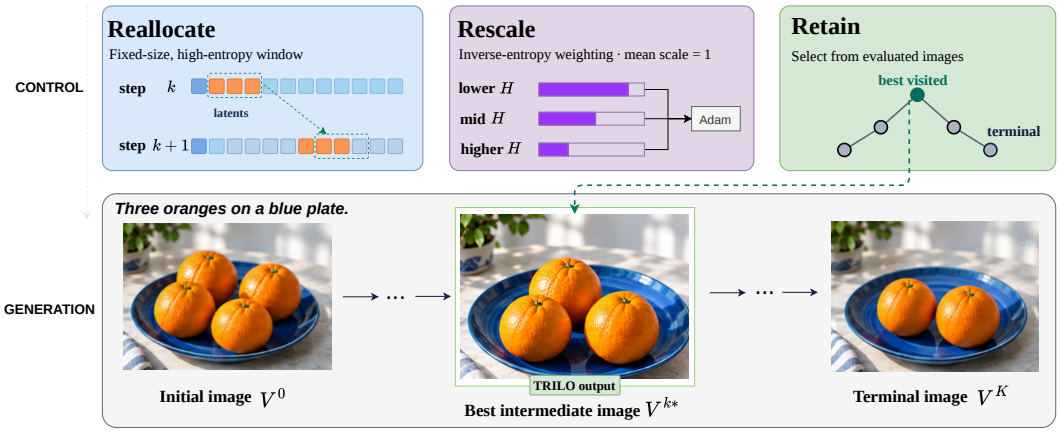

Figure 1: Reallocate, Rescale, Retain. Top: adaptive window selection, relative signal scaling, and selection among evaluated images. Bottom: an illustrative trajectory in which an intermediate image matches the requested object count, while the terminal image loses this alignment.

We address this mismatch with **TRILO**, a training-free framework for state-dependent test-time latent optimization. It adapts three complementary decisions: *where* to refine, *how* to scale update signals, and *which* evaluated image to return. Figure 1 illustrates these controls and how an intermediate image can better satisfy the instruction than the terminal output.

Reallocate: where to refine. Repeatedly updating the same text positions can concentrate computation on locally saturated regions. We therefore select the fixed-size contiguous text-latent window with the highest mean predictive entropy at each active step. This allows refinement to follow the evolving uncertainty landscape while keeping the number of updated positions unchanged.

Rescale: how to update. Positions within the selected window can differ in update reliability. We use entropy as an approximate ordering signal and apply inverse-entropy weighting, giving less confident positions more conservative relative scales while preserving a mean multiplier of one. The two controls thus assign uncertainty distinct roles: it guides allocation away from saturation and encourages caution when weighting the selected update signals.

Retain: which image to return. Later iterates need not improve upon earlier images. When the budget is exhausted without meeting the stopping criterion, we return the highest-reward image among all evaluated candidates. This preserves the best observed reward without additional generation or reward evaluation.

Together, these controls adapt refinement with frozen model parameters, unchanged per-step refinement sizes and maximum update budgets, and the original fixed-prefix image-latent refinement rule. Our local theoretical analysis motivates reallocation through score saturation and conservative scaling through an idealized model of gradient noise. These results provide design principles without establishing optimality of the practical entropy-based policy.

We evaluate TRILO on GenEval (Ghosh et al., 2023), T2I-CompBench (Huang et al., 2023), and WISE (Niu et al., 2026), covering compositional and knowledge-intensive generation. TRILO consistently improves over the frozen Janus-Pro-1B and Janus-Pro-7B backbones. On 7B, using benchmark evaluators as online rewards and matching refinement ratios, stopping rules, and maximum update budgets, TRILO outperforms the MILR rerun on all three benchmarks with 12.8–24.0% lower measured end-to-end runtime. Gains over the backbone persist with practical alternative rewards, while ablations and state- and trajectory-level diagnostics support the complementary benefits of adaptive allocation, signal scaling, and output retention.

2 Related Work

Reasoning-Augmented Image Generation. Reasoning-augmented image generation improves instruction following through planning, verification, and revision. Prior work explores prompt decomposition and refinement (Wang et al., 2024; Gani et al., 2024; Zhan et al., 2024), chain-of-thought and interleaved textual reasoning (Jiang et al., 2025; Guo et al., 2026; Zhang et al., 2025;

Tong et al., 2025), and iterative image reflection (Yang et al., 2024; Guo et al., 2025; Li et al., 2025b; Zhuo et al., 2025). Supervised and reinforcement learning further develop generation and reasoning capabilities (Duan et al., 2026; Liu et al., 2025; Yan et al., 2026). We study a complementary intervention: controlling reward-guided updates to continuous hidden states of a frozen unified multimodal large language model, without additional parameter training.

Latent Reasoning and Test-Time Latent Optimization. Latent reasoning performs intermediate computation in continuous representations, including recurrent hidden states, latent thinking tokens, and compressed reasoning (Hao et al., 2024; Cheng & Van Durme, 2024; Shen et al., 2025; Xu et al., 2025). For inference-time latent optimization, LatentSeek (Li et al., 2025a) uses reward-guided policy-gradient updates, while GradCuit (Yu et al., 2026) introduces intermediate-layer latents for direct gradient credit assignment from the generated continuation. Visual and multimodal extensions also explore implicit reasoning states (Li et al., 2026a; Dong et al., 2026). For image generation, LatentMorph (Chen et al., 2026) combines learned latent reasoning modules with an RL-trained invocation policy. MILR (Mi et al., 2026), our closest formulation, directly optimizes existing text and image latents through reward-guided test-time updates without modifying pretrained parameters. We retain its frozen-generator setting and optimization variables, while adapting refinement control to evolving latent states and optimization trajectories.

Test-Time Scaling and Adaptive Control. Test-time scaling improves outputs through sampling, search, verification, or refinement, while adaptive inference allocates computation according to confidence, uncertainty, or task difficulty (Zhao et al., 2025; Shi et al., 2025). Image-generation approaches explore candidate selection and noise-trajectory search (Ma et al., 2025; Ramesh & Mardani, 2025; Zhang et al., 2026b). dMLLM-TTS (Xin et al., 2026) combines trajectory exploration with self-verified refinement, RAISE (Jiang et al., 2026) uses requirement-driven evolutionary revision, and TexTailor (Li et al., 2026b) tailors block-wise textual guidance within diffusion transformers. Our focus is control within a reward-guided latent optimization trajectory. At the optimizer level, AdaGrad and Adam adapt updates using gradient statistics (Duchi et al., 2011; Kingma & Ba, 2015). TRILO complements this adaptation: *Reallocate* selects active text-latent positions, and *Rescale* weights their update signals before Adam. At the trajectory level, *Retain* selects the highest-reward evaluated image when the stopping criterion remains unmet after budget exhaustion, without additional decoding or reward evaluation. Together, these controls adapt support, signal scaling, and output selection under fixed per-step refinement and maximum iteration budgets.

3 METHOD

TRILO controls reward-guided latent optimization at two levels: *Reallocate* and *Rescale* adapt text-latent updates to the current state, while *Retain* selects the output from the evaluated trajectory (Figure 2). All pretrained parameters remain frozen, and both the per-step refinement sizes and maximum update budget are unchanged.

3.1 LATENT OPTIMIZATION AND CONTROL

Given an instruction c, a unified multimodal large language model generates M textual reasoning tokens followed by N image tokens. Let $\mathbf{z}^{(t)} = (z_1^{(t)}, \dots, z_M^{(t)})$ and $\mathbf{z}^{(v)} = (z_1^{(v)}, \dots, z_N^{(v)})$ denote their continuous hidden states, with $\mathbf{z} = [\mathbf{z}^{(t)}; \mathbf{z}^{(v)}]$. Following MILR (Mi et al., 2026), we refine these instance-specific latents to improve the expected image reward:

$$\mathbf{z}^\star \in \arg\max_{\mathbf{z}} J(\mathbf{z}), \qquad J(\mathbf{z}) = \mathbb{E}_{V \sim p(\cdot|\mathbf{z},c)}[R(V,c)], \tag{1}$$

where V is a decoded image and $R(V,c)$ is the optimization reward. Text refinement is restricted to the original M reasoning positions; no new text tokens are appended.

Reward-guided update signal. At step k, decoding $\mathbf{z}^k$ produces an image V_k with reward $R_k = R(V_k, c)$. For a text or image latent block z_i^k, the underlying update signal is

$$s_i^k = R_k \nabla_{z_i^k} \log p\big(y_i^k \mid z_i^k, \mathcal{C}_i^k\big), \tag{2}$$

where $\mathcal{C}_i^k$ denotes the remaining conditioning context and $y_i^k \in \arg\max_y p(y \mid z_i^k, \mathcal{C}_i^k)$ is recomputed at each step. The reward and greedy token are held fixed during differentiation. This specifies the implemented score signal, without assuming it is an unbiased estimator of $\nabla_{z_i^k} J(\mathbf{z}^k)$.

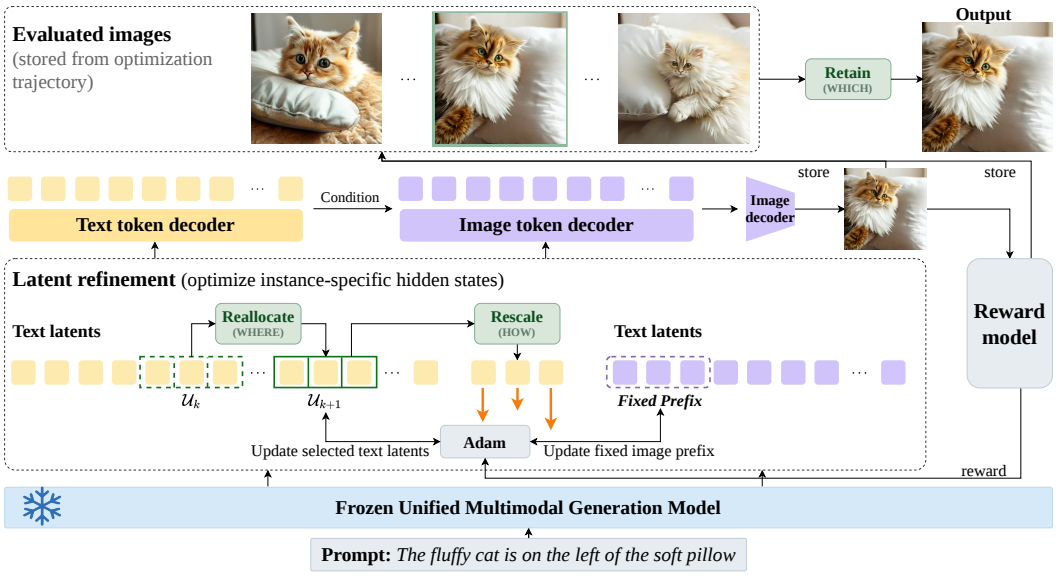

Figure 2: Overview of TRILO. (a) At step k, *Reallocate* selects a fixed-size text window by mean predictive entropy, and *Rescale* weights its update signals before Adam. Image latents use a fixed prefix with unit signal scaling. **(b)** If the stopping criterion remains unmet after budget exhaustion, *Retain* returns the highest-reward image already evaluated along the trajectory.

Support and signal control. TRILO separates selection of the active positions from weighting of their update signals:

$$d_i^k = m_i^k \alpha_i^k, \qquad \widetilde{g}_i^k = d_i^k s_i^k, \tag{3}$$

where $m_i^k \in \{0,1\}$ is a support mask and $\alpha_i^k > 0$ is a relative signal multiplier. Adam (Kingma & Ba, 2015) uses $\widetilde{g}_i^k$ to update the active latent variables. The multipliers therefore act on the signals supplied to Adam, rather than directly specifying latent displacements.

Budget and stopping. We allow at most K latent updates and evaluate each state before updating it. If $R_k > \tau$ for the task-specific threshold τ, we return V_k immediately; otherwise, we update when $k < K$. Thus, at most $K+1$ images are evaluated, including V_0. If the threshold remains unmet after evaluating V_K, *Retain* selects the output (Section 3.3).

3.2 Adaptive Text-Latent Refinement

At each active step, the frozen language head maps text latent $z_i^{(t),k}$ to a distribution p_i^k over vocabulary $\mathcal{V}$. We compute its predictive entropy as

$$H_i^k = -\sum_{a \in \mathcal{V}} p_i^k(a) \log p_i^k(a), \tag{4}$$

with $0 \log 0 = 0$. This entropy profile informs both support selection and relative signal weighting.

Reallocate: where to refine. Low-entropy positions can exhibit weak local score responses (Section 4), motivating reallocation as the latent state evolves. For a refinement ratio $0 < \lambda_t \leq 1$, let $L_t = \max\{1, \lfloor \lambda_t M \rfloor\}$. We select the contiguous window with the highest mean entropy:

$$\begin{aligned} s_k^\star &\in \arg\max_{1 \leq s \leq M - L_t + 1} \frac{1}{L_t} \sum_{i=s}^{s+L_t-1} H_i^k, \\ \mathcal{U}_k &= \{s_k^\star, \ldots, s_k^\star + L_t - 1\}, \qquad m_i^k = \mathbb{I}[i \in \mathcal{U}_k]. \end{aligned} \tag{5}$$

Recomputing $\mathcal{U}_k$ at every active step allows the refinement location to change while preserving $|\mathcal{U}_k| = L_t$. Maximum-entropy selection is an empirical allocation rule; it does not assume that entropy ranks realized reward gains. We compare support-selection strategies in Appendix C.3.2.

Rescale: how to weight the selected signals. Selecting a window does not imply that all its update signals are equally reliable. Motivated by the idealized analysis of signal reliability in Section 4, we

assign higher-entropy positions no larger relative multipliers. For $\epsilon > 0$, define the stabilized mean entropy

$$\bar{H}_k^\epsilon = \max\left\{ \frac{1}{|\mathcal{U}_k|} \sum_{j \in \mathcal{U}_k} H_j^k, \epsilon \right\}. \tag{6}$$

We clip the inverse-entropy ratios to fixed positive bounds $0 < \alpha_{\min} \le \alpha_{\max}$, then normalize their mean:

$$\begin{aligned} \widetilde{\alpha}_i^k &= \begin{cases} \mathrm{clip}\left(\dfrac{\bar{H}_k^\epsilon}{H_i^k}, \alpha_{\min}, \alpha_{\max} \right), & H_i^k > 0, \\ \alpha_{\max}, & H_i^k = 0, \end{cases} \\ \alpha_i^k &= \frac{\widetilde{\alpha}_i^k}{\frac{1}{|\mathcal{U}_k|} \sum_{j \in \mathcal{U}_k} \widetilde{\alpha}_j^k}, \qquad i \in \mathcal{U}_k. \end{aligned} \tag{7}$$

Positive clipping bounds keep the denominator nonzero and ensure $|\mathcal{U}_k|^{-1} \sum_{i \in \mathcal{U}_k} \alpha_i^k = 1$. Normalization preserves the mean multiplier, not the joint-update norm. Outside $\mathcal{U}_k$, set $\alpha_i^k = 1$; the support mask already excludes these positions from refinement.

Entropy serves distinct roles in the two controls: *Reallocate* directs computation away from local saturation, while *Rescale* applies more conservative relative weighting to less confident positions within the chosen window. The latter uses entropy as a heuristic ordering proxy, without estimating gradient noise or specifying an optimal Adam update. Diagnostics and scaling ablations are provided in Appendices C.4.1 and C.3.3.

Image refinement and optimizer state. Image latents retain a fixed prefix of length $L_v = \lfloor \lambda_v N \rfloor$, with $0 < \lambda_v \le 1$, and unit signal scaling. When the text window changes, its Adam optimizer is re-instantiated over the newly selected variables; the fixed image prefix uses a persistent optimizer. Further settings appear in Appendix B.1.

3.3 Trajectory-Aware Output Selection

Retain: which image to return. A later iterate need not improve upon an earlier image. If the stopping criterion remains unmet after all K updates, we select the highest-reward evaluated candidate:

$$k^\star \in \arg\max_{0 \le k \le K} R_k, \qquad \widehat{V} = V_{k^\star}. \tag{8}$$

Including both V_0 and V_K ensures $R_{k^\star} = \max_{0 \le k \le K} R_k \ge R_K$. We return the stored image itself, requiring no additional decoding or reward evaluation and leaving the optimization trajectory unchanged. This guarantee concerns recorded optimization rewards on the same trajectory; performance under an independent evaluator is an empirical question.

4 Theoretical Analysis

We distinguish properties of the implemented control rules from an idealized analysis of signal scaling. Full assumptions and proofs are provided in Appendix A.

Reallocate: local score saturation. Fix the conditioning context and consider the greedy score s_i in Eq. 2. If $|R| \le R_{\max}$ and the local logit Jacobian has operator norm at most κ_i, then

$$\|s_i\|_2^2 \le 2R_{\max}^2 \kappa_i^2 \left(1 - e^{-H_i}\right)^2, \tag{9}$$

where H_i is predictive entropy with natural logarithms. Under uniform bounds on $R_{\max}$ and κ_i, the score therefore vanishes as $H_i \to 0$. This motivates monitoring local saturation, but does not imply that higher-entropy windows yield larger reward gains (Appendix A.1).

Rescale: alignment and signal energy. For a selected block, consider the idealized direct update $z_i^+ = z_i + \eta \alpha_i \xi_i$ under an L-smooth objective J, with $\eta, L > 0$. Conditional on the current state and support, define $g_i = \nabla_{z_i} J$, $a_i = \langle g_i, \mathbb{E}[\xi_i] \rangle$, and $b_i = \mathbb{E}\|\xi_i\|_2^2 \in (0, \infty)$. Independently maximizing its quadratic expected-progress lower bound over $\alpha_i \ge 0$ gives

$$\alpha_i^{\mathrm{ind}} = \frac{[a_i]_+}{L\eta b_i}, \qquad [x]_+ = \max\{x, 0\}. \tag{10}$$

Table 1: Main results on GenEval and T2I-CompBench. TRILO achieves the highest overall score on GenEval and ranks second overall on T2I-CompBench, ahead of all other reported test-time reasoning methods.

Method	GenEval				T2I-CompBench					
	Count.↑	Position↑	Attr.↑	Overall↑	Color↑	Shape↑	Texture↑	Spatial↑	Complex↑	Overall↑
Non-reasoning Models										
BAGEL (Deng et al., 2025)	0.81	0.64	0.63	0.82	0.8081	0.5794	0.7154	0.3403	0.3778	0.5220
Qwen-Image (Wu et al., 2025)	0.89	0.76	0.77	0.87	0.8280	0.5986	0.7451	0.4430	0.3838	0.5526
Training-based Reasoning Models										
Flow-GRPO (Liu et al., 2025)	0.95	0.99	0.86	0.95	0.8379	0.6130	0.7236	0.5447	–	–
CoR-Painter (Yan et al., 2026)	0.63	0.90	0.77	0.87	0.8447	0.6194	0.7693	**0.6384**	0.4118	0.5997
Qwen-Image (+ELDiff) (Pan et al., 2026)	0.90	0.80	0.79	0.90	0.8525	0.6167	0.7617	0.4855	0.4024	0.5724
Janus-Pro-7B (+LatentMorph) (Chen et al., 2026)	–	0.98	0.92	0.96	0.8404	0.6946	0.7989	0.5093	**0.6360**	**0.6458**
Test-time Reasoning Models										
Janus-Pro-1B (+MILR) (Mi et al., 2026)	0.78	0.86	0.86	0.89	0.6066	0.2796	0.4177	0.2796	0.2613	0.3512
Janus-Pro-7B (+MILR) (Mi et al., 2026)	0.90	0.98	0.91	0.95	0.8508	0.5117	0.6949	0.4613	0.3684	0.5325
Janus-Pro-7B (+TwiG-ZS) (Guo et al., 2026)	–	0.84	0.67	0.86	0.7311	0.4155	0.6477	0.2198	0.3665	0.4483
Qwen-Image (+TexTailor) (Li et al., 2026b)	0.86	0.75	0.80	0.88	0.8677	0.6348	0.7796	0.4560	0.4104	0.5768
Janus-Pro-1B (Chen et al., 2025)	0.51	0.65	0.56	0.73	0.3411	0.2261	0.2696	0.0968	0.2721	0.2478
Janus-Pro-1B+TRILO	0.94	0.93	0.95	0.96	0.8054	0.5438	0.7019	0.3864	0.4089	0.5262
Janus-Pro-7B (Chen et al., 2025)	0.56	0.77	0.64	0.78	0.6359	0.3528	0.4936	0.2061	0.3559	0.3921
Janus-Pro-7B+TRILO	**1.00**	**1.00**	**0.98**	**0.99**	**0.8883**	**0.7127**	**0.8211**	0.5085	0.4619	0.6207

Thus, alignment and signal energy jointly determine the preferred scale; in the unbiased special case, it decreases with the noise-to-signal ratio. This motivates relative weighting, but does not derive the implemented inverse-entropy rule: entropy does not determine these moments, mean normalization couples the scales, and Adam changes the resulting displacement. The general bound, constrained analysis, and exact properties of the practical multipliers appear in Appendices A.2.1–A.2.5.

Retain: recorded-reward dominance. On a budget-exhausted trajectory, the selection rule in Section 3.3 cannot return a lower recorded reward than the terminal image (Appendix A.3). Transfer to independent evaluators remains an empirical question.

5 Experiments

Our evaluation asks whether adapting latent control improves generation without increasing the permitted refinement budget, and examines how the three controls contribute to the observed gains.

5.1 Experimental Settings

Benchmarks and Baselines. We evaluate TRILO on GenEval (Ghosh et al., 2023), T2I-CompBench (Huang et al., 2023), and WISE (Niu et al., 2026), covering text-to-image alignment, compositional generation, and knowledge-intensive generation. Comparisons include non-reasoning generators, training-based reasoning methods, and test-time reasoning methods. Complete benchmark-wise results are provided in Appendix C.1.

Implementation Details. We use Janus-Pro-1B and Janus-Pro-7B (Chen et al., 2025), with the 7B model for controlled analyses unless otherwise stated. All pretrained parameters remain frozen. Latent optimization uses Adam (Kingma & Ba, 2015) with learning rate 0.03, refinement ratios $\lambda_t = 0.2$ and $\lambda_v = 0.02$, and at most $K = 20$ joint text and image latent updates. The default generation seed is 41. Further settings are given in Appendix B.

5.2 Main Results

Improved Generation Under Matched Budgets. On Janus-Pro-7B, TRILO improves over the matched MILR rerun from 0.9506 to 0.9932 on GenEval, from 0.5399 to 0.6207 on T2I-CompBench, and from 0.6098 to 0.7105 on WISE. These comparisons keep both the refinement ratios and the maximum update budget fixed. All three prompt-paired 95% bootstrap confidence intervals for the score differences are strictly positive (Appendix C.9). The advantage also holds for each of the three evaluated generation seeds on all three benchmarks (Appendix C.10), providing evidence beyond a single generation run.



Figure 3: Qualitative comparison across generation tasks. TRILO more faithfully satisfies explicit compositional constraints and implicit scene requirements in these examples, such as generating exactly three oranges and depicting a fogged mirror after a hot shower.

Table 2: Component and design ablations on Janus-Pro-7B. (a) Reallocate, Rescale, and Retain each improve the matched baseline, while their combination achieves the best or tied-best scores across all three benchmarks at $K = 20$. **(b)** Uncertainty-guided reallocation and inverse-entropy rescaling outperform the tested alternatives, highlighting the importance of both support selection and state-specific signal weighting.

(a) Component Contributions

Variant	GenEval ↑	T2I-Comp. ↑	WISE ↑
Vanilla (Chen et al., 2025)	0.7800	0.3921	0.3500
MILR (matched) (Mi et al., 2026)	0.9506	0.5399	0.6098
w/ Reallocate	0.9857	0.5464	0.6430
w/ Rescale	0.9873	0.5422	0.6180
w/ Retain	0.9512	0.5630	0.6124
w/ Reallocate + Rescale	**0.9932**	0.5764	0.6730
w/ Reallocate + Retain	0.9857	0.5847	0.6680
TRILO	**0.9932**	**0.6207**	**0.7105**

(b) Reallocate and Rescale Strategies

Type	Strategy	GenEval ↑	T2I ↑	WISE ↑
Reallocate	Fixed Prefix	0.9873	0.6046	0.5486
	Random Window	0.9819	0.6108	0.5773
	Avoid-Low + Random	0.9892	0.6164	0.5826
	Top-20% Entropy + Random	0.9856	0.6189	0.6936
	Uncertainty-Guided	**0.9932**	**0.6207**	**0.7105**
Rescale	Uniform	0.9857	0.5847	0.6680
	Entropy-Proportional	0.9928	0.5492	0.5807
	Shuffled Inv.-Entropy	0.9873	0.5580	0.5831
	Inverse-Entropy	**0.9932**	**0.6207**	**0.7105**

Performance Across Scales and Generation Tasks. Table 1 places these gains in the context of published methods. Within this comparison, TRILO achieves the highest overall GenEval score and the highest Color, Shape, and Texture scores on T2I-CompBench. Its T2I-CompBench aggregate ranks second at 0.6207, below the training-based LatentMorph result of 0.6458. The benefits also extend to Janus-Pro-1B, which reaches 0.96, 0.5262, and 0.50 on GenEval, T2I-CompBench, and WISE, respectively, improving the corresponding backbone on all three. Additional evaluation on T2I-CompBench++ yields 0.6390 for the 7B model (Appendix C.2). Figure 3 illustrates generation outcomes across the evaluated instruction types.

5.3 Understanding Adaptive Latent Optimization

Control Contributions Across Budgets. Table 2(a) compares partial policies with matched MILR at $K = 20$; the full method is best or tied-best across all three benchmarks. Without *Retain*, *Reallocate* and *Rescale* already improve T2I-CompBench and WISE from 0.5399 and 0.6098 to 0.5764 and 0.6730. Adding *Retain* raises these scores to 0.6207 and 0.7105, while GenEval remains at 0.9932. Across budget sweeps on both scales, the full policy remains strongest or competitive with the best partial configuration. At $K = 16$, it exceeds the published MILR references in all six model–benchmark combinations; these literature comparisons are separate from the matched $K = 20$ reruns.

Reallocate: Where to Refine. Table 2(b) varies the support-selection rule while keeping the remaining control settings fixed. Randomly moving a same-sized window gives mixed results relative to a fixed prefix. Excluding low-entropy regions before random selection improves all three benchmarks, and selecting the maximum-entropy window performs best among the tested rules. Thus, the benefit is not explained by changing the window alone; how its location is chosen also matters.

Figure 4 examines the underlying variation in refinement opportunity. In panel (a), the lowest-entropy quintile has only 2.5%–5.3% of the mean normalized score norm of the remaining positions, measured before masking, rescaling, and Adam on nonzero-response steps. Panels (b,c) use standardized one-step interventions from the same saved state, holding the image update and decoding seed fixed,

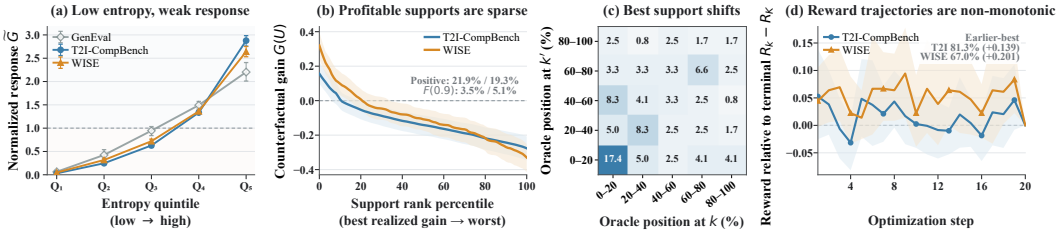

Figure 4: State and trajectory diagnostics for TRILO. **(a)** The lowest-entropy latents contribute only a small fraction of the normalized local response. **(b)** Exhaustive one-step interventions show that only 21.9% of T2I-CompBench supports and 19.3% of WISE supports yield positive gains under the same refinement budget. **(c)** The best realized support changes markedly across optimization states on T2I-CompBench. **(d)** Among budget-exhausted trajectories, an earlier iterate outperforms the terminal one in 81.3% of T2I-CompBench and 67.0% of WISE cases. Together, these results reveal a heterogeneous and evolving latent optimization landscape, motivating adaptive control rather than fixed allocation policies.

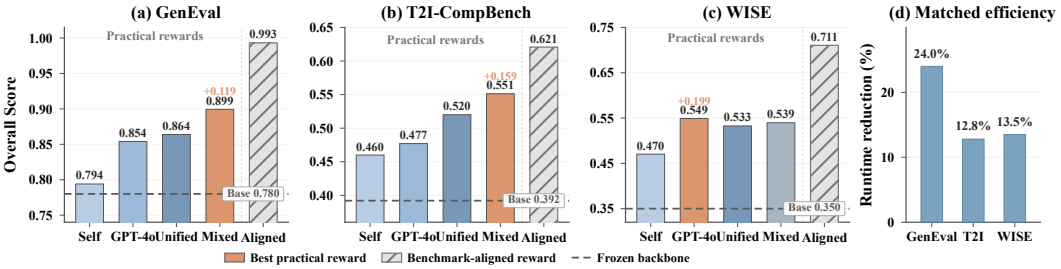

Figure 5: Robustness to online feedback and measured efficiency. **(a–c)** Across four practical reward models, TRILO improves the frozen Janus-Pro-7B backbone on all three benchmarks. **(d)** Under matched benchmark-aligned settings, TRILO reduces measured end-to-end runtime by 24.0%, 12.8%, and 13.5%. These results show robustness beyond direct benchmark feedback and improve the measured quality–cost trade-off under the controlled benchmark-aligned setting.

with uniform text scaling and fresh Adam state. On average, only 21.9% and 19.3% of candidate windows yield positive realized gains on T2I-CompBench and WISE, respectively; the best realized window also shifts across states on T2I-CompBench. These diagnostics support nonuniform, evolving refinement opportunities, not the stronger claim that entropy identifies the best realized update (Appendix C.4).

Rescale: How to Scale Updates. Within the uncertainty-guided window, Table 2(b) compares uniform, entropy-proportional, shuffled inverse-entropy, and inverse-entropy signal scaling. The shuffled control preserves the computed multiplier values but permutes their assignment to selected positions. Inverse-entropy scaling performs best on all three benchmarks; reversing or shuffling the assignment substantially degrades T2I-CompBench and WISE performance. These comparisons support the importance of state-dependent multiplier assignment rather than nonuniform weighting alone. They concern the implemented signal-scaling policy, not optimal step sizes or a direct estimate of gradient noise (Appendix C.3.3).

Retain: Which Image to Return. Figure 4(d) examines trajectories that exhaust all $K = 20$ updates. Among these trajectories, an intermediate optimized image strictly outperforms the terminal image in 81.3% of T2I-CompBench cases and 67.0% of WISE cases. This diagnostic excludes the initial image, whereas the actual retention rule also considers it. The effect is not assessed solely by the optimized reward: an independent Qwen3.5-35B-A3B evaluator, used only after generation and selection, gives a positive mean retained-minus-terminal score difference across all 294 budget-exhausted WISE trajectories (+0.139, 95% CI $[+0.082, +0.194]$; Appendix C.4). This provides post-hoc evidence for retaining earlier gains without requiring additional generation during inference.

Robustness to Online Rewards and Final Evaluators. Figure 5(a–c) compares four practical online-reward configurations: SelfReward, GPT-4o, UnifiedReward, and MixedReward, while keeping official benchmark evaluation of the final images. All four improve the frozen backbone on every benchmark. MixedReward reaches 0.8994 on GenEval and 0.5513 on T2I-CompBench, while GPT-4o reaches 0.5490 on WISE. The strongest practical reward varies by benchmark, and scores remain below the benchmark-aligned reference (Appendix C.7). Separately, re-scoring the same WISE images with an alternative final evaluator preserves the ordering TRILO > matched MILR > frozen backbone on both scales, although absolute scores and margins change (Appendix C.8).

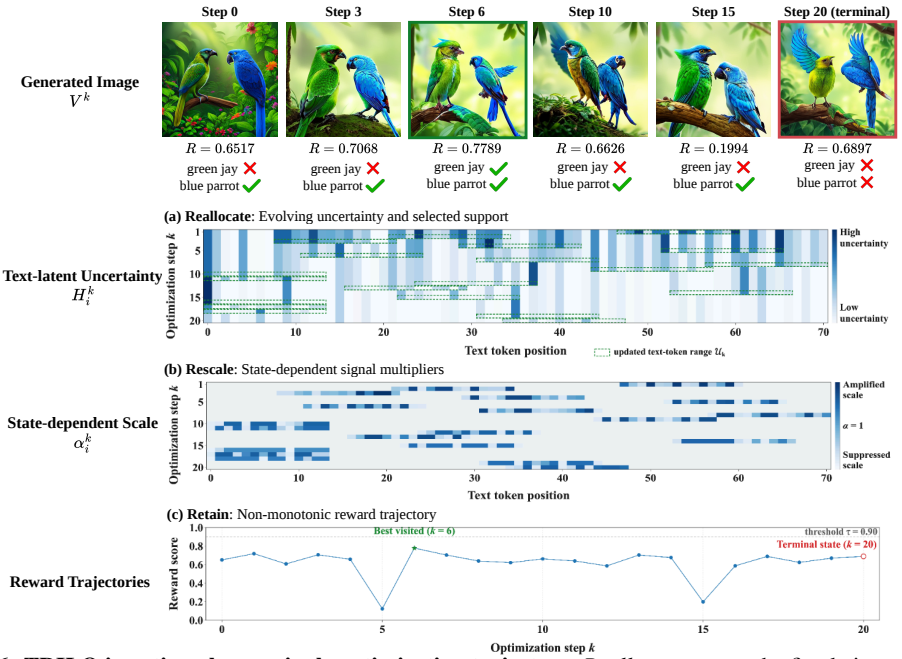

Figure 6: TRILO in action along a single optimization trajectory. *Reallocate* moves the fixed-size text-latent support with the evolving uncertainty landscape, while *Rescale* assigns state-dependent signal multipliers within the selected support. The reward trajectory is non-monotonic, so *Retain* returns the highest-reward visited image rather than the terminal output when the stopping criterion remains unmet.

Measured Performance and Efficiency. Under matched benchmark-aligned settings on the same hardware, TRILO improves all three benchmark scores while reducing measured end-to-end runtime by 24.0%, 12.8%, and 13.5% on GenEval, T2I-CompBench, and WISE, respectively (Figure 5(d)). Timing includes initial generation, latent optimization, and benchmark evaluation. The number of executed updates does not explain all runtime differences: on T2I-CompBench, the mean changes only from 11.682 to 11.614 steps. We therefore report an empirical performance and runtime advantage under the measured protocol, without attributing it to faster convergence or unmeasured pipeline stages (Appendix C.11).

TRILO Along a Single Optimization Trajectory. Figure 6 illustrates how the changing entropy profile moves the refinement window, while signal multipliers vary across positions and steps. The best image appears at an intermediate step; subsequent refinement reduces reward and prompt satisfaction. With the stopping threshold still unmet, *Retain* recovers this earlier image, illustrating how the three controls operate along one trajectory.

Limitations. The evaluation covers two Janus-Pro scales, not multiple generator architectures, and the primary matched comparisons use benchmark-aligned feedback. Alternative rewards and post-hoc evaluators provide complementary checks but do not establish reward-independent performance. Entropy remains a heuristic control signal, and adaptive allocation and scaling are confined to text latents. Generalization to other generator families and adaptive image-latent control remains to be tested.

6 Conclusion

We introduced TRILO, a training-free framework for image generation that treats test-time latent optimization as an evolving process to control. *Reallocate* adapts the text-latent refinement window, *Rescale* weights the selected update signals, and *Retain* returns the highest-reward evaluated image when the stopping criterion remains unmet after budget exhaustion. Together, these operations adapt refinement decisions while keeping all pretrained multimodal large language model parameters frozen and the permitted refinement budgets unchanged. Experiments across three benchmarks and two Janus-Pro scales show consistent improvements over the frozen backbones. Matched comparisons on the 7B model demonstrate higher scores and lower measured runtime than MILR under benchmark-aligned feedback, while improvements over the backbone persist with alternative online rewards.

Ethics Statement

This work does not involve the collection of new human-subject data or personally identifiable information. Experiments use existing benchmarks and pretrained models under their applicable licenses. As with other text-to-image generation methods, TRILO may inherit biases or limitations from pretrained models and reward evaluators, and generated content may be misused. Appropriate safeguards and compliance with relevant licenses should therefore be considered.

Reproducibility Statement

The main paper and appendix provide the method description, experimental settings, implementation details, evaluation protocols, ablations, and theoretical analysis needed to reproduce our results. Complete derivations and proofs are included in the appendix. We will release the implementation, experiment configurations, and evaluation scripts.

AI Use Statement

Generative AI tools were used solely to assist with language editing, including grammar correction, sentence-level polishing, and improving the clarity and readability of the manuscript. They were not used to formulate the research questions, design the methodology, conduct experiments, generate or modify experimental results, perform data analysis, or draw scientific conclusions. All technical content, mathematical derivations, experimental results, interpretations, and final manuscript decisions were produced and verified by the authors, who take full responsibility for the content of this work.

References

Black Forest Labs. FLUX.2: Frontier visual intelligence, November 2025. Official model release.

Harold Haodong Chen, Xinxiang Yin, Wen-Jie Shu, Hongfei Zhang, Zixin Zhang, Chenfei Liao, Litao Guo, Qifeng Chen, and Ying-Cong Chen. Show, don't tell: Morphing latent reasoning into image generation. In *ICML*, 2026.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-Pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*, 2024.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li,

Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, Yan Li, Yubo Wang, Zhijie Cao, Zhiqian Lin, Zhitao Yang, Zhongang Cai, Yuwei Niu, Yue Zhu, Bo Liu, Chengguang Lv, Haojia Yu, Haozhe Xie, Hongli Wang, Jianan Fan, Jiaqi Li, Jiefan Lu, Jingcheng Ni, Junxiang Xu, Kaihuan Liang, Lianqiang Shi, Linjun Dai, Linyan Wang, Oscar Qian, Peng Gao, Pengfei Liu, Qingping Sun, Rui Shen, Ruisi Wang, Shengnan Ma, Shuang Yang, Siyi Xie, Siying Li, Tianbo Zhong, Xiangli Kong, Xuanke Shi, Yang Gao, Yongqiang Yao, Yves Wang, Zhengqi Bai, Zhengyu Lin, Zixin Yin, Wenxiu Sun, Ruihao Gong, Quan Wang, Lewei Lu, Lei Yang, Ziwei Liu, and Dahua Lin. SenseNova-U1: Unifying multimodal understanding and generation with neo-unify architecture, 2026. URL https://arxiv.org/abs/2605.12500.

Shuai Dong, Siyuan Wang, Xingyu Liu, Chenglin Li, Haowen Hou, and Zhongyu Wei. Interleaved latent visual reasoning with selective perceptual modeling. In *ACL*, pp. 29316–29335, 2026.

Chengqi Duan, Rongyao Fang, Yuqing Wang, Kun Wang, Linjiang Huang, Xingyu Zeng, Hongsheng Li, and Xihui Liu. GoT-R1: Unleashing reasoning capability of autoregressive visual generation with reinforcement learning. In *ICLR*, 2026.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12(7), 2011.

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 *ICML*, 2024.

Hanan Gani, Shariq Farooq Bhat, Muzammal Naseer, Salman Khan, and Peter Wonka. LLM blueprint: Enabling text-to-image generation with complex and detailed prompts. In *ICLR*, 2024.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*, volume 36, pp. 52132–52152, 2023.

Google DeepMind. Nano banana pro: Gemini 3 pro image, November 2025. Official model release.

Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Rui Huang, Haoquan Zhang, Manyuan Zhang, Jiaming Liu, Shanghang Zhang, Peng Gao, Hongsheng Li, and Pheng-Ann Heng. Can we generate images with CoT? let's verify and reinforce image generation step by step. *arXiv preprint arXiv:2501.13926*, 2025.

Ziyu Guo, Renrui Zhang, Hongyu Li, Manyuan Zhang, Xinyan Chen, Sifan Wang, Yan Feng, Peng Pei, and Pheng-Ann Heng. Thinking-while-generating: Interleaving textual reasoning throughout visual generation. In *CVPR*, pp. 26295–26305, 2026.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

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*, volume 36, pp. 78723–78747, 2023.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *TPAMI*, 2025.

Achin Jain, Jie An, Siddharth Chaudhary, and Davide Modolo. Mural: Transferring LLM knowledge to image generation via mixture-of-transformers. *arXiv preprint arXiv:2606.29013*, 2026.

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. In *NeurIPS*, volume 38, pp. 39856–39890, 2025.

Liyao Jiang, Ruichen Chen, Chao Gao, and Di Niu. RAISE: Requirement-adaptive evolutionary refinement for training-free text-to-image alignment. In *CVPR*, pp. 22038–22048, 2026.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. In *ICLR*, volume 2026, pp. 148076–148090, 2026a.

Binglei Li, Mengping Yang, Zhiyu Tan, Junping Zhang, and Hao Li. TexTailor: Inference-time textual guidance tailoring for multimodal diffusion transformers. In *ECCV*, 2026b.

Hengli Li, Chenxi Li, Tong Wu, Xuekai Zhu, Yuxuan Wang, Zhaoxin Yu, Eric Hanchen Jiang, Song-Chun Zhu, Zixia Jia, Ying Nian Wu, and Zilong Zheng. Seek in the dark: Reasoning via test-time instance-level policy gradient in latent space, 2025a.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Arsh Koneru, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Reflect-DiT: Inference-time scaling for text-to-image diffusion transformers via in-context reflection. In *ICCV*, pp. 15657–15668, 2025b.

Zixuan Li, Haokun Lin, Yicheng Xiao, Zhiwei Li, Xinyang Song, Zelong Zheng, Yong He, Heng Yao, Ke Ding, Chao Yu, et al. IV-CoT: Implicit visual chain-of-thought for structure-aware text-to-image generation. *arXiv preprint arXiv:2606.24849*, 2026c.

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. In *NeurIPS*, 2025.

Zhiheng Liu, Weiming Ren, Haozhe Liu, Zijian Zhou, Shoufa Chen, Haonan Qiu, Xiaoke Huang, Zhaochong An, Fanny Yang, Aditya Patel, et al. TUNA: Taming unified visual representations for native unified multimodal models. In *CVPR*, pp. 15740–15751, 2026.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

Yapeng Mi, Yanpeng Zhao, Hengli Li, Chenxi Li, Huimin Wu, Xiaojian Ma, Song-Chun Zhu, Ying Nian Wu, and Qing Li. MILR: Improving multimodal image generation via test-time latent reasoning. In *ICLR*, 2026.

Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Chaoran Feng, Fanqing Meng, Kun-Peng Ning, Bin Zhu, and Li Yuan. WISE: World knowledge-informed semantic evaluation for text-to-image generation. In *ICML*, 2026.

Yoonjin Oh, Yongjin Kim, Hyomin Kim, Donghwan Chi, and Sungwoong Kim. OSPO: Object-centric self-improving preference optimization for text-to-image generation. In *CVPR*, pp. 7620–7631, 2026.

OpenAI. Introducing GPT-Image-1.5, December 2025. Official model release.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Qingtao Pan, Kai Ye, Zhihao Dou, Bing Ji, and Shuo Li. Eldiff: When evidential learning meets text-to-image diffusion. In *ECCV*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Vignav Ramesh and Morteza Mardani. Test-time scaling of diffusion models via noise trajectory search. In *NeurIPS*, pp. 87284–87317, 2025. doi: 10.52202/085713-2919.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10674–10685, 2022.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. Codi: Compressing chain-of-thought into continuous space via self-distillation. *arXiv preprint arXiv:2502.21074*, 2025.

Zhenning Shi, Yijia Zhu, Yi Xie, Junhan Shi, Guorui Xie, Haotian Zhang, Yong Jiang, Congcong Miao, and Qing Li. Reasoning under uncertainty: Efficient llm inference via unsupervised confidence dilution and convergent adaptive sampling. In *EMNLP*, pp. 32192–32206, 2025.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation, 2024. URL https://arxiv.org/abs/2406.06525.

Tencent Hunyuan Foundation Model Team. Hunyuanimage 3.0 technical report, 2026. URL https://arxiv.org/abs/2509.23951.

Chengzhuo Tong, Ziyu Guo, Renrui Zhang, Wenyu Shan, Xinyu Wei, Zhenghao Xing, Hongsheng Li, and Pheng-Ann Heng. Delving into RL for image generation with cot: A study on DPO vs. GRPO. In *NeurIPS*, 2025.

Dianyi Wang, Ruihang Li, Feng Han, Chaofan Ma, Wei Song, Siyuan Wang, Yibin Wang, Yi Xin, Hongjian Liu, Zhixiong Zhang, Shengyuan Ding, Tianhang Wang, Zhenglin Cheng, Tao Lin, Cheng Jin, Kaicheng Yu, Jingjing Chen, Wenjie Wang, Zhongyu Wei, and Jiaqi Wang. DeepGen 1.0: A lightweight unified multimodal model for advancing image generation and editing. *arXiv preprint arXiv:2602.12205*, 2026a.

Linqing Wang, Zhiyong Xu, Ximing Xing, Yiji Cheng, Zhiyuan Zhao, Donghao Li, Tiankai Hang, Zhenxi Li, Jiale Tao, Qixun Wang, et al. PromptEnhancer: Taming your rewriter for text-to-image generation via fine-grained reward. In *CVPR*, pp. 14895–14904, 2026b.

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

Zhenyu Wang, Enze Xie, Aoxue Li, Zhongdao Wang, Xihui Liu, and Zhenguo Li. Divide and conquer: Language models can plan and self-correct for compositional text-to-image generation. *arXiv preprint arXiv:2401.15688*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

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*, 2025.

Yicheng Xiao, Lin Song, Yukang Chen, Yingmin Luo, Yuxin Chen, Yukang Gan, Wei Huang, Xiu Li, Xiaojuan Qi, and Ying Shan. MindOmni: Unleashing reasoning generation in vision language models with rgpo. In *NeurIPS*, volume 38, pp. 88786–88810, 2025.

Yi Xin, Siqi Luo, Tianxiang Xu, Qi Qin, Haoxing Chen, Kaiwen Zhu, Zhiwei Zhang, Yangfan He, Rongchao Zhang, Jinbin Bai, et al. dMLLM-TTS: Self-verified and efficient test-time scaling for diffusion multi-modal large language models. In *CVPR*, pp. 35726–35735, 2026.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs. In *ACL*, pp. 23336–23351, 2025.

Ruxue Yan, Xubo Liu, Wenya Guo, Zhengkun Zhang, Ying Zhang, and Xiaojie Yuan. From" what" to" how": Constrained reasoning for autoregressive image generation. *arXiv preprint arXiv:2603.02712*, 2026.

Zhengyuan Yang, Jianfeng Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. Idea2Img: Iterative self-refinement with GPT-4V for automatic image design and generation. In *ECCV*, pp. 167–184, 2024.

Zhaoxin Yu, Qi Shen, Hengli Li, Zhaowei Zhang, Song-Chun Zhu, Chi Zhang, and Zilong Zheng. GradCuit: Credit-assigned gradient flow enables robust and interpretable test-time latent reasoning. *arXiv preprint arXiv:2608.02585*, 2026.

Jingtao Zhan, Qingyao Ai, Yiqun Liu, Yingwei Pan, Ting Yao, Jiaxin Mao, Shaoping Ma, and Tao Mei. Prompt refinement with image pivot for text-to-image generation. In *ACL*, pp. 941–954, 2024.

Huichao Zhang, Liao Qu, Yiheng Liu, Hang Chen, Yangyang Song, Yongsheng Dong, Shikun Sun, Xian Li, Xu Wang, Yi Jiang, Hu Ye, Bo Chen, Yiming Gao, Peng Liu, Akide Liu, Zhipeng Yang, Qili Deng, Linjie Xing, Jiyang Liu, Zhao Wang, Yang Zhou, Mingcong Liu, Yi Zhang, Qian He, Xiwei Hu, Zhongqi Qi, Jie Shao, Zhiye Fu, Shuai Wang, Fangmin Chen, Xuezhi Chai, Zhihua Wu, Yitong Wang, Zehuan Yuan, Daniel K. Du, and Xinglong Wu. NextFlow: Unified sequential modeling activates multimodal understanding and generation. *arXiv preprint arXiv:2601.02204*, 2026a.

Xiangcheng Zhang, Haowei Lin, Haotian Ye, James Y Zou, Jianzhu Ma, Yitao Liang, and Yilun Du. Inference-time scaling of diffusion models through classical search. In *ICLR*, volume 2026, pp. 39133–39166, 2026b.

Yu Zhang, Yunqi Li, Yifan Yang, Rui Wang, Yuqing Yang, Dai Qi, Jianmin Bao, Dongdong Chen, Chong Luo, and Lili Qiu. ReasonGen-R1: Cot for autoregressive image generation models through SFT and RL, 2025. URL https://arxiv.org/abs/2505.24875.

Zekai Zhang, Jiahao Li, Jie Zhang, Kaiyuan Gao, Kun Yan, Lihan Jiang, Ningyuan Tang, Shengming Yin, Tianhe Wu, Xiaoyue Chen, Xiao Xu, Yan Shu, Yanran Zhang, Yixian Xu, Yuxiang Chen, Zhendong Wang, Zihao Liu, Zikai Zhou, Huishuai Zhang, Dongyan Zhao, and Chenfei Wu. Qwen-Image-Agent: Bridging the context gap in real-world image generation, 2026c. URL https://arxiv.org/abs/2606.26907.

Bing Zhao, Chenfei Wu, Deqing Li, Hao Meng, Jiahao Li, Jie Zhang, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kuan Cao, Kun Yan, Liang Peng, Lihan Jiang, Niantong Li, Ningyuan Tang, Shengming Yin, Tianhe Wu, Xiao Xu, Xiaoyue Chen, Xihua Wang, Yan Shu, Yanran Zhang, Yi Wang, Yilei Chen, Ying Ba, Yixian Xu, Yujia Wu, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhendong Wang, Zihao Liu, Zikai Zhou, An Yang, Chen Cheng, Chenxu Lv, Dayiheng Liu, Fan Zhou, Hantian Xiong, Hongzhu Shi, Hu Wei, Huihong Zhao, Ivy Liu, Jianwei Zhang, Jiawei Zhang, Kai Chen, Kang He, Levon Xue, Lin Qu, Linhan Tang, Luwen Feng, Minggang Wu, Minmin Sun, Na Ni, Rui Men, Shuai Bai, Sishou Zheng, Tao Lan, Tianqi Zhang, Tingkun Wen, Wei Wang, Weixu Qiao, Weiyi Lu, Wenmeng Zhou, Xiaodong Deng, Xiaoxiao Xu, Xinlei Fang, Xionghui Chen, Yanan Wang, Yang Fan, Yichang Zhang, Yixuan Xu, Yu Wu, Zhiyuan Ma, and Zhizhi Cai. Qwen-image-2.0 technical report, 2026. URL https://arxiv.org/abs/2605.10730.

Zhengyi Zhao, Shubo Zhang, Zezhong Wang, Huimin Wang, Yutian Zhao, Bin Liang, Yefeng Zheng, Binyang Li, Kam-Fai Wong, and Xian Wu. T^2: An adaptive test-time scaling strategy for contextual question answering. In *EMNLP*, pp. 3731–3756, 2025.

Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *ICCV*, pp. 15329–15339, 2025.

A Analysis and Proofs

We separate three levels of analysis: an exact bound on the fixed-context greedy score, an idealized local model of signal scaling, and exact properties of the implemented scaling and output-selection rules. The idealization provides design guidance; it does not establish global convergence or optimality of the entropy-based policy. We follow the order of the controls in Section 4.

A.1 Reallocate: Predictive Entropy and Greedy-Score Saturation

Fix a text-latent block z_i and its conditioning context. Let $\boldsymbol{\ell}_i = f_i(z_i)$, $\mathbf{p}_i = \mathrm{softmax}(\boldsymbol{\ell}_i)$, and $y_i \in \arg\max_a p_i(a)$. Write $H_i = -\sum_a p_i(a) \log p_i(a)$ for its entropy, using natural logarithms and $0 \log 0 = 0$, and let $A_i = \partial \boldsymbol{\ell}_i / \partial z_i$ be the logit Jacobian. As in Eq. 2, the score is $s_i = R \nabla_{z_i} \log p_i(y_i)$, with the realized reward and greedy token held fixed during differentiation.

Proposition A.1 (Low-entropy bound for the greedy score)**.** *If $\|A_i\|_{\mathrm{op}} \le \kappa_i$ and $|R| \le R_{\max}$ almost surely, then*

$$\|s_i\|_2^2 \le 2R_{\max}^2 \kappa_i^2 \big(1 - p_i(y_i)\big)^2 \le 2R_{\max}^2 \kappa_i^2 \big(1 - e^{-H_i}\big)^2. \tag{11}$$

Proof. The softmax identity and chain rule give $s_i = R A_i^\top (\mathbf{e}_{y_i} - \mathbf{p}_i)$. Writing $p_y = p_i(y_i)$,

$$\|\mathbf{e}_{y_i} - \mathbf{p}_i\|_2^2 = (1 - p_y)^2 + \sum_{a \ne y_i} p_i(a)^2 \le (1 - p_y)^2 + \left(\sum_{a \ne y_i} p_i(a) \right)^2 = 2(1 - p_y)^2. \tag{12}$$

This establishes the first inequality after applying the reward and Jacobian bounds. Since y_i is greedy, $p_i(a) \le p_y$ for every a, so $H_i \ge -\log p_y$ and hence $p_y \ge e^{-H_i}$. Substitution proves the second inequality. □

Corollary A.2 (Low-entropy saturation)**.** *If $R_{\max}$ and κ_i remain uniformly bounded over the considered states, then $H_i \to 0$ implies $\|s_i\|_2 \to 0$.*

This implication is one-sided: higher entropy relaxes the bound but does not rank realized score norms, post-update reward gains, or refinement supports. It concerns the score before Adam, without asserting token correctness, vanishing Adam displacement, or absence of downstream refinement opportunity.

Reward offsets affect the implemented signal. For an unshifted score S and constant offset δ, let $R = S + \delta$ and $h_i = \nabla_{z_i} \log p_i(y_i)$. Although $J_R = J_S + \delta$ leaves the objective gradient unchanged, the greedy score changes as $s_i^{(R)} = s_i^{(S)} + \delta h_i$. The fixed-greedy signal has no general baseline-cancellation identity. The moments analyzed below therefore use the implemented reward convention, including the -1 offset in the primary experiments. An offset preserves candidate rankings; it preserves stopping events only when the threshold is shifted accordingly.

A.2 Rescale: Local Signal-Scaling Analysis

A.2.1 Local progress with a possibly biased signal

Fix the current latent state $\mathbf{z} = \mathbf{z}^k$, partitioned into disjoint coordinate blocks $(z_1, \dots, z_n)$, and let $J(\mathbf{z}) = \mathbb{E}_{V \sim p(\cdot|\mathbf{z},c)}[R(V, c)]$. The instruction is fixed. Let $\mathcal{F}_k$ contain the state and controls before drawing the analyzed signal, and write $\mathbb{E}_k[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_k]$. For each signal block ξ_i, define

$$g_i = \nabla_{z_i} J(\mathbf{z}), \qquad \mu_i = \mathbb{E}_k[\xi_i], \qquad a_i = \langle g_i, \mu_i \rangle, \qquad b_i = \mathbb{E}_k \|\xi_i\|_2^2 < \infty. \tag{13}$$

No unbiasedness or independence between blocks is assumed. The finite controls $d_i = m_i \alpha_i \ge 0$, with $m_i \in \{0, 1\}$, are $\mathcal{F}_k$-measurable.

Assumption A.3 (Local smoothness)**.** For some $L > 0$, J has an L-Lipschitz gradient on a region containing the analyzed update segments almost surely.

Consider the idealized direct update

$$z_i^+ = z_i + \eta d_i \xi_i, \qquad \eta > 0. \tag{14}$$

This surrogate isolates signal control from Adam dynamics.

Proposition A.4 (Local progress under a possibly biased signal)**.** *Under Assumption A.3,*

$$\mathbb{E}_k[J(\mathbf{z}^+) - J(\mathbf{z})] \geq \eta \sum_i d_i a_i - \frac{L\eta^2}{2} \sum_i d_i^2 b_i \equiv \mathcal{B}(\mathbf{d}). \tag{15}$$

Proof. Let Δ concatenate the perturbations $\eta d_i \xi_i$. Smoothness gives $J(\mathbf{z}+\Delta) - J(\mathbf{z}) \geq \nabla J(\mathbf{z})^\top \Delta - (L/2)\|\Delta\|_2^2$. Since the controls and current state are fixed under $\mathbb{E}_k$, the conditional expectation of the linear term is $\eta \sum_i d_i \langle g_i, \mu_i \rangle$. Disjoint coordinates give $\|\Delta\|_2^2 = \eta^2 \sum_i d_i^2 \|\xi_i\|_2^2$. Taking expectations establishes the bound without requiring independent blocks. □

All scaling comparisons below keep the state, conditional signal law, L, and η fixed. They optimize the quadratic expression $\mathcal{B}$; its expected-progress interpretation applies only to candidate updates covered by Assumption A.3.

A.2.2 Independent scales and the role of signal bias

For a selected block, the contribution to the bound is

$$\mathcal{B}_i(\alpha_i) = \eta a_i \alpha_i - \frac{L\eta^2}{2} b_i \alpha_i^2, \qquad \alpha_i \geq 0. \tag{16}$$

Proposition A.5 (Independent maximizer of the local bound)**.** *If $b_i > 0$, the unique maximizer of Eq. 16 is*

$$\alpha_i^{\mathrm{ind}} = \frac{[a_i]_+}{L\eta b_i}, \qquad [x]_+ = \max\{x, 0\}. \tag{17}$$

If $b_i = 0$, the contribution is zero for every scale, and we may choose $\alpha_i^{\mathrm{ind}} = 0$.

Proof. For $b_i > 0$, the derivative is $\eta a_i - L\eta^2 b_i \alpha_i$ and the second derivative is strictly negative. The stationary point is feasible when $a_i > 0$; otherwise the maximum is at zero. If $b_i = 0$, then $\xi_i = 0$ almost surely under the conditional law, giving $\mu_i = 0$ and $a_i = 0$. □

Corollary A.6 (Unbiased special case)**.** *If $\mu_i = g_i \neq 0$, define $\sigma_i^2 = \mathbb{E}_k\|\xi_i - g_i\|_2^2$. Then*

$$\alpha_i^{\mathrm{ind}} = \frac{1}{L\eta(1+q_i)}, \qquad q_i = \frac{\sigma_i^2}{\|g_i\|_2^2}. \tag{18}$$

Proof. Unbiasedness gives $a_i = \|g_i\|_2^2$ and $b_i = \|g_i\|_2^2 + \sigma_i^2$. Substitute into Eq. 17. □

The preferred independent scale decreases with q_i in this special case. For a possibly biased signal with $\mu_i \neq 0$, write $v_i = \mathbb{E}_k\|\xi_i - \mu_i\|_2^2$ and $\widetilde{q}_i = v_i / \|\mu_i\|_2^2$. Since $b_i = \|\mu_i\|_2^2 + v_i$, the general scale is

$$\alpha_i^{\mathrm{ind}} = \frac{\rho_i}{L\eta(1+\widetilde{q}_i)}, \qquad \rho_i = \frac{[\langle g_i, \mu_i \rangle]_+}{\|\mu_i\|_2^2}. \tag{19}$$

Thus, variability relative to the mean signal does not identify the preferred scale without its alignment with the true gradient. If $\mu_i = 0$, we may take the preferred scale to be zero.

A.2.3 Coupling under mean normalization

For a fixed finite, nonempty support $\mathcal{U}$, consider

$$\max_{\alpha_i \geq 0} \quad \sum_{i \in \mathcal{U}} \mathcal{B}_i(\alpha_i) \quad \text{subject to} \quad \sum_{i \in \mathcal{U}} \alpha_i = |\mathcal{U}|. \tag{20}$$

This isolates the practical mean-one constraint without imposing clipping, an entropy-based functional form, or Adam dynamics.

Proposition A.7 (Mean-constrained maximizer of the local bound)**.** *If $b_i > 0$ for every $i \in \mathcal{U}$, the unique optimizer is*

$$\alpha_i^{\mathrm{mean}} = \frac{[a_i - \nu]_+}{L\eta b_i}, \qquad \sum_{i \in \mathcal{U}} \frac{[a_i - \nu]_+}{L\eta b_i} = |\mathcal{U}|, \tag{21}$$

where $\nu < \max_i a_i$ is uniquely determined by the equality.

Proof. The objective is strictly concave on a nonempty compact simplex, so its maximizer exists and is unique. The KKT stationarity and complementary-slackness conditions give $\alpha_i = [a_i - \nu]_+/(L\eta b_i)$, where $\eta\nu$ is the multiplier for the sum constraint. The left-hand sum is continuous and strictly decreasing for $\nu < \max_i a_i$, tending to infinity as $\nu \to -\infty$ and to zero as $\nu \uparrow \max_i a_i$. It therefore equals $|\mathcal{U}|$ at a unique ν. □

In the unbiased case with $g_i \neq 0$, this becomes $\alpha_i^{\text{mean}} = [1 - \nu/\|g_i\|_2^2]_+/[L\eta(1+q_i)]$. Mean normalization thus introduces dependence on gradient magnitude and the shared multiplier, beyond q_i alone. Normalizing the independent optima does not generally solve this constrained problem.

A.2.4 Exact properties of the implemented rule

Fix a finite, nonempty support $\mathcal{U}$ and suppress the step index in Eq. 7. Let $D = |\mathcal{U}|^{-1} \sum_{i\in\mathcal{U}} \widetilde{\alpha}_i$ denote the mean raw multiplier.

Proposition A.8 (Algebraic properties of Rescale)**.** *For $0 < \epsilon \leq \alpha_{\min} \leq \alpha_{\max}$, the implemented multipliers satisfy*

$$\begin{aligned} &H_i \geq H_j \implies \alpha_i \leq \alpha_j, \qquad i,j \in \mathcal{U}, \\ &\frac{1}{|\mathcal{U}|}\sum_{i\in\mathcal{U}} \alpha_i = 1, \qquad \frac{\alpha_{\min}}{\alpha_{\max}} \leq \alpha_i \leq \frac{\alpha_{\max}}{\alpha_{\min}}. \end{aligned} \tag{22}$$

Proof. Every raw multiplier lies in $[\alpha_{\min}, \alpha_{\max}]$, so D is positive and lies in the same interval. For a fixed entropy profile, H^ϵ is common and positive; inverse entropy is non-increasing, including the definition $\widetilde{\alpha}_i = \alpha_{\max}$ at $H_i = 0$. Clipping and division by the common positive D preserve this weak ordering. The mean identity follows from the definition of D, and dividing the raw-multiplier bounds by the bounds on D gives the stated interval for the normalized multipliers. □

The denominator guard $\max\{D, \epsilon\}$ is inactive under this condition because $D \geq \alpha_{\min} \geq \epsilon$, as in our configuration. The clipping bounds apply to raw multipliers; normalization enforces unit mean, without fixing the joint signal norm or Adam displacement.

Relation to the idealized scales. For $b_i > 0$, define $r_i = [a_i]_+/b_i$. Agreement in direction with the independent idealization along entropy comparisons is ensured by the additional condition $H_i \geq H_j \Rightarrow r_i \leq r_j$. Clipping may introduce ties, and entropy alone does not establish this condition. For a biased signal, Eq. 19 also shows why an entropy–variability association alone is insufficient: alignment remains unobserved. The practical policy is therefore evaluated through the controlled ablations in Appendix C.3.3.

A.2.5 Signal scaling and Adam displacement

The practical method supplies $m_i^k \alpha_i^k s_i^k$ to Adam, as in Eq. 3. The resulting displacement depends on both moment estimates, their history, and numerical stabilization, so it is not generally proportional to α_i^k.

For example, consider the first step of standard Adam with zero initial moments, bias correction, no weight decay, learning rate η, and numerical stabilizer $\varepsilon_A > 0$. In a selected coordinate with score s and multiplier $\alpha > 0$, the update magnitude is

$$|\Delta z| = \eta \frac{\alpha|s|}{\alpha|s| + \varepsilon_A} = \eta \frac{|s|}{|s| + \varepsilon_A/\alpha}. \tag{23}$$

When $\alpha|s| \gg \varepsilon_A$, the multiplier largely cancels in this first-step normalization. This special case is relevant because a changed text window re-instantiates the text optimizer, while the fixed image prefix preserves optimizer state. It does not characterize all subsequent steps, but illustrates why the effect of Rescale must be assessed with the implemented optimizer rather than inferred from the direct-update surrogate.

A.3 RETAIN: RECORDED-REWARD DOMINANCE

Consider a realized budget-exhausted trajectory with stored images $V_0, \ldots, V_K$ and recorded rewards $R_0, \ldots, R_K$. The strict stopping condition has remained unmet, so $\max_{0 \le k \le K} R_k \le \tau$. Let

$$k^\star = \min \arg \max_{0 \le k \le K} R_k,$$

and return $\widehat{V} = V_{k^\star}$, matching the practical Retain rule in Section 3.3.

Proposition A.9 (Recorded-reward dominance)**.** *The retained image satisfies*

$$R_{k^\star} = \max_{0 \le k \le K} R_k \ge R_K. \tag{24}$$

Proof. The terminal image belongs to the maximization set. □

The statement is pathwise and requires no smoothness or gradient assumptions. Returning the stored image adds no generation or reward-evaluation calls and leaves the trajectory unchanged. The guarantee applies only to the recorded scores: it does not order the expected rewards of the associated latents, fresh reward realizations, or independent-evaluator scores. The independent-evaluator analysis in Appendix C.4.4 tests the latter empirically.

B EXPERIMENTAL DETAILS

B.1 IMPLEMENTATION DETAILS

We instantiate TRILO on Janus-Pro-1B and Janus-Pro-7B, using the 7B model for controlled analyses unless otherwise specified. All pretrained parameters remain frozen. Latent optimization uses Adam with learning rate 0.03, text and image refinement ratios $\lambda_t = 0.2$ and $\lambda_v = 0.02$, and at most $K = 20$ joint text–image update steps.

Rescale uses $\epsilon = 10^{-8}$, $\alpha_{\min} = 0.5$, and $\alpha_{\max} = 2.0$ across benchmarks and model scales without task-specific tuning. For numerical stability, we clamp the support-wise mean entropy and the final scale-normalization denominator from below by ϵ; the latter safeguard is inactive under the strictly positive clipping range $\alpha_{\min} = 0.5 > 0$.

The text optimization pool contains only the M original reasoning positions; no new text tokens are appended. At each active step, *Reallocate* selects a contiguous support of fixed size

$$L_t = \max\{1, \lfloor \lambda_t M \rfloor\},$$

changing where, but not how many, text positions are optimized per step. For image latents, we retain a fixed prefix of size

$$L_v = \lfloor \lambda_v N \rfloor.$$

Janus-Pro uses $N = 576$ image positions, giving $L_v = 11$ for $\lambda_v = 0.02$. The image branch uses unit signal scaling and applies neither entropy-guided Reallocate nor Rescale.

Adam uses $\beta_1 = 0.9$, $\beta_2 = 0.999$, and zero weight decay. When the selected text support changes, the text optimizer is re-instantiated over the newly selected variables, whereas the fixed image support preserves its optimizer state. This lifecycle is induced by Reallocate's changing-support implementation rather than by an independently tuned optimizer choice.

Unless otherwise specified, generation uses multinomial image-token decoding with temperature 1.0, classifier-free guidance scale 5.0, and 576 image tokens, without top-p, top-k, or beam-search truncation. Experiments use bfloat16 precision, batch size 1, and run-level random seed 41.

B.2 BENCHMARK PROTOCOLS

We follow the official prompt sets, evaluation pipelines, and aggregate metrics of GenEval (Ghosh et al., 2023), T2I-CompBench (Huang et al., 2023), and WISE (Niu et al., 2026). Controlled TRILO–MILR comparisons use the same backbone, prompts and ordering, run-level seed, decoding configuration, optimization reward, task-specific strict stopping criterion and threshold, refinement

ratios, maximum update budget, and final evaluation protocol. After budget exhaustion, only TRILO applies *Retain* to its stored candidates.

Sharing a run-level seed and stochastic decoding configuration does not imply bitwise-identical per-prompt initial generations once preceding optimization trajectories have diverged, because autoregressive decoding consumes the random-number stream sequentially. Unless otherwise specified, final results use the official benchmark evaluator; the online evaluator is specified separately below.

B.3 Reward and Stopping Configuration

Following MILR, the primary controlled experiments use the corresponding benchmark evaluator as the online optimization reward, which we call *benchmark-aligned feedback*. Continuous scores are shifted by -1, preserving their ordering and placing high-quality generations near reward 0. Because this shift also changes the implemented fixed-greedy update signal, it is not optimizer-invariant; controlled MILR and TRILO runs use the same convention. GenEval equivalently assigns reward 0 to a correct generation and -1 otherwise.

At step k, decoding the current latent state produces V_k with reward R_k. Optimization terminates when

$$R_k > \tau,$$

and otherwise performs another update if $k < K$. Thus, at most $K + 1$ images are evaluated, beginning with V_0; when the full update budget is exhausted, the evaluated sequence extends through V_K. If the strict condition remains unmet after V_K, then

$$\max_{0 \le k \le K} R_k \le \tau,$$

and *Retain* returns the highest-reward evaluated candidate, as defined in Section 3.3. When multiple evaluated states attain the same maximum recorded reward, the implementation retains the earliest such state. Thresholds therefore lie in the shifted optimization-reward space, not the benchmark-score space reported in the main tables.

Table 3: Optimization rewards and stopping thresholds in the primary controlled experiments. Continuous scores are shifted by -1 during online optimization.

Task	Optimization Reward	τ
GenEval	$\{0, -1\}$ correctness reward	-0.10
T2I Color / Shape / Texture	$S_{\text{BLIP}} - 1$	-0.20
T2I Spatial	$S_{\text{spatial}} - 1$	-0.10
T2I Non-Spatial	$S_{\text{CLIP}} - 1$	-0.65
T2I Complex	$S_{\text{3in1}} - 1$	-0.10
WISE	$S_{\text{WISE}} - 1$	-0.50

For example, $\tau = -0.20$ in the BLIP-VQA-based T2I categories corresponds to an unshifted score above 0.80, whereas $\tau = -0.65$ for T2I Non-Spatial corresponds to a CLIPScore above 0.35. For WISE, GPT-4o is part of the benchmark-defined protocol rather than an additional external oracle. Alternative online rewards are reported in Appendix C.7; their final generations are still evaluated by the corresponding official benchmark evaluator.

B.4 Baseline and Reproduction Details

Unless stated otherwise, external baselines are taken from their original publications, and each table or caption identifies whether a result is published or reproduced.

MILR. The main benchmark tables use the published aggregate MILR results. Experiments requiring prompt-level outputs or tightly matched conditions use our reproduction from the released implementation, including paired statistics, generation-seed robustness, controlled efficiency comparisons, and alternative WISE evaluation. For T2I-CompBench++, we combine published MILR values on categories shared with T2I-CompBench with reproduced values on the newly introduced categories; this is a hybrid reference rather than a matched full-task rerun.

Table 4: Detailed results on GenEval. Best and second-best values among the methods reported in the table are shown in **blue bold** and cyan underline, respectively.

Method	Single Obj. ↑	Two Obj. ↑	Counting ↑	Colors ↑	Position ↑	Attr. Binding ↑	Overall ↑
Non-reasoning Models							
BAGEL (Deng et al., 2025)	0.99	0.94	0.81	0.88	0.64	0.63	0.82
Qwen-Image (Wu et al., 2025)	0.99	0.92	0.89	0.88	0.76	0.77	0.87
TUNA (Liu et al., 2026)	**1.00**	0.97	0.81	0.91	0.88	0.83	0.90
SenseNova-U1 (Diao et al., 2026)	**1.00**	0.96	0.92	0.92	0.91	0.76	0.91
Training-based Reasoning Models							
Flow-GRPO (Liu et al., 2025)	**1.00**	0.99	0.95	0.92	0.99	0.86	0.95
CoR-Painter (Yan et al., 2026)	**1.00**	0.96	0.63	0.93	0.90	0.77	0.87
IV-CoT (Li et al., 2026c)	**1.00**	0.96	0.78	0.92	0.86	0.79	0.88
Qwen-Image (+ELDiff) (Pan et al., 2026)	**1.00**	0.96	0.90	0.92	0.80	0.79	0.90
Qwen-Image (+PE) (Wang et al., 2026b)	**1.00**	0.98	0.93	0.78	0.87	0.67	0.86
Janus-Pro-7B (+LatentMorph) (Chen et al., 2026)	**1.00**	0.97	–	–	0.98	0.92	0.96
Test-time Reasoning Models							
ReflectionFlow (Zhuo et al., 2025)	**1.00**	0.98	0.90	0.96	0.93	0.72	0.91
Janus-Pro-1B (+MILR) (Mi et al., 2026)	**1.00**	0.91	0.78	0.92	0.86	0.86	0.89
Janus-Pro-7B (+PARM) (Guo et al., 2025)	**1.00**	0.95	0.80	0.93	0.91	0.85	0.91
Janus-Pro-7B (+MILR) (Mi et al., 2026)	**1.00**	0.96	0.90	0.98	0.98	0.91	0.95
Janus-Pro-7B (+TwiG-ZS) (Guo et al., 2026)	–	0.94	–	–	0.84	0.67	0.86
RAISE (Jiang et al., 2026)	**1.00**	**1.00**	0.95	0.98	0.83	0.87	0.94
Qwen-Image (+TexTailor) (Li et al., 2026b)	0.99	0.97	0.86	0.90	0.75	0.80	0.88
Janus-Pro-1B (Chen et al., 2025)	0.98	0.82	0.51	0.89	0.65	0.56	0.73
Janus-Pro-1B+TRILO	**1.00**	0.98	0.94	**0.99**	0.93	0.95	0.96
Janus-Pro-7B (Chen et al., 2025)	0.98	0.85	0.56	0.89	0.77	0.64	0.78
Janus-Pro-7B+TRILO	**1.00**	0.99	**1.00**	**0.99**	**1.00**	**0.98**	**0.99**

Matched MILR reruns follow the controlled protocol in Appendix B.2, with learning rate $\eta = 0.03$, $\lambda_t = 0.2$, $\lambda_v = 0.02$, and $K = 20$, matching the corresponding TRILO runs. Reproduced results are identified in their tables and captions.

Optimizer-state handling. According to the released MILR implementation, MILR's fixed text support preserves its Adam state, whereas TRILO re-instantiates the text optimizer when Reallocate changes the selected support. Both methods use fixed image support with persistent Adam state. Therefore, fixed- and moving-support comparisons differ in the text optimizer-state lifecycle induced by their support policies, rather than by an independently tuned optimizer choice.

Sensitivity to optimizer-state persistence. To assess whether optimizer-state persistence explains the observed differences, we evaluate a TRILO variant that preserves per-token Adam moments across support changes while keeping Reallocate and all other settings fixed. On the matched 300-prompt T2I-CompBench Color subset, it obtains 0.8892, compared with 0.8883 for the default lifecycle ($\Delta = +0.0009$, 95% paired-bootstrap CI $[-0.0054, +0.0087]$). This near-identical result suggests that Adam-moment resets do not primarily explain the improvement on this subset; we treat this analysis as an implementation sanity check rather than a global robustness claim.

C More Experimental Results

C.1 Detailed Benchmark Results

Tables 4–6 report complete category-level results on GenEval, T2I-CompBench, and WISE.

GenEval. TRILO improves Janus-Pro-7B from 0.78 to 0.99 overall, with notable improvements in Counting ($0.56 \rightarrow 1.00$), Position ($0.77 \rightarrow 1.00$), and Attribute Binding ($0.64 \rightarrow 0.98$). Janus-Pro-1B similarly improves from 0.73 to 0.96, indicating that the improvement is not limited to the larger backbone.

T2I-CompBench. Janus-Pro-7B+TRILO reaches 0.6207 overall, compared with 0.3921 for the frozen backbone and 0.5325 for the literature-reported MILR reference. It achieves strong performance across compositional categories, including Color, Shape, and Texture.

WISE. TRILO improves Janus-Pro-7B from 0.35 to 0.71 overall, compared with 0.63 for the literature-reported MILR reference, and improves all six categories over the frozen backbone. Janus-Pro-1B similarly improves from 0.26 to 0.50.

Table 5: Detailed results on T2I-CompBench. Best and second-best values among the methods reported in the table are shown in **blue bold** and cyan underline, respectively.

Method	Attribute Binding			Object Relationship		Complex↑	Overall↑
	Color↑	Shape↑	Texture↑	Spatial↑	Non-Spatial↑		
Non-reasoning Models							
FLUX.1 (Labs, 2024)	0.7407	0.5718	0.6922	0.2863	0.3127	0.3703	0.4957
BAGEL (Deng et al., 2025)	0.8081	0.5794	0.7154	0.3403	0.3111	0.3778	0.5220
Qwen-Image (Wu et al., 2025)	0.8280	0.5986	0.7451	0.4430	0.3171	0.3838	0.5526
Training-based Reasoning Models							
Flow-GRPO (Liu et al., 2025)	0.8379	0.6130	0.7236	0.5447	0.3195	–	–
CoR-Painter (Yan et al., 2026)	0.8447	0.6194	0.7693	**0.6384**	0.3144	0.4118	0.5997
Qwen-Image (+ELDiff) (Pan et al., 2026)	0.8525	0.6167	0.7617	0.4855	0.3153	0.4024	0.5724
Qwen-Image (+PE) (Wang et al., 2026b)	0.8442	0.6212	0.7296	0.4472	–	0.3864	–
Janus-Pro-7B (+TwiG-RL) (Guo et al., 2026)	0.8249	0.6128	0.7319	0.3406	0.3199	0.4031	0.5389
Janus-Pro-7B (+OSPO) (Oh et al., 2026)	0.8567	0.6386	0.7727	0.3562	**0.5908**	0.4147	0.6049
Janus-Pro-7B (+LatentMorph) (Chen et al., 2026)	0.8404	0.6946	0.7989	0.5093	0.3956	**0.6360**	**0.6458**
Test-time Reasoning Models							
Janus-Pro-1B (+MILR) (Mi et al., 2026)	0.6066	0.2796	0.4177	0.2796	0.2622	0.2613	0.3512
Janus-Pro-7B (+MILR) (Mi et al., 2026)	0.8508	0.5117	0.6949	0.4613	0.3078	0.3684	0.5325
Janus-Pro-7B (+TwiG-ZS) (Guo et al., 2026)	0.7311	0.4155	0.6477	0.2198	0.3090	0.3665	0.4483
Qwen-Image (+TexTailor) (Li et al., 2026b)	0.8677	0.6348	0.7796	0.4560	0.3123	0.4104	0.5768
Janus-Pro-1B (Chen et al., 2025)	0.3411	0.2261	0.2696	0.0968	0.2808	0.2721	0.2478
Janus-Pro-1B+TRILO	0.8054	0.5438	0.7019	0.3864	0.3109	0.4089	0.5262
Janus-Pro-7B (Chen et al., 2025)	0.6359	0.3528	0.4936	0.2061	0.3085	0.3559	0.3921
Janus-Pro-7B+TRILO	**0.8883**	**0.7127**	**0.8211**	0.5085	0.3316	0.4619	0.6207

Table 6: Detailed results on WISE. Best and second-best values among the methods reported in the table are shown in **blue bold** and cyan underline, respectively.

Method	Cultural↑	Spatio-Temporal		Natural Science			Overall↑
		Time↑	Space↑	Biology↑	Physics↑	Chemistry↑	
Non-reasoning Models							
BAGEL (Deng et al., 2025)	0.44	0.55	0.68	0.44	0.60	0.39	0.52
Qwen-Image (Wu et al., 2025)	0.62	0.63	0.77	0.57	0.75	0.40	0.62
MindOmni (Xiao et al., 2025)	0.60	0.62	0.64	0.56	0.68	0.48	0.60
HunyuanImage-3.0 (Team, 2026)	0.58	0.57	0.70	0.56	0.63	0.31	0.57
Mural (Jain et al., 2026)	–	–	–	–	–	–	0.51
Training-based Reasoning Models							
T2I-R1 (Jiang et al., 2025)	0.56	0.55	0.63	0.54	0.55	0.30	0.54
BAGEL (w/ CoT) (Deng et al., 2025)	**0.76**	**0.69**	0.75	0.65	0.75	0.58	0.70
CoR-Painter (Yan et al., 2026)	0.61	0.56	0.66	0.55	0.64	0.33	0.58
NextFlow-RL (Zhang et al., 2026a)	0.63	0.63	0.77	0.58	0.67	0.39	0.62
Test-time Reasoning Models							
Mural (+CoT) (Jain et al., 2026)	–	–	–	–	–	–	0.66
Janus-Pro-1B (+MILR) (Mi et al., 2026)	–	–	–	–	–	–	0.40
Janus-Pro-7B (+MILR) (Mi et al., 2026)	0.64	0.65	0.72	0.66	0.71	0.37	0.63
Janus-Pro-7B (+TwiG-ZS) (Guo et al., 2026)	–	–	–	–	–	–	0.64
Janus-Pro-1B (Chen et al., 2025)	0.20	0.28	0.45	0.24	0.32	0.16	0.26
Janus-Pro-1B+TRILO	0.48	0.54	0.62	0.50	0.52	0.38	0.50
Janus-Pro-7B (Chen et al., 2025)	0.30	0.37	0.49	0.36	0.42	0.26	0.35
Janus-Pro-7B+TRILO	0.71	**0.69**	**0.78**	**0.70**	**0.78**	**0.59**	**0.71**

C.2 Results on T2I-CompBench++

T2I-CompBench++ (Huang et al., 2025) provides a more fine-grained evaluation of compositional generation.

Table 7 shows that Janus-Pro-7B+TRILO reaches 0.6390 overall, compared with 0.3869 for the frozen backbone and 0.5122 for the hybrid MILR reference described in Appendix B.4. This MILR reference combines published values on overlapping categories with reproduced values on newly introduced categories and is not a matched full-task rerun. The improvements span attribute binding, spatial relations, numeracy, and complex composition.

Table 7: Detailed results on T2I-CompBench++ (Huang et al., 2025). Best and second-best values among the methods reported in the table are shown in **blue bold** and cyan underline, respectively. Entries marked with * are reproduced by us under the same evaluation settings as TRILO. For published baselines, overall scores marked with * are recomputed from the complete set of category scores, including reproduced categories.

Method	Attribute Binding			Object Relationship			Numeracy↑	Complex↑	Overall↑
	Color↑	Shape↑	Texture↑	2D-Spatial↑	3D-Spatial↑	Non-Spatial↑			
Non-reasoning Models									
FLUX.1 (Labs, 2024)	0.7407	0.5718	0.6922	0.2863	0.3866	0.3127	0.6185	0.3703	0.4974
BAGEL (Deng et al., 2025)	0.8081	0.5794	0.7154	0.3403	0.3778	0.3111	0.6552	0.3778	0.5206
Qwen-Image (Wu et al., 2025)	0.8280	0.5986	0.7451	0.4430	0.4536	0.3171	0.7582	0.3838	0.5659
Training-based Reasoning Models									
Flow-GRPO (Liu et al., 2025)	0.8379	0.6130	0.7236	**0.5447**	0.4471	0.3195	0.6752	–	–
Qwen-Image (+ELDiff) (Pan et al., 2026)	0.8525	0.6167	0.7617	0.4855	0.4761	0.3153	**0.7812**	0.4024	0.5864
Qwen-Image (+PE) (Wang et al., 2026b)	0.8442	0.6212	0.7296	0.4472	0.3943	–	0.7347	0.3864	–
Janus-Pro-7B (+TwiG-RL) (Guo et al., 2026)	0.8249	0.6128	0.7319	0.3406	0.3887	0.3199	0.6193	0.4031	0.5302
Janus-Pro-7B (+OSPO) (Oh et al., 2026)	0.8567	0.6386	0.7727	0.3562	0.3926	**0.5908**	0.3161	0.4147	0.5423
Janus-Pro-7B (+LatentMorph) (Chen et al., 2026)	0.8404	0.6946	0.7989	0.5093	0.4700	0.3956	0.6233	**0.6360**	0.6210
Test-time Reasoning Models									
Janus-Pro-1B (+MILR) (Mi et al., 2026)	0.6066	0.2796	0.4177	0.2796	0.1866*	0.2622	0.2376*	0.2613	0.3164*
Janus-Pro-7B (+MILR) (Mi et al., 2026)	0.8508	0.5117	0.6949	0.4613	0.3410*	0.3078	0.5619*	0.3684	0.5122*
Janus-Pro-7B (+TwiG-ZS) (Guo et al., 2026)	0.7311	0.4155	0.6477	0.2198	0.3368	0.3090	0.5001	0.3665	0.4408
Qwen-Image (+TexTailor) (Li et al., 2026b)	0.8677	0.6348	0.7796	0.4560	0.4202	0.3123	0.7359	0.4104	0.5771
Janus-Pro-1B (Chen et al., 2025)	0.3411	0.2261	0.2696	0.0968	0.1977	0.2808	0.2146	0.2721	0.2374
Janus-Pro-1B+TRILO	0.8054	0.5438	0.7019	0.3864	0.4537	0.3109	0.5087	0.4089	0.5150
Janus-Pro-7B (Chen et al., 2025)	0.6359	0.3528	0.4936	0.2061	0.3294	0.3085	0.4132	0.3559	0.3869
Janus-Pro-7B+TRILO	**0.8883**	**0.7127**	**0.8211**	0.5085	**0.6090**	0.3316	0.7786	0.4619	**0.6390**

C.3 Component and Design Ablations

We provide cross-scale component results and controlled analyses of the support-selection and signal-scaling choices underlying *Reallocate* and *Rescale*.

C.3.1 Cross-Scale Component Contributions

Table 8 extends the component analysis to both model scales. Published Vanilla and MILR results provide literature references, while cross-scale comparisons focus on partial TRILO configurations evaluated under the common TRILO protocol.

Table 8: Cross-scale component analysis on Janus-Pro-1B and Janus-Pro-7B. Published Vanilla and MILR results are included as literature references. The partial TRILO configurations are compared within each backbone to examine the complementary effects of Reallocate, Rescale, and Retain across model scales.

Backbone	Variant	GenEval ↑	T2I-Comp. ↑	WISE ↑
Janus-Pro-1B	Vanilla (Chen et al., 2025)	0.73	0.2478	0.26
	MILR (Mi et al., 2026)	0.89	0.3512	0.40
	w/ Reallocate	0.9397	0.3910	0.4120
	w/ Reallocate + Rescale	**0.9645**	0.4592	0.4420
	w/ Reallocate + Retain	0.9577	0.4392	0.4603
	TRILO	**0.9645**	**0.5262**	**0.5023**
Janus-Pro-7B	Vanilla (Chen et al., 2025)	0.78	0.3921	0.35
	MILR (Mi et al., 2026)	0.95	0.5325	0.63
	w/ Reallocate	0.9857	0.5464	0.6430
	w/ Reallocate + Rescale	**0.9932**	0.5764	0.6730
	w/ Reallocate + Retain	0.9857	0.5847	0.6680
	TRILO	**0.9932**	**0.6207**	**0.7105**

On Janus-Pro-7B, Table 2(a) compares component configurations under the default optimizer-state lifecycle with a matched $K = 20$ MILR baseline. Because these are independently generated runs, their differences should not be interpreted as same-trajectory causal effects. In particular, the binary GenEval reward and strict threshold imply $R_0 = \cdots = R_K = -1$ on every budget-exhausted trajectory; *Retain* therefore cannot obtain a strict recorded-reward improvement in this setting. GenEval differences between independent runs are not evidence of a Retain contribution.



Figure 7: Qualitative component ablation on Janus-Pro-7B. We compare full TRILO with the vanilla backbone and partial *Reallocate*, *Reallocate* + *Rescale*, and *Reallocate* + *Retain* configurations across attribute binding, numeracy, spatial relations, commonsense, and knowledge-intensive generation.

On graded-reward T2I-CompBench and WISE, configurations that include *Rescale* or *Retain* generally achieve higher aggregate scores, and the complete configuration is best or tied-best. Across both model scales, the partial configurations support complementary roles for *Reallocate* and *Rescale*, and for *Retain* when recorded rewards distinguish earlier and terminal states. Figure 7 provides qualitative examples.

C.3.2 Reallocate: Support-Selection Strategy

Table 2(b) compares support-selection policies while fixing the backbone, optimization budget, hyperparameters, *Rescale*, *Retain*, reward, and stopping rule. Optimizer-state handling follows each policy: fixed support preserves text Adam moments, whereas changing support re-instantiates the optimizer (Appendix B.1). Therefore, this comparison evaluates the implemented support policies rather than isolating only the support location with identical optimizer states.

Fixed Prefix disables *Reallocate* while retaining the other two controls. *Random Window* samples a same-sized contiguous support uniformly. *Avoid-Low + Random* removes strongly low-entropy regions before random selection. *Top-20% Entropy + Random* samples uniformly among candidate windows ranked in the top 20% by mean entropy. *Uncertainty-Guided* is the proposed rule in Eq. 5, selecting the maximum-mean-entropy span.

Random movement provides mixed results relative to a fixed prefix. Avoid-Low + Random improves over Random Window across all three benchmarks; restricting candidates to high-entropy regions further improves T2I-CompBench and WISE, with a small GenEval decrease. Maximum-entropy selection achieves the strongest overall performance among the tested strategies.

These results suggest that entropy-based allocation provides useful guidance beyond random movement and simple saturation avoidance. However, this does not imply that entropy directly ranks realized post-update gains. Appendix C.4.1 characterizes entropy as a local saturation indicator, while Appendix C.4.2 shows that it does not recover the retrospective reward-optimal support. We therefore view maximum-entropy Reallocate as an effective online allocation policy, rather than a theoretically optimal support-selection rule.

C.3.3 Rescale: Signal-Scaling Strategy

Table 2(b) studies whether *Rescale* benefits from heterogeneous magnitudes alone or from assigning them according to state-dependent statistics. All variants use the same uncertainty-guided support.

Uniform sets $\alpha_i = 1$; *Entropy-Proportional* reverses the proposed ordering; *Shuffled Inverse-Entropy* permutes the same inverse-entropy multipliers across positions; and *Inverse-Entropy* uses the proposed clipped and mean-normalized scaling rule. All other settings are fixed.

Inverse-entropy scaling performs best across all three benchmarks. Uniform scaling is weaker, while reversing or shuffling the assignment degrades T2I-CompBench and WISE. These results indicate that assigning heterogeneous multipliers according to the selected state statistics is more effective than using heterogeneous magnitudes alone.

Table 2(a) also shows higher aggregate scores for the Rescale-only configuration than for matched MILR without Reallocate. In Table 2(b), *Uniform* corresponds to *Reallocate + Retain* in panel (a). The inverse-entropy configuration scores higher than this uniform counterpart on all three benchmarks, supporting complementary state-dependent control.

These interventions evaluate the implemented policy. In the idealized unbiased and independently scaled case, larger gradient noise-to-signal ratios motivate more conservative scales; for the implemented biased signal, alignment with the true optimization direction also matters. Therefore, the theory serves as a design motivation rather than evidence that entropy estimates the exact ratio, determines globally optimal scales, or yields an optimal Adam update.

C.4 Mechanistic Analysis of Adaptive Latent Optimization

We analyze the three adaptive decisions underlying TRILO: local saturation and changing refinement opportunity for *Reallocate*, state-dependent signal scaling for *Rescale*, and trajectory non-monotonicity with independent post-hoc evaluation for *Retain*.

C.4.1 Predictive Entropy and Local Optimization Responsiveness

Section 4 shows that, under bounded reward and local logit sensitivity, the fixed-context greedy pre-optimizer score vanishes as predictive entropy approaches zero. We first examine whether this local quantity is reflected in realized TRILO trajectories.

Greedy-score responsiveness. Before masking, Rescale, and latent updates, we probe every text position. For

$$y_i^k \in \arg\max_y p(y \mid z_i^{(t),k}, \mathcal{C}_i^k),$$

we measure the reward-weighted greedy-score norm

$$G_i^k = |R_k| \left\| \nabla_{z_i^{(t),k}} \log p\left(y_i^k \mid z_i^{(t),k}, \mathcal{C}_i^k\right) \right\|_2 . \tag{25}$$

Here $\mathcal{C}_i^k$ denotes the remaining context, as in Eq. 2. The greedy token is recomputed at each checkpoint, making this the same local pre-optimizer quantity considered in Proposition A.1. The probe is discarded and does not alter the optimization trajectory.

To remove sample- and step-level scale differences, we normalize

$$\widetilde{G}_i^k = \frac{G_i^k}{\frac{1}{M}\sum_{j=1}^{M} G_j^k + \epsilon}, \tag{26}$$

and partition text positions into entropy quintiles independently at each optimization step. Steps with all-zero reward-weighted responses are excluded.

Table 9 shows that the lowest-entropy quintile exhibits only 2.5%, 3.9%, and 5.3% of the normalized response of the remaining positions on T2I-CompBench, WISE, and GenEval, respectively. This observation supports predictive entropy as a one-sided indicator of local saturation. It does not imply that higher entropy always corresponds to larger realized gains or better refinement locations.

Sampled-signal variability. We next analyze a complementary diagnostic motivated by the local signal-scaling discussion for *Rescale*. At 80 saved checkpoints from 34 unique sample–split units, we keep the text latent and recorded reward fixed and draw $S = 8$ sampled tokens for each valid position:

$$y_i^{(s)} \sim p(\cdot \mid z_i^{(t)}, \mathcal{C}_i), \qquad s = 1, \ldots, S.$$

Table 9: Predictive entropy and local optimization responsiveness. Text latents are partitioned into entropy quintiles independently at each optimization step. We report the normalized response $\widetilde{G}$ for the lowest-entropy quintile Q_1 and the remaining positions $Q_{2:5}$, together with $\rho_G = \mu_{Q_1}/\mu_{Q_{2:5}}$. Confidence intervals are obtained by sample-level bootstrap.

Benchmark	G-valid	μ_{Q_1}	$\mu_{Q_{2:5}}$	ρ_G (95% CI)
T2I-CompBench	153 / 200	0.031	1.253	0.025 [0.019, 0.032]
WISE	166 / 200	0.048	1.245	0.039 [0.032, 0.048]
GenEval	20 / 100	0.065	1.245	0.053 [0.031, 0.077]

G-valid denotes samples containing at least one optimization step with a nonzero reward-weighted response. For the sampled GenEval subset, the initial accuracy is 0.80, leaving few unsuccessful samples to contribute to this diagnostic.

For each sampled token, we compute

$$\xi_i^{(s)} = R\nabla_{z_i^{(t)}} \log p\Big(y_i^{(s)} \mid z_i^{(t)}, \mathcal{C}_i\Big). \tag{27}$$

Image tokens and rewards are not resampled. We summarize sampled signal-energy variability by

$$\widehat{\nu}_i = \frac{\mathrm{Var}_s\left[\|\xi_i^{(s)}\|_2^2\right]}{\mathbb{E}_s\left[\|\xi_i^{(s)}\|_2^2\right] + \epsilon}. \tag{28}$$

Across the $3,989$ positions with nonzero $\widehat{\nu}_i$, predictive entropy has a positive Spearman association with sampled-signal variability:

$$\rho_{\mathrm{Sp}} = 0.418,$$

with 95% CI $[0.329, 0.483]$.

The confidence interval uses $B = 2000$ cluster-bootstrap resamples over the 34 $(\mathrm{split}, \mathrm{sample_id})$ units. The complete diagnostic contains $5,217$ evaluated positions; only positions with nonzero variability statistics are included in the correlation analysis.

This statistic should not be interpreted as a direct empirical estimate of the theoretical quantity

$$q_i = \frac{\sigma_i^2}{\|g_i\|_2^2}.$$

The true objective gradient is unavailable, and sampled score-function signals need not provide an unbiased estimate under the implemented procedure. Moreover, these sampled-token probes are not used by TRILO: the actual optimization signal remains the greedy-token signal in Eq. 2. We therefore use this analysis only to characterize state-dependent signal variability, while evaluating the practical scaling policy through controlled Rescale ablations.

C.4.2 Counterfactual Analysis of Refinement Opportunity

We next examine whether equal-budget text refinements applied to different supports produce different realized outcomes. We treat the original TRILO trajectory as Phase 1 and perform a state-conditioned counterfactual replay as Phase 2.

From full Phase-1 trajectories, we retain eligible checkpoints $k \in \{0, 5, 10, 15\}$ for which stopping is unmet and another update would normally occur. We enumerate all contiguous text supports of size L_t.

Each candidate starts from the same saved latent state, uses the same precomputed reward-guided text signals, and applies the same one-step argmax image-prefix update, which is independent of the candidate text support. We disable Rescale by setting $\alpha_i = 1$ and apply a fresh Adam text update with learning rate 0.03. Candidates are decoded using the same configured decoding seed.

The intervention therefore varies the active text support under a standardized one-step update while holding the recorded configuration fixed, rather than attempting to exactly replay the next TRILO step.

For each support $\mathcal{U}$, realized utility is measured relative to the corresponding Phase-2 no-text-update replay:

$$G_k(\mathcal{U}) = R_{\text{new},k}(\mathcal{U}) - R_{\text{noop},k}. \tag{29}$$

Stochastic replay. Janus-Pro samples image tokens multinomially at temperature 1.0, so re-decoding a saved latent state may not exactly reproduce its stored Phase-1 image or reward. Phase 2 reloads the saved text- and image-latent payloads; remaining differences can arise from stochastic decoding and, on WISE, evaluator-side variability.

Across the replay census, the median absolute stored–replayed reward deviation is 0.029 on T2I-CompBench and 0.000 on WISE. We therefore define all counterfactual gains relative to the matched Phase-2 no-text-update replay rather than the stored Phase-1 reward.

Overall, we evaluate 14,584 candidate supports from 273 checkpoints: 181 checkpoints from 60 T2I-CompBench samples and 92 from 35 WISE samples. Statistics are computed at checkpoint level, aggregated within sample, and bootstrapped over samples with 2,000 replicates.

Table 10: Counterfactual one-step refinement opportunity. For each eligible checkpoint, we exhaustively vary only the contiguous text-refinement support under an otherwise matched one-step intervention. $P(G > 0)$ is the fraction of candidate supports yielding positive realized gain, $G_k^\star = \max_{\mathcal{U}} G_k(\mathcal{U})$ is the best realized gain, and $G_k^\star - \text{Med}_{\mathcal{U}}[G_k(\mathcal{U})]$ measures its advantage over the median candidate support. $F(\gamma)$ is the fraction of candidates satisfying $G_k(\mathcal{U}) \geq \gamma G_k^\star$ and is evaluated only at checkpoints with $G_k^\star > 0$. Exact turnover measures the change between exact retrospective oracle sets at adjacent eligible checkpoints of the same sample. Values are sample-level means with 95% bootstrap confidence intervals.

Metric	T2I-CompBench	WISE
$P(G > 0)$	0.219 [0.165, 0.284]	0.193 [0.119, 0.280]
$G^\star$	0.156 [0.102, 0.224]	0.320 [0.243, 0.396]
$G^\star - \text{Med}(G)$	0.297 [0.225, 0.369]	0.428 [0.357, 0.498]
$F(0.8)$	0.046 [0.037, 0.055]	0.061 [0.047, 0.077]
$F(0.9)$	0.035 [0.028, 0.042]	0.051 [0.040, 0.064]
Exact Turnover	0.680 [0.587, 0.772]	0.252 [0.150, 0.359]

$F(0.8)$ and $F(0.9)$ are computed only on checkpoints with positive oracle gain (52 T2I-CompBench and 32 WISE checkpoints). With $\mathcal{M}_k = \{\mathcal{U} : G_k(\mathcal{U}) \geq G_k^\star - 10^{-9}\}$, exact turnover between adjacent eligible checkpoints $k < k'$ from the same sample is $1 - \max_{\mathcal{U}\in\mathcal{M}_k,\, \mathcal{V}\in\mathcal{M}_{k'}} \text{IoU}(\mathcal{U}, \mathcal{V})$.

Refinement opportunity is selective. Only 21.9% of T2I-CompBench and 19.3% of WISE candidate supports yield positive gain. However, the best support produces mean gains of 0.156 and 0.320, exceeding the median candidate by 0.297 and 0.428, respectively. Among checkpoints with positive oracle gain, only 3.5% and 5.1% of candidates achieve at least 90% of the best gain.

These results show that equal-budget supports can have substantially different refinement outcomes, motivating adaptive support allocation rather than a fixed support policy.

For checkpoint k, define

$$G_k^\star = \max_{\mathcal{U}} G_k(\mathcal{U}), \qquad \mathcal{M}_k = \left\{\mathcal{U} : G_k(\mathcal{U}) \geq G_k^\star - 10^{-9}\right\}, \tag{30}$$

where $\mathcal{M}_k$ denotes the retrospective oracle set after evaluating all candidate supports.

Support dynamics are benchmark dependent. For adjacent eligible checkpoints from the same sample, we measure oracle support turnover:

$$1 - \max_{\mathcal{U}\in\mathcal{M}_k, \mathcal{V}\in\mathcal{M}_{k'}} \text{IoU}(\mathcal{U}, \mathcal{V}). \tag{31}$$

Exact turnover averages 0.680 on T2I-CompBench and 0.252 on WISE.

Because WISE contains more discrete same-image scoring variability, we also evaluate a near-optimal set

$$\mathcal{M}_k^{(\delta)} = \left\{\mathcal{U} : G_k^\star - G_k(\mathcal{U}) \leq \delta + 10^{-9}\right\}, \qquad \delta = 0.15. \tag{32}$$

Table 11: Sensitivity of WISE support dynamics to evaluator variability. Turnover is computed between near-optimal oracle sets at adjacent eligible checkpoints of the same sample. Within-δ Oracle Hit is the fraction of checkpoints for which the online TRILO support lies within δ of the best realized gain.

Metric	WISE
Turnover ($\delta = 0.15$)	0.117 [0.046, 0.204]
Within-δ Oracle Hit	0.314 [0.186, 0.443]
Mean # δ-Optimal Supports	14.74 [9.11, 21.03]

Near-optimal turnover is $1 - \max_{\mathcal{U} \in \mathcal{M}_k^{(\delta)},\, \mathcal{V} \in \mathcal{M}_{k'}^{(\delta)}} \mathrm{IoU}(\mathcal{U}, \mathcal{V})$ for adjacent eligible checkpoints $k < k'$ from the same sample.

Table 12: Online TRILO support versus the retrospective counterfactual oracle. The oracle is defined retrospectively from realized post-update rewards over all candidate supports and is unavailable to the online method. Regret is $G_k^\star - G_k(\mathcal{U}_k^{\mathrm{TRILO}})$. Top-3 Hit uses competition rank, counting only candidates whose realized gain exceeds that of the online support by more than 10^{-9}. Support IoU is the maximum IoU between the online support and any exact oracle support. Normalized Start Distance is measured relative to the maximum-IoU oracle support and normalized by the maximum possible start-position displacement $\max(1, M - L_t)$. Values are sample-level means with 95% bootstrap confidence intervals.

Metric	T2I-CompBench	WISE
Regret ↓	0.270 [0.204, 0.342]	0.413 [0.331, 0.495]
Top-3 Hit ↑	0.200 [0.125, 0.288]	0.193 [0.093, 0.300]
Support IoU ↑	0.212 [0.148, 0.282]	0.377 [0.300, 0.457]
Normalized Start Distance ↓	0.338 [0.291, 0.388]	0.217 [0.167, 0.272]

Competition rank is $1 + |\{\mathcal{U} : G_k(\mathcal{U}) > G_k(\mathcal{U}_k^{\mathrm{TRILO}}) + 10^{-9}\}|$. For $\mathcal{M}_k = \{\mathcal{U} : G_k(\mathcal{U}) \geq G_k^\star - 10^{-9}\}$, Support IoU is $\max_{\mathcal{U} \in \mathcal{M}_k} \mathrm{IoU}(\mathcal{U}_k^{\mathrm{TRILO}}, \mathcal{U})$. Normalized Start Distance uses an oracle support attaining this maximum IoU; residual IoU ties are broken by candidate order.

Under this tolerance, WISE turnover decreases from 0.252 to 0.117. Therefore, we interpret the rapid support changes observed on T2I-CompBench more strongly and treat WISE dynamics more conservatively.

Online support versus retrospective oracle. Table 12 compares the online entropy-guided support with the retrospective oracle defined from exhaustive candidate evaluation.

The nonzero regret and limited overlap indicate that Reallocate is not a close approximation to this hindsight oracle. This is expected because Reallocate selects supports before observing post-update outcomes, whereas the oracle is defined only after evaluating all candidates.

For each candidate support $\mathcal{U}$, we compute the same window-mean entropy used by Reallocate:

$$\overline{H}_k(\mathcal{U}) = \frac{1}{|\mathcal{U}|} \sum_{i \in \mathcal{U}} H_i^k. \tag{33}$$

Candidate windows are partitioned independently within each checkpoint according to $\overline{H}_k(\mathcal{U})$.

Table 13 shows similar positive-gain probabilities and positive-part gains for the lowest-entropy group and the remaining supports on both benchmarks.

Together with the local-response analysis, these results distinguish two roles of entropy: entropy provides information about local saturation, but it does not retrospectively rank realized reward gains. Maximum-entropy selection should therefore be viewed as an effective online allocation heuristic rather than a reward-optimal oracle.

C.4.3 Non-Monotonic Optimization Trajectories

To evaluate the premise behind *Retain*, we analyze trajectories that exhaust the full $K = 20$ update budget. For this diagnostic only, we exclude V_0 to isolate degradation after optimization begins and define

Table 13: Predictive entropy versus realized one-step refinement utility. For each candidate support $\mathcal{U}$, we compute its window-mean predictive entropy $\overline{H}_k(\mathcal{U}) = |\mathcal{U}|^{-1} \sum_{i \in \mathcal{U}} H_i^k$. Candidate supports are partitioned independently within each checkpoint into the lowest-entropy group Q_1 and the remaining supports $Q_{2:5}$. We report the probability of positive realized gain and the positive-part gain $G_+ = \max(G, 0)$. Values are sample-level means with 95% bootstrap confidence intervals.

Benchmark	Metric	Q_1	$Q_{2:5}$
T2I-CompBench	$P(G > 0)$	0.206 [0.145, 0.273]	0.223 [0.162, 0.285]
	$\mathbb{E}[G_+]$	0.023 [0.011, 0.038]	0.026 [0.015, 0.039]
WISE	$P(G > 0)$	0.189 [0.111, 0.281]	0.194 [0.119, 0.282]
	$\mathbb{E}[G_+]$	0.044 [0.025, 0.067]	0.046 [0.029, 0.065]

$$k^\star_{\text{opt}} \in \arg \max_{1 \le k \le K} R_k. \tag{34}$$

Ties are assigned to $k^\star_{\text{opt}} = K$. Hence $k^\star_{\text{opt}} < K$ only when an intermediate optimized state strictly outperforms the terminal state. This convention is specific to the diagnostic; practical Retain also considers V_0 and selects the earliest maximum.

Table 14: Non-monotonicity of budget-exhausted optimization trajectories. For this diagnostic, the initial state V_0 is excluded and $k^\star_{\text{opt}} \in \arg\max_{1 \le k \le K} R_k$, with ties assigned to $k^\star_{\text{opt}} = K$. *Earlier Best* is the fraction of exhausted trajectories satisfying $k^\star_{\text{opt}} < K$, i.e., an intermediate optimized state strictly outperforms the terminal state. *Avg. Opt.-Best Gap* is $\mathbb{E}[R_{k^\star_{\text{opt}}} - R_K \mid \text{exhausted}]$.

Benchmark	Exhausted	Earlier Best (%)	Avg. Best Step	Avg. Opt.-Best Gap
GenEval	4 / 553 (0.7%)	0.0	20.00	0.000
T2I-CompBench	914 / 1800 (50.8%)	**81.3**	11.67	**+0.139**
WISE	294 / 1000 (29.4%)	**67.0**	10.61	**+0.201**

Among budget-exhausted trajectories, an intermediate optimized state strictly outperforms the terminal state in 81.3% of T2I-CompBench and 67.0% of WISE cases. The mean optimized-best-minus-terminal reward gaps are $+0.139$ and $+0.201$, respectively.

Under the binary benchmark-aligned GenEval reward with $\tau = -0.10$, all four budget-exhausted trajectories satisfy $R_0 = \cdots = R_K = -1$ and therefore do not admit strict recorded-reward Retain improvements.

These results establish the empirical motivation for trajectory-aware selection: budget-exhausted graded-reward trajectories are frequently non-monotonic, and the terminal state is not always the best visited state.

C.4.4 Retain under Independent Evaluators

The guarantee in Section 3.3 and Proposition A.9 concerns only the recorded optimization reward along the same trajectory. We therefore separately test whether independent post-hoc evaluators prefer the Retain-selected image.

The practical Retain rule keeps the earliest state attaining the maximum recorded reward:

$$k^\star_R = \min \arg \max_{0 \le k \le K} R_k. \tag{35}$$

Thus, V_0 is eligible and a stored best state is replaced only by a strictly higher recorded reward.

For evaluator E, we define

$$\Delta_E = E(V_{k^\star_R}) - E(V_K). \tag{36}$$

We keep the 294 budget-exhausted WISE trajectories and their original Retain decisions fixed. On the 196 trajectories satisfying $R(V_{k^\star_R}) > R(V_K)$, retained and terminal images are each scored once with Qwen3.5-35B-A3B (Qwen Team, 2026) and UnifiedReward-Qwen-7B (Wang et al., 2025).

For Qwen3.5-35B-A3B, we follow the official WISE evaluation protocol using the image, prompt, and explanation; the evaluator returns a binary score in $\{0, 1\}$ with temperature 0 and thinking disabled. For UnifiedReward-Qwen-7B, E is the raw `Final Score` in $[1, 5]$, with absolute score differences below 10^{-4} treated as ties. Neither evaluator participates in optimization or selection.

Table 15: Retention under independent post-hoc evaluators on WISE. Results are reported on the 196 strict-improvement trajectories satisfying $R(V_{k_R^\star}) > R(V_K)$, where $k_R^\star = \min \arg\max_{0 \le k \le K} R_k$ is the actual Retain decision and therefore keeps the earliest maximum. The retained state is selected exclusively by the original WISE optimization reward; independent evaluators are used only for post-hoc comparison with the terminal state. DWR denotes the decisive win rate $N_{\text{win}}/(N_{\text{win}} + N_{\text{loss}})$ and excludes evaluator ties. Confidence intervals are obtained by trajectory-level bootstrap.

Independent Evaluator	Win ↑	Tie	Loss ↓	DWR ↑	Mean Δ_E (95% CI) ↑
Qwen3.5-35B-A3B	27.6%	69.4%	3.1%	**90.0%**	$+\mathbf{0.245}$ $[+0.173, +0.316]$
UnifiedReward-Qwen-7B	36.2%	50.0%	13.8%	72.4%	$+\mathbf{0.198}$ $[+0.116, +0.282]$

For Qwen3.5-35B-A3B, evaluator scores are binary. For UnifiedReward-Qwen-7B, E is the raw `Final Score` in $[1, 5]$, with absolute score differences below 10^{-4} treated as ties. Each retained and terminal image is evaluated once.

On the strict-improvement subset, Qwen3.5-35B-A3B prefers the retained image in 54 of 60 decisive comparisons, giving a 90.0% decisive win rate (DWR). UnifiedReward-Qwen-7B gives a 72.4% DWR. The corresponding mean retained-minus-terminal differences are $+0.245$ and $+0.198$.

Across all 294 exhausted trajectories, including online-reward ties, Qwen3.5-35B-A3B gives Win/Tie/Loss rates of $20.4\%/73.1\%/6.5\%$ and

$$\overline{\Delta}_E = \frac{60 - 19}{294} = 0.139,$$

with a 95% trajectory-level percentile-bootstrap confidence interval of $[+0.082, +0.194]$.

These external evaluations complement, rather than extend, the deterministic Retain guarantee. Retain guarantees non-decreasing recorded optimization reward, while preference under an independent evaluator remains an empirical property.

C.5 Additional Test-Time Scaling Results

Figure 8 varies the maximum optimization budget K for full and partial TRILO configurations on both model scales and all three benchmarks. Performance generally improves with increasing budget, with diminishing gains at larger K, while full TRILO remains strongest or competitive with the best partial configuration.

At $K = 16$, matching the budget associated with the literature-reported MILR results, full TRILO exceeds that reference in all six model–benchmark settings. Among the 18 corresponding partial-configuration comparisons, 17 also exceed the MILR reference; the exception is Reallocate alone on Janus-Pro-1B WISE. These results indicate that the improvement is not specific to the default $K = 20$ setting.

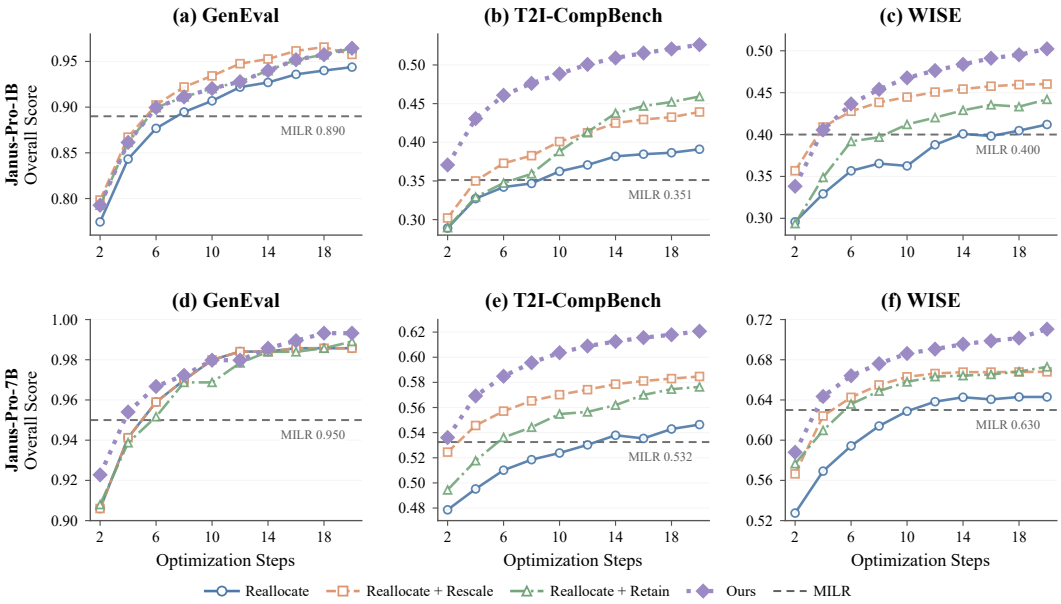

Figure 8: Test-time scaling across model scales and optimization budgets. Rows show Janus-Pro-1B and Janus-Pro-7B; columns show GenEval, T2I-CompBench, and WISE. Curves compare full TRILO with partial *Reallocate*, *Reallocate* + *Rescale*, and *Reallocate* + *Retain* configurations as K increases. Dashed lines denote the corresponding literature-reported MILR scores at $K = 16$.

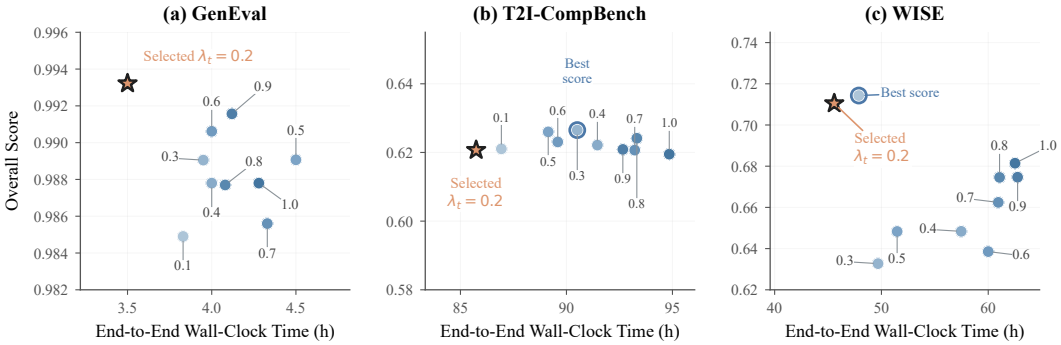

Figure 9: Performance–wall-clock trade-off across text refinement ratios λ_t. The star denotes the default $\lambda_t = 0.2$; the outlined marker denotes the highest-scoring tested setting when different.

C.6 Effect of Text Refinement Ratio

Using Janus-Pro-7B, we vary the per-step text refinement ratio $\lambda_t \in \{0.1, 0.2, \ldots, 1.0\}$ while keeping the remaining configuration fixed, and measure both generation quality and end-to-end wall-clock time.

Increasing λ_t does not monotonically improve quality and generally increases runtime. The default $\lambda_t = 0.2$ is the fastest tested setting and achieves the best GenEval score; its T2I-CompBench and WISE scores are within 0.0058 and 0.0038, respectively, of the best tested values. We therefore use $\lambda_t = 0.2$ as a common performance–efficiency operating point rather than a benchmark-specific quality optimum.

Figure 9 shows that the selected setting lies on or near a favorable empirical trade-off: updating more text positions adds computation without consistently improving generation quality.

Figure 10 similarly illustrates that refining more text latents does not uniformly improve compositional or knowledge-dependent generation.



Figure 10: Qualitative sensitivity to λ_t. We compare $\lambda_t \in \{0.1, 0.2, \ldots, 1.0\}$ under otherwise identical settings on counting, spatial, scene-attribute, and knowledge-intensive examples.

Table 16: Sensitivity to the text refinement ratio λ_t on Janus-Pro-7B. The default $\lambda_t = 0.2$ is highlighted. Best and second-best scores are shown in blue bold and cyan underline, respectively; time is measured as end-to-end wall-clock time and lower is better.

(a) GenEval

λ_t	Single Obj.	Two Obj.	Counting	Colors	Position	Attr. Binding	**Overall ↑**	**End-to-End Wall-Clock Time (h) ↓**
0.1	1.0000	1.0000	1.0000	0.9894	0.9800	0.9400	0.9849	3.83
0.2	1.0000	0.9899	1.0000	0.9894	1.0000	0.9800	**0.9932**	**3.49**
0.3	1.0000	0.9899	0.9750	0.9894	1.0000	0.9800	0.9891	3.95
0.4	1.0000	0.9899	0.9875	0.9894	0.9900	0.9700	0.9878	4.00
0.5	1.0000	1.0000	0.9750	0.9894	0.9900	0.9800	0.9891	4.50
0.6	1.0000	1.0000	0.9750	0.9787	1.0000	0.9900	0.9906	4.00
0.7	1.0000	0.9899	0.9750	0.9787	0.9900	0.9800	0.9856	4.33
0.8	1.0000	1.0000	0.9875	0.9787	0.9900	0.9700	0.9877	4.08
0.9	1.0000	1.0000	1.0000	0.9894	0.9900	0.9700	0.9916	4.12
1.0	1.0000	0.9899	0.9875	0.9894	0.9900	0.9700	0.9878	4.28

(b) T2I-CompBench

λ_t	Color	Shape	Texture	Spatial	Non-Spatial	Complex	**Overall ↑**	**End-to-End Wall-Clock Time (h)↓**
0.1	0.8801	0.7119	0.8189	0.5195	0.3299	0.4660	0.6211	86.93
0.2	0.8883	0.7127	0.8211	0.5085	0.3316	0.4619	0.6207	**85.75**
0.3	0.8887	0.7264	0.8154	0.5300	0.3316	0.4670	**0.6265**	90.51
0.4	0.8835	0.7133	0.8228	0.5205	0.3306	0.4622	0.6222	91.46
0.5	0.8887	0.7264	0.8157	0.5289	0.3300	0.4663	0.6260	89.14
0.6	0.8865	0.7340	0.8154	0.5069	0.3313	0.4643	0.6231	89.59
0.7	0.8881	0.7156	0.8151	0.5167	0.3303	0.4583	0.6207	93.22
0.8	0.8829	0.7255	0.8172	0.5295	0.3307	0.4590	0.6241	93.32
0.9	0.8831	0.7081	0.8120	0.5318	0.3317	0.4586	0.6209	92.66
1.0	0.8781	0.7190	0.8106	0.5174	0.3288	0.4631	0.6195	94.85

(c) WISE

λ_t	Cultural	Time	Space	Biology	Physics	Chemistry	**Overall ↑**	**End-to-End Wall-Clock Time (h)↓**
0.1	0.7100	0.7000	0.8000	0.6700	0.7900	0.6100	**0.7143**	47.88
0.2	0.7103	0.6940	0.7808	0.6995	0.7780	0.5885	0.7105	**45.58**
0.3	0.6813	0.4589	0.5786	0.6834	0.8067	0.5759	0.6327	49.68
0.4	0.6648	0.6695	0.7426	0.6714	0.4916	0.5558	0.6484	57.47
0.5	0.6672	0.6710	0.7475	0.6251	0.5293	0.5452	0.6483	51.47
0.6	0.6684	0.6548	0.7361	0.6368	0.4708	0.5318	0.6386	60.00
0.7	0.6538	0.6712	0.7385	0.6612	0.7384	0.5064	0.6624	60.94
0.8	0.6736	0.6765	0.7358	0.6623	0.7441	0.5365	0.6746	61.06
0.9	0.6781	0.6679	0.7576	0.6654	0.7362	0.5096	0.6747	62.75
1.0	0.6670	0.6961	0.7682	0.6758	0.7534	0.5331	0.6814	62.51

C.7 Detailed Reward-Model Results

We evaluate four practical online-reward configurations on Janus-Pro-7B: SelfReward, GPT-4o, UnifiedReward, and MixedReward, together with benchmark-aligned feedback as a controlled reference. T2I-CompBench and WISE follow the practical MILR reward protocols where applicable (Mi et al., 2026), while GenEval uses the same reward families with benchmark-specific observation and scoring.

Across configurations, the online reward supplies the scalar used to construct latent signals, determine early stopping, and select the Retain output after budget exhaustion. Final reported performance is always computed by the corresponding official benchmark evaluator.

All configurations follow the same stopping-criterion form,

$$R > \tau,$$

but τ is the reward-specific threshold configured for each run rather than a common threshold shared across reward families. Because both scalar ranges and configured thresholds differ, these experiments should be interpreted as robustness across practical online-reward configurations, not as a single-variable common-threshold critic intervention.

SelfReward. Janus-Pro scores each generated image once using deterministic decoding (`do_sample=False`, `max_new_tokens=256`). For structured include/exclude annotations, let $\mathcal{I}$ denote the required object set, and let need_j and obs_j denote the required and model-observed counts for item j. The reward is

$$R_{\mathrm{Self}} = \frac{1}{|\mathcal{I}|} \sum_{j \in \mathcal{I}} \max\left(0,\, 1 - \frac{|\mathrm{obs}_j - \mathrm{need}_j|}{\mathrm{need}_j}\right) - 1. \tag{37}$$

A color mismatch sets the corresponding item contribution to zero before the final shift, while an exclude violation forces $R_{\mathrm{Self}} = -1$. Without structured annotations, the model performs caption-level judgment and returns a binary score in $\{0, -1\}$; parsing failures are also assigned -1. Thus,

$$R_{\mathrm{Self}} \in [-1, 0].$$

GPT-4o reward. We use `gpt-4o-2024-11-20`. The reported experiments use one deterministic evaluation per image with temperature 0. Structured include/exclude cases use the same fractional scoring rule as Eq. 37; otherwise the evaluator is constrained to return a single score in $\{0, -1\}$. Failed calls receive -1, so

$$R_{\mathrm{GPT4o}} \in [-1, 0].$$

UnifiedReward. We use the local UnifiedReward-Qwen-7B model with greedy decoding and one evaluation per image. If its parsed `Final Score` is $r_{\mathrm{raw}} \in [1, 5]$, the optimization reward is

$$R_{\mathrm{Uni}} = r_{\mathrm{raw}} - 5 \in [-4, 0]. \tag{38}$$

MixedReward. MixedReward combines GroundingDINO, a GIT-based color score, and UnifiedReward. Each component is normalized to $[0, 1]$ by the implemented scoring pipeline; for UnifiedReward,

$$S_{\mathrm{Uni}} = \frac{r_{\mathrm{raw}} - 1}{4}.$$

For prompts tagged as `colors` or `color_attr`,

$$R_{\mathrm{Mix}} = 0.4 S_{\mathrm{DINO}} + 0.4 S_{\mathrm{GIT}} + 0.2 S_{\mathrm{Uni}} - 1, \tag{39}$$

whereas all other tags use

$$R_{\mathrm{Mix}} = 0.6 S_{\mathrm{DINO}} + 0.2 S_{\mathrm{GIT}} + 0.2 S_{\mathrm{Uni}} - 1. \tag{40}$$

Hence

$$R_{\mathrm{Mix}} \in [-1, 0].$$

Benchmark-aligned feedback follows the definitions in Appendix B.3.

All four practical alternative reward configurations improve over the frozen backbone on all three benchmarks. MixedReward gives the highest tested alternative score on GenEval (0.8994) and

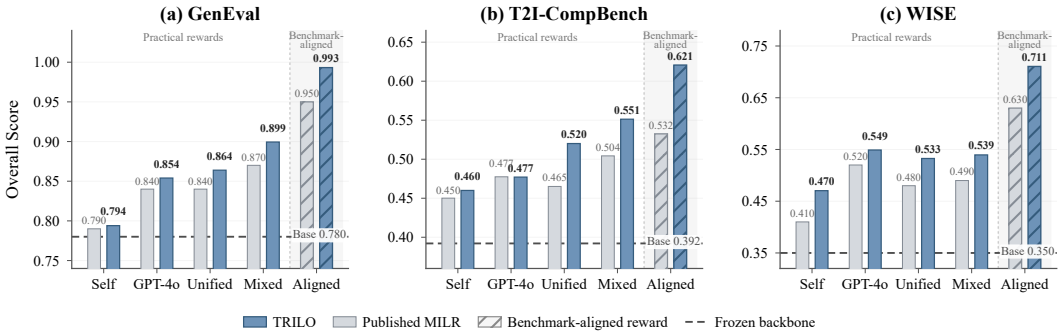

Figure 11: Performance under different online-reward configurations. We compare SelfReward, GPT-4o, UnifiedReward, MixedReward, and benchmark-aligned feedback. Each reward uses its corresponding configured stopping threshold; final scores use the official benchmark evaluators. Literature-reported MILR results are included for context.

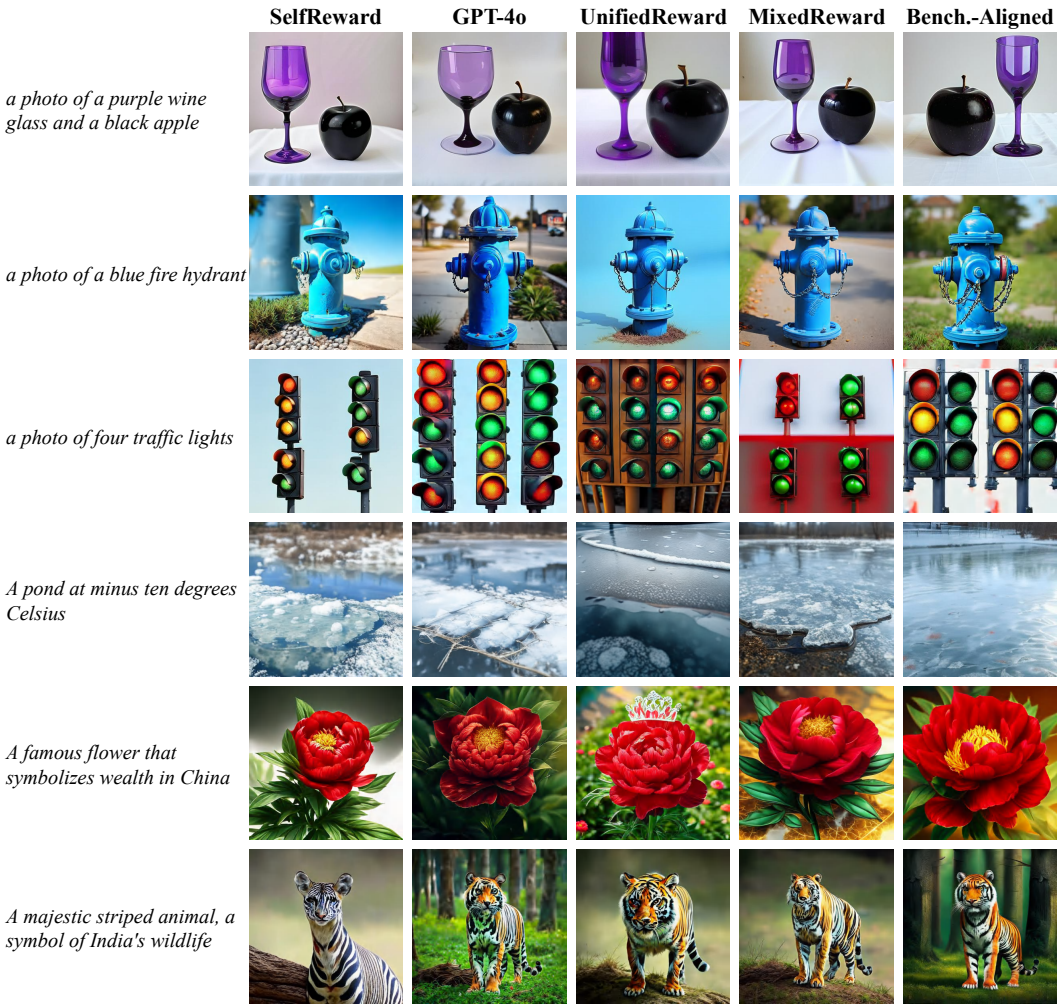

Figure 12: Qualitative comparison across online-reward configurations. We compare SelfReward, GPT-4o, UnifiedReward, MixedReward, and benchmark-aligned feedback using the same backbone and control procedure, with each reward using its corresponding configured stopping threshold.

T2I-CompBench (0.5513), while the GPT-4o configuration is highest on WISE (0.5490). The highest-scoring tested practical configuration therefore varies by benchmark.

For literature context, TRILO exceeds the corresponding MILR result in 11 of 12 benchmark–reward-family comparisons; the exception is T2I-CompBench with GPT-4o, where the difference is 0.0004 (0.4771 versus 0.4775).

Table 17: Effect of practical online-reward configurations on GenEval, T2I-CompBench, and WISE. All reported scores are computed with the corresponding official benchmark evaluator after generation. Reward labels denote the online critic used to construct latent signals, control stopping, and select Retain outputs after budget exhaustion. All configurations use the same backbone and test-time control procedure, but each reward uses its corresponding configured stopping threshold rather than a common τ. The best result among the four practical alternative rewards in each evaluation dimension is shown in bold. Vanilla denotes the frozen Janus-Pro-7B backbone. Bench.-Aligned directly uses the corresponding benchmark evaluator as the online critic and is reported as a controlled reference.

(a) GenEval

Reward Model	Single Obj.	Two Obj.	Counting	Colors	Position	Attr. Binding	Overall
Vanilla	0.9800	0.8500	0.5600	0.8900	0.7700	0.6400	0.7800
SelfReward	0.9875	0.8990	0.5250	0.8723	0.7800	0.7000	0.7940
GPT-4o	**1.0000**	0.9192	0.8125	0.8723	0.7800	0.7400	0.8540
UnifiedReward	**1.0000**	**0.9293**	0.8000	0.9043	0.8300	0.7200	0.8639
MixedReward	**1.0000**	0.9192	**0.8625**	**0.9149**	**0.9000**	**0.8000**	**0.8994**
Bench.-Aligned	1.0000	0.9899	1.0000	0.9894	1.0000	0.9800	0.9932

(b) T2I-CompBench

Reward Model	Color	Shape	Texture	Spatial	Non-Spatial	Complex	Overall
Vanilla	0.6359	0.3528	0.4936	0.2061	0.3085	0.3559	0.3921
SelfReward	0.7191	0.4631	0.6244	0.2634	0.3088	0.3811	0.4600
GPT-4o	0.7670	0.5270	0.6374	0.2382	0.3104	0.3823	0.4771
UnifiedReward	**0.8363**	0.5438	0.6542	0.3545	0.3178	0.4142	0.5201
MixedReward	0.8160	**0.5622**	**0.7408**	**0.4504**	**0.3190**	**0.4193**	**0.5513**
Bench.-Aligned	0.8883	0.7127	0.8211	0.5085	0.3316	0.4619	0.6207

(c) WISE

Reward Model	Cultural	Time	Space	Biology	Physics	Chemistry	Overall
Vanilla	0.3000	0.3700	0.4900	0.3600	0.4200	0.2600	0.3500
SelfReward	0.4376	0.4946	0.6128	0.4240	0.5495	0.3395	0.4704
GPT-4o	**0.5298**	0.5416	**0.7199**	**0.5200**	0.5870	0.4015	**0.5490**
UnifiedReward	0.4862	**0.5764**	0.6873	0.4808	**0.6041**	0.4196	0.5326
MixedReward	0.5068	0.5561	0.6847	0.4726	0.5819	**0.4742**	0.5395
Bench.-Aligned	0.7103	0.6940	0.7808	0.6995	0.7780	0.5885	0.7105

Table 17 reports category-level results. Reward labels refer to online-reward configurations rather than final evaluators. Benchmark-aligned feedback reaches 0.9932, 0.6207, and 0.7105 on GenEval, T2I-CompBench, and WISE, respectively; these results serve as controlled references rather than practical alternative-reward settings.

Figure 12 illustrates that different online-reward configurations can induce different optimization trajectories and outputs under the same latent-control procedure.

C.8 Robustness to Evaluation Models

We test sensitivity to the choice of final evaluator by re-scoring the same WISE generations with Qwen3.5-35B-A3B (Qwen Team, 2026), without re-optimization or regeneration.

Following the official WISE evaluation protocol, the evaluator receives the image, prompt, and explanation and returns a binary score in $\{0, 1\}$. We use temperature 0 with thinking disabled. The

Table 18: WISE results under Qwen3.5-35B-A3B evaluation. All images are scored post hoc with Qwen3.5-35B-A3B (Qwen Team, 2026) following the official WISE evaluation protocol; generation, optimization, early stopping, and output selection are unchanged. Best and second-best values among the methods reported in the table are shown in **blue bold** and cyan underline, respectively. Entries marked with * are reproduced by us from the released MILR implementation under the corresponding matched generation settings and re-scored with the same evaluator.

Method	Cultural↑	Spatio-Temporal		Natural Science			Overall↑
		Time↑	Space↑	Biology↑	Physics↑	Chemistry↑	
Non-reasoning Models							
BAGEL (Deng et al., 2025)	0.4125	0.3500	0.3083	0.2000	0.4417	0.2583	0.3520
SD3-Medium (Esser et al., 2024)	0.4700	0.4083	0.4000	0.2000	0.3750	0.2583	0.3850
FLUX.1-dev (Labs, 2024)	0.5225	0.4000	0.5333	0.1750	0.3750	0.2417	0.4160
HunyuanImage-3.0 (Team, 2026)	0.5250	0.3917	0.4833	0.3083	0.4500	0.2417	0.4350
Qwen-Image (Wu et al., 2025)	0.6275	0.5250	0.5583	0.3417	0.4833	0.2500	0.5100
FLUX.2-dev (Black Forest Labs, 2025)	0.6650	0.5667	0.6583	0.3667	0.5250	0.3750	0.5650
Qwen-Image-2.0 (Zhao et al., 2026)	0.8219	0.6500	0.8992	0.7917	0.8000	0.7479	0.7954
GPT-Image-1.5 (OpenAI, 2025)	0.8900	0.6917	0.8833	0.8000	0.7583	0.7750	0.8250
NanoBanana-Pro (Google DeepMind, 2025)	0.8975	0.8167	**0.9333**	0.8167	0.8667	0.8750	0.8760
Qwen-Image-Agent (Zhang et al., 2026c)	0.9200	**0.9167**	**0.9333**	0.8333	0.8667	**0.9000**	**0.9020**
Training-based Reasoning Models							
DeepGen 1.0 (Wang et al., 2026a)	0.6500	0.4100	0.7200	0.3900	0.5900	0.4500	0.5700
BAGEL (w/ CoT) (Deng et al., 2025)	0.7800	0.6333	0.5667	0.3750	0.5500	0.5083	0.6280
SenseNova-U1-8B-MoT (Diao et al., 2026)	0.6900	0.6000	0.7417	0.6333	0.7167	0.6917	0.6820
SenseNova-U1-A3B-MoT (Diao et al., 2026)	0.7500	0.6250	0.7417	0.6750	0.7667	0.6750	0.7180
SenseNova-U1-8B-MoT (w/ CoT) (Diao et al., 2026)	0.7975	0.7417	0.8583	0.7583	0.7750	0.8000	0.7910
SenseNova-U1-A3B-MoT (w/ CoT) (Diao et al., 2026)	0.8325	0.7750	0.9167	**0.8417**	0.8000	0.8417	0.8340
Test-time Reasoning Models							
Janus-Pro-1B (+MILR)* (Mi et al., 2026)	0.7125	0.5833	0.3833	0.5750	0.5000	0.1833	0.5520
Janus-Pro-7B (+MILR)* (Mi et al., 2026)	0.8975	0.8167	0.7917	0.6500	0.8417	0.6000	0.8030
Janus-Pro-1B (Chen et al., 2025)	0.3050	0.2333	0.2333	0.2167	0.3083	0.2000	0.2650
Janus-Pro-1B+TRILO	0.7950	0.7000	0.4500	0.6333	0.4917	0.3417	0.6320
Janus-Pro-7B (Chen et al., 2025)	0.3700	0.3500	0.2833	0.2833	0.4000	0.2333	0.3340
Janus-Pro-7B+TRILO	**0.9350**	0.8500	0.7750	0.6917	**0.8750**	0.5917	0.8280

evaluator is used only for post-hoc scoring and does not affect latent optimization, early stopping, or Retain selection.

Unlike the trajectory-level Retain analysis in Appendix C.4.4, this experiment evaluates aggregate method-level sensitivity to the choice of final evaluator.

Under Qwen3.5-35B-A3B evaluation, Janus-Pro-7B+TRILO scores 0.8280, compared with 0.8030 for matched MILR and 0.3340 for the frozen backbone. The corresponding Janus-Pro-1B scores are 0.6320, 0.5520, and 0.2650.

The MILR images are taken from our matched reproduced runs and are re-scored without changing their generation procedure. Thus, on both model scales, the within-backbone ordering

$$\text{Frozen backbone} < \text{MILR} < \text{TRILO}$$

is preserved under this alternative final evaluator. Absolute scores and margins nevertheless remain evaluator dependent.

C.9 Paired Statistical Robustness

We estimate prompt-level uncertainty for controlled TRILO–MILR comparisons using matched outputs from the reproduced runs, rather than the published aggregate MILR values used for literature comparison in the main tables.

The reruns use the same backbone, prompt set and ordering, run-level seed, decoding configuration, optimization reward, task-specific stopping criterion and threshold, refinement ratios, learning rate, maximum budget, and final evaluation protocol. At early stopping or budget exhaustion, each method applies its own output-selection rule.

For each benchmark, we jointly resample prompt-aligned outputs and recompute the official aggregate metric. Pairing is by prompt under the matched run-level protocol and does not require bitwise-identical initial generations. Define

$$\Delta = S_{\text{TRILO}} - S_{\text{MILR}}.$$

We use 10,000 paired-bootstrap replicates with seed 2027 and the official benchmark-specific aggregation rule.

Table 19: Paired sample-level bootstrap for TRILO versus a matched MILR rerun. $\widehat{\Delta}$ denotes the observed difference in overall benchmark score. Confidence intervals are percentile 95% intervals from 10,000 paired bootstrap replicates. Differences are computed from the underlying unrounded benchmark aggregates; displayed MILR and TRILO scores are rounded independently. The MILR scores come from the matched prompt-level rerun used for this analysis and may differ slightly from the literature-reported aggregates in the main benchmark tables.

Benchmark	N	MILR	TRILO	$\widehat{\Delta}$	95% CI
GenEval	553	0.9506	0.9932	**+0.0425**	$[+0.0259, +0.0606]$
T2I-CompBench	1800	0.5399	0.6207	**+0.0808**	$[+0.0724, +0.0895]$
WISE	1000	0.6098	0.7105	**+0.1007**	$[+0.0858, +0.1157]$

All three 95% confidence intervals exclude zero. The observed gains are $+0.0425$, $+0.0808$, and $+0.1007$ on GenEval, T2I-CompBench, and WISE, respectively, and the empirical fraction of bootstrap replicates satisfying $\Delta > 0$ is 1.000 for each benchmark.

Conditional on these fixed matched-run outputs, the advantage is therefore stable under prompt-level resampling. This analysis complements the multi-seed experiment below, which instead varies generation randomness.

C.10 Robustness Across Generation Seeds

Table 20: Robustness across generation seeds on Janus-Pro-7B. MILR and TRILO are evaluated under matched experimental settings while varying only the generation seed. Mean and standard deviation are computed over three seeds. Per-seed and mean differences are computed from the underlying unrounded scores, while displayed benchmark scores are rounded independently.

Benchmark	Method	Seed 41	Seed 42	Seed 43	Mean $\pm$ Std.
GenEval	MILR	0.9506	0.9379	0.9547	0.9477 ± 0.0087
	TRILO	0.9932	0.9878	0.9895	$\mathbf{0.9902 \pm 0.0028}$
	Δ	+0.0427	+0.0499	+0.0348	**+0.0424**
T2I-CompBench	MILR	0.5399	0.5422	0.5422	0.5414 ± 0.0013
	TRILO	0.6207	0.6265	0.6252	$\mathbf{0.6241 \pm 0.0030}$
	Δ	+0.0808	+0.0843	+0.0830	**+0.0827**
WISE	MILR	0.6098	0.5925	0.5784	0.5936 ± 0.0157
	TRILO	0.7105	0.7091	0.6992	$\mathbf{0.7063 \pm 0.0062}$
	Δ	+0.1007	+0.1166	+0.1208	**+0.1127**

Using three common run-level seeds on Janus-Pro-7B, MILR and TRILO follow the matched protocol in Appendix B.2. Because autoregressive stochastic decoding consumes the random-number stream sequentially, sharing a run-level seed does not imply bitwise-identical per-prompt initial images after the two optimization trajectories diverge.

Across all three evaluated seeds, TRILO scores higher than matched MILR on each benchmark. The average gains are $+0.0424$, $+0.0827$, and $+0.1127$ on GenEval, T2I-CompBench, and WISE, respectively. T2I-CompBench gains vary from $+0.0808$ to $+0.0843$ across seeds.

Together with the paired-bootstrap analysis, these results provide evidence beyond a single prompt sample and a single generation seed.

C.11 Efficiency Analysis

We report both cross-method resource context and a controlled TRILO–MILR efficiency comparison.

Training-based methods generally require additional data and accelerator resources, whereas MILR and TRILO introduce no parameter training. Because the methods in Table 21 differ in backbone,

Table 21: Resource requirements of representative reasoning-based image generation methods. We report training and inference information disclosed by prior work when available. Because methods use different backbones, hardware, generation pipelines, and evaluation protocols, the values provide resource context rather than a hardware-normalized efficiency comparison.

Method	Additional Training	Training Resources	Training Data / Procedure	Inference Hardware	Reported Inference Cost
T2I-R1 (Jiang et al., 2025)	✓	8×H800, 16 h	CoT + RL	1×A800[†]	15.261 s/sample[†]
Flow-GRPO (Liu et al., 2025)	✓	24×A800, ∼100 h	GRPO	–	–
IV-CoT (Li et al., 2026c)	✓	A800 80GB[‡]	128K image–text pairs	1×A800 80GB	1.693 s/sample
TwiG-RL (Guo et al., 2026)	✓	–	TWIG-50K + GRPO	1×A800[†]	25.362 s/sample[†]
LatentMorph (Chen et al., 2026)	✓	8×H200	20K-pair SFT + GRPO	–	44.3% lower[§]
MILR (Mi et al., 2026)	✗	–	None	1×RTX 4090[*]	See Table 22
TRILO	✗	–	None	1×RTX 4090	See Table 22

[†] T2I-R1 and TwiG-RL latencies are reported by IV-CoT under a unified protocol using one NVIDIA A800 80GB GPU, batch size 1, and 100 prompts. The measurement includes text processing, method-specific reasoning or intermediate generation, and final image synthesis, while excluding model loading.
[‡] IV-CoT reports training on NVIDIA A800 80GB GPUs, but does not specify the number of GPUs or total training wall-clock time.
[§] LatentMorph reports a 44.3% relative reduction in inference time against explicit-reasoning baselines on T2I-CompBench rather than an absolute hardware-normalized latency.
[*] MILR is reproduced by us under the same hardware and matched test-time experimental settings as TRILO; controlled rerun results are reported in Table 22.

hardware, pipeline, and evaluation protocol, this comparison is intended as resource context rather than a hardware-normalized efficiency ranking.

The controlled comparison follows the matched protocol in Appendix B.2 with $K = 20$, using one NVIDIA RTX 4090 GPU (24GB) and 16 Intel Xeon Platinum 8358P CPU cores. Wall-clock time includes initial generation, latent optimization, and benchmark evaluation. MILR uses dedicated released-code reruns rather than the published aggregate results reported in the main benchmark tables.

Table 22: Controlled performance–efficiency comparison with MILR. Both methods use the same Janus-Pro backbone, benchmark prompts, optimization rewards, stopping criteria, maximum optimization budget $K = 20$, hardware, and evaluation protocol. Avg. Steps denotes the mean number of executed latent-update steps over all benchmark prompts, including prompts that terminate without latent optimization. Time denotes measured end-to-end wall-clock cost, including initial generation, test-time latent optimization, and benchmark evaluation.

Method	GenEval			T2I-CompBench			WISE		
	Score ↑	Avg. Steps ↓	Time (h) ↓	Score ↑	Avg. Steps ↓	Time (h) ↓	Score ↑	Avg. Steps ↓	Time (h) ↓
MILR[*] (Mi et al., 2026)	0.9506	1.096	4.59	0.5399	11.682	98.29	0.6098	8.649	52.70
TRILO	**0.9932**	**0.810**	**3.49**	**0.6207**	**11.614**	**85.75**	**0.7105**	**8.047**	**45.58**
Wall-Time Reduction	–	–	**24.0%**	–	–	**12.8%**	–	–	**13.5%**

[*] MILR is reproduced from its released implementation under the same matched evaluation and hardware settings as TRILO. The scores therefore correspond to dedicated matched reruns and may differ from the literature-reported aggregates used in the main benchmark comparison.

Under this matched protocol, TRILO improves all three benchmark scores while reducing measured end-to-end time from 4.59 to 3.49 hours on GenEval, 98.29 to 85.75 hours on T2I-CompBench, and 52.70 to 45.58 hours on WISE. These correspond to reductions of 24.0%, 12.8%, and 13.5%, respectively.

Trace-level audit.

Mean executed updates per prompt decrease from 1.096 to 0.810 on GenEval, 11.682 to 11.614 on T2I-CompBench, and 8.649 to 8.047 on WISE, corresponding to reductions of 26.1%, 0.6%, and 7.0%. The corresponding numbers of updated text positions decrease by 33.2%, 0.7%, and 8.0%.

The aggregate wall-time-to-update ratios are 27.3 versus 28.1, 16.8 versus 14.8, and 21.9 versus 20.4 seconds for MILR and TRILO on GenEval, T2I-CompBench, and WISE, respectively. These quantities should not be interpreted as per-update latency because the numerator is total runtime, including initial generation, prompts requiring no optimization, reward computation, benchmark evaluation, and other pipeline overhead.

Table 23: Trace-level audit of the matched efficiency runs. N_{opt} denotes the number of prompts that execute at least one latent update. Total Steps is the number of executed latent-update steps over the full benchmark. Mean Opt. is the average number of steps conditional on optimization being activated. Text Updates denotes the total number of text-latent positions updated across all executed optimization steps. Sec./Step is the total measured wall-clock time divided by Total Steps and is reported only as an amortized diagnostic rather than an instrumented per-step latency.

Benchmark	Method	N_{opt}	Total Steps	Mean Opt.	Text Updates	Sec./Step	Sec./Prompt
GenEval	MILR	137	606	4.42	4,977	27.3	29.9
	TRILO	122	448	3.67	3,327	28.1	22.7
T2I-CompBench	MILR	1,445	21,027	14.55	252,590	16.8	196.6
	TRILO	1,442	20,905	14.50	250,884	14.8	171.5
WISE	MILR	778	8,649	11.12	110,857	21.9	189.7
	TRILO	751	8,047	10.72	102,027	20.4	164.1

GenEval's runtime reduction coincides with substantially fewer updates. T2I-CompBench, by contrast, has nearly identical update counts despite a 12.8% runtime difference, while WISE lies between these cases. Without stage-level timing, we therefore do not attribute the remaining runtime differences to individual components.

We interpret Table 22 as a controlled empirical performance–efficiency comparison rather than a mechanistic explanation of runtime.