EARLY TIME CONDITIONING FOR IMAGE VARIETY

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-image models often produce similar compositions across independent seeds, limiting the range of visual interpretations available for a prompt. We find that the clean-image estimate implied by the first velocity prediction can already reveal a strong layout hypothesis, which subsequent updates tend to refine rather than fundamentally overturn. Merely skipping the earliest updates is sufficient to increase output variety, but this intervention simultaneously changes both the time condition queried by the model and the integration intervals used by the sampler. Controlled decomposition attributes this effect primarily to the model's time query: holding each seed's latent state and integration interval fixed, later-stage time conditioning differentiates first-pass clean-image estimates that otherwise exhibit similar layouts across seeds. Based on this finding, we introduce Early Time Conditioning, which briefly queries the pretrained model with later-stage time conditions while retaining the native integration schedule, text conditioning, and guidance. Across FLUX.1-dev, FLUX.1-schnell, and Qwen-Image on 553 prompts from GenEval with eight outputs each, all four configurations increase spatial and global feature diversity and Vendi Score over native sampling. Dev's single-query rule reaches diversity comparable to Group Inference with near-native HPS, and diversity gains persist among equal-sized sets of GenEval-correct images. The intervention preserves independent sampling and the native model-evaluation count, while requiring neither cross-sample coordination nor internal feature manipulation, making it compatible with standard independent and batched generation pipelines.

1 INTRODUCTION

Text-to-image generation is useful not only for realizing a specified image, but also for exploring how a description could be pictured. A prompt often leaves composition, viewpoint, and subject placement open. Yet current models can repeatedly choose similar arrangements across independent random seeds (Parmar et al., 2026; Kim et al., 2026). Generating more individually plausible images then offers fewer distinct visual interpretations than the prompt permits. We study this same-prompt variety problem in pretrained flow-based image generators (Figure 1). Existing approaches diversify generation by adjusting guidance (Kynkäänniemi et al., 2024), optimizing initial noise (Kim et al., 2026; Harrington et al., 2026), or introducing repulsion between samples (Corso et al., 2024; Kirchhof et al., 2025; Singh et al., 2024; Bhat et al., 2026). Group inference improves the returned set by progressively selecting from a larger candidate population (Parmar et al., 2026). These approaches act on different parts of inference, with different requirements for optimization, cross-sample computation, internal feature manipulation, or candidate budgets. Our starting point is instead a question about the generator itself: when does its tendency toward similar compositions first become visible, and can the early computation that produces this tendency be directly controlled?

We inspect the clean-image estimate implied by the velocity output of the first forward pass. Despite starting from independent noise seeds, these estimates can already depict strikingly similar layouts, and subsequent updates tend to refine this early image hypothesis rather than fundamentally overturn its overall arrangement. This makes the beginning of sampling a natural place to intervene: if different seeds are mapped to similar image hypotheses at the outset, their trajectories have less opportunity to develop into distinct visual interpretations. Indeed, merely skipping the earliest updates is sufficient to increase output variety. Yet skipping is a coupled intervention: it changes both the time condition supplied to the network and the integration intervals through which its predictions

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.

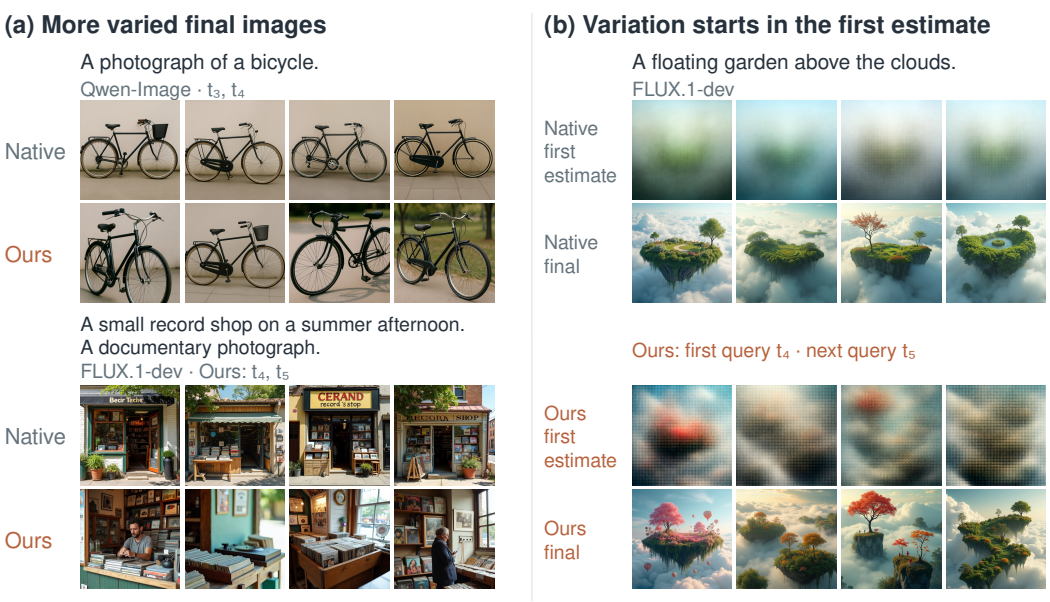

Figure 1: **Exploring a prompt through early time conditioning.** (a) Alternative viewpoints of a bicycle and exterior, interior, and close-up interpretations of a record shop. (b) First-forward clean-image estimates (Native) and implied estimates (Ours), with corresponding final images for a floating garden; changing the early time queries alters the emerging scene arrangement. Each example shows four selected, paired seeds; complete eight-seed sets appear in the supplement. Ours uses t_3, t_4 on Qwen and t_4, t_5 on Dev; Native uses the original time conditions. In (b), the altered first estimate uses only t_4, with t_5 applied at the next update. Integration schedules and prediction counts are unchanged; both first estimates are reconstructed at $\sigma_0 = 1$.

update the latent. This raises a more specific question: which of these changes is responsible for the increased variety?

We disentangle the model's time query from the sampler's integration interval. Holding each seed's latent state and integration interval fixed, querying a later-stage time condition differentiates first-pass clean-image estimates that otherwise exhibit similar layouts across seeds. Controlled continuations further attribute the resulting change in variety primarily to the time query rather than to the altered integration interval. Crucially, the effect persists when every native integration interval is restored: changing only a short prefix of time queries is sufficient to increase final-image variety. These observations identify early time conditioning as a direct control over the image hypotheses formed at the start of generation.

Based on this finding, we introduce Early Time Conditioning (ETC). During a short initial window, Early Time Conditioning queries the pretrained model using the time condition of a later sampling stage and then resumes the native time sequence. The latent is still advanced through every original integration interval, while text conditioning, guidance, and the sampler itself remain unchanged. The intervention adds no model evaluations and requires neither cross-sample coordination nor manipulation of intermediate features or attention. Each trajectory therefore retains the standard independent-sampling execution pattern, allowing Early Time Conditioning to be used one sample at a time or scaled through ordinary batching and parallel generation.

Experiments on FLUX.1-dev, FLUX.1-schnell, and Qwen-Image (Black Forest Labs, 2024; Wu et al., 2025) evaluate same-prompt image variety together with text alignment, image preference, and computational cost. We study the strength, duration, and temporal placement of the intervention, use fixed-state analyses to characterize how the time query changes early predictions, and compare against skipping and existing diversity-oriented inference methods. Across the three models, selected operating points increase same-prompt image variety, reflected in higher spatial feature diversity, with MUSIQ close to Native across all four rules. Runtime measurements further show that the conditioning-only intervention introduces negligible inference overhead. On 553 prompts

from GenEval with eight outputs each, all four selected configurations improve Patch-DINO, CLS-DINO, and Vendi over Native. Dev's single-query rule raises Patch-DINO from .5770 to .6658 with essentially unchanged mean HPS, while the two-query rule reaches .7237.

Our contributions are threefold: (1) we show that similar layouts can already be visible in clean-image estimates implied by the first model prediction, and identify the early time query as a controllable source of variation in these image hypotheses; (2) we introduce Early Time Conditioning, a simple training-free intervention that differentiates early image hypotheses while retaining the native integration schedule, independent sampling, and model-evaluation count; and (3) we establish its behavior across three pretrained generators through variety–quality evaluation, fixed-state prediction analysis, and ablations of intervention strength, duration, and placement.

2 Related work

Text-to-image diffusion and flow models. Diffusion models generate images by reversing a gradual noising process (Ho et al., 2020); latent diffusion makes high-resolution text-conditioned synthesis practical in a compressed image space (Rombach et al., 2022). Flow matching and rectified flow learn time-dependent vector fields that transport noise to data (Lipman et al., 2022; Liu et al., 2022). Recent text-to-image generators combine transformer backbones (Peebles & Xie, 2023) with flow-based formulations, including Stable Diffusion 3 (Esser et al., 2024), FLUX (Black Forest Labs, 2024), and Qwen-Image (Wu et al., 2025). Sampling schedules, network parameterizations, and numerical solvers are distinct design choices (Karras et al., 2022). In the flow samplers considered here, the network is normally queried at the current schedule coordinate, and the solver advances its prediction through the next integration interval. We instead vary the early network time input while retaining every native integration interval.

Same-prompt image variety. Guidance interval changes when classifier-free guidance is applied (Kynkäänniemi et al., 2024). Other methods diversify the initial conditions through noise optimization (Kim et al., 2026; Harrington et al., 2026). Particle Guidance and SPELL introduce interactions between generated samples (Corso et al., 2024; Kirchhof et al., 2025), whereas NegToMe and Feature Self-Guidance manipulate intermediate representations to discourage similar outputs (Singh et al., 2024; Bhat et al., 2026). A complementary approach, Group Inference, maintains a larger candidate population and uses external CLIP and DINO scoring models to progressively select a diverse returned set (Parmar et al., 2026). Early Time Conditioning instead changes the time input of each independent trajectory. It neither perturbs the text condition nor requires a target layout, feature repulsion, or additional candidates.

3 From early predictions to time-conditioning control

3.1 Early predictions and native sampling

We consider flow-based generators sampled with Euler updates (Lipman et al., 2022; Liu et al., 2022). Let z_i denote the latent at sampling step i, starting from Gaussian noise $z_0 = \epsilon$. A descending schedule $1 = \sigma_0 > \sigma_1 > \cdots > \sigma_T = 0$ specifies T integration intervals. The network receives a corresponding native time condition $t_i = g(\sigma_i)$, where g denotes the model's time-input convention. For text condition c, write the native Euler update as

$$v_i = v_\theta(z_i, t_i; c), \qquad h_i = \sigma_{i+1} - \sigma_i < 0, \qquad z_{i+1} = z_i + h_i v_i. \tag{1}$$

Here v_θ includes the model's original guidance computation. Since sampling follows decreasing σ, the signed interval h_i is negative: the plus sign in the Euler update advances against v_i. The latent decoder D converts z_T into the final image.

The underlying linear flow path is $z(\sigma) = (1-\sigma)x + \sigma\epsilon$ for a clean latent x and noise ϵ, with target velocity $\epsilon - x$ (Liu et al., 2022; Esser et al., 2024). The corresponding clean-image estimate obtained from a network forward pass is

$$q_i = z_i - \sigma_i v_i. \tag{2}$$

Equation 2 extrapolates the velocity to $\sigma = 0$, whereas Equation 1 takes only one interval. Decoding $D(q_0)$ therefore inspects what the first velocity implies about the image, without running the

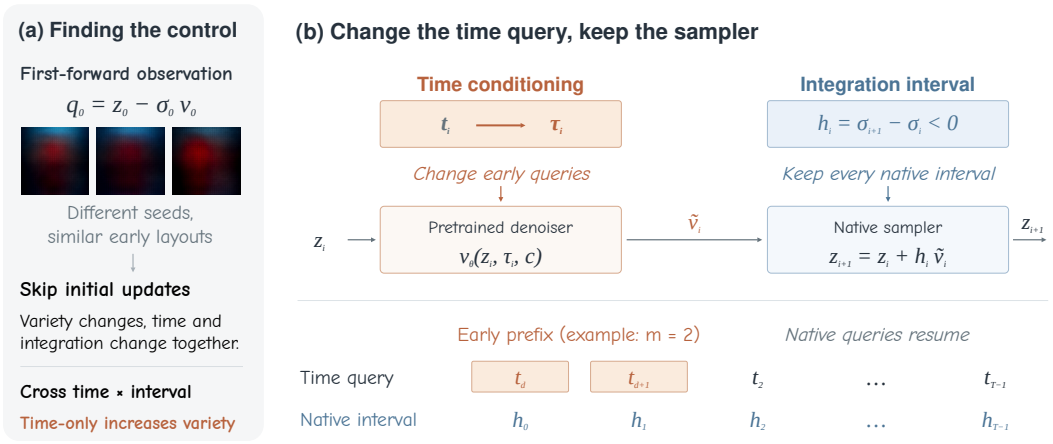

Figure 2: **Decoupling time conditioning from integration.** (a) First-forward observations lead to skip experiments; fixed-continuation controls separate time conditioning from integration. (b) Our intervention changes the early time queries, then resumes native conditioning. The integration schedule, text, guidance, and prediction count remain unchanged.

remaining denoising steps. Despite different initial noise seeds, the native first-pass reconstructions exhibit largely similar compositions (Figure 1b). Different noise samples therefore need not yield different early image hypotheses. Studies of diffusion dynamics describe a progression from coarse structure to finer detail, with output features emerging within distinct temporal windows (Wang & Vastola, 2023; Li & Chen, 2024). This identifies early prediction formation as a natural control point for variety: subsequent updates can develop closely related scene arrangements rather than distinct interpretations of the prompt.

To understand how an early image hypothesis enters the trajectory, we separate the estimate from its influence on the next state. Substituting Equation 2 into Equation 1 gives

$$z_{i+1} = \frac{\sigma_{i+1}}{\sigma_i} z_i + \left(1 - \frac{\sigma_{i+1}}{\sigma_i}\right) q_i, \qquad \sigma_i > 0. \tag{3}$$

The sampler therefore moves toward the estimated clean latent by a fraction set by the integration interval. This separates two possible interventions: changing the estimate through the network's time input, or changing how far the sampler moves toward it.

3.2 Separating time conditioning from integration

To test whether the earliest updates limit variety, we first skip these updates and observe that this can increase output variety (Section 4.3). Skipping s steps leaves the original noise unchanged, bypasses their predictions and updates, and starts with $v_\theta(\epsilon, t_s; c)$ and intervals $h_s, \ldots, h_{T-1}$. This changes the first time condition, the first integration interval, and the remaining sequence of updates together.

To separate the first two changes, we evaluate native and later-stage time conditions at the same initial latent and apply each prediction through either the native or later integration interval. All four controls then follow the same subsequent sampling schedule.

Inspecting the first-pass estimates reveals what changing the time query can do. For each seed, we retain the same initial noise and reconstruct both predictions at σ_0. Under native time conditioning, the first-pass reconstructions show similar compositions across seeds. Changing only the time condition produces more varied compositions from the same initial noise samples (Figure 1b). Through Equation 3, these changed predictions alter the first update and hence the state from which subsequent sampling proceeds. Our intuition is to redirect early scene formation, allowing subsequent native updates to develop and refine these alternative visual interpretations.

The controlled continuations test whether the changed predictions translate into greater final-image variety. With the subsequent sampling schedule fixed, changing the time query has a larger effect

on variety than changing the initial integration interval (Table 2). We then restore every native integration interval and change only the first time query. The variety gain persists (Section 4.3), showing that omitting updates is not necessary. This leads to a direct intervention: evaluate the current latent under a different time condition, while retaining the original sampler.

3.3 Early Time Conditioning

Our method changes the model time input during the first m updates, querying conditions associated with a later stage of sampling and a lower nominal noise level. Let τ_i be the chosen time condition at step i. We use

$$\widetilde{v}_i = v_\theta(z_i, \tau_i; c), \qquad z_{i+1} = z_i + h_i \widetilde{v}_i, \qquad \tau_i = t_i \text{ for } i \geq m. \tag{4}$$

All h_i retain their native values. Using zero-based indices and a positive integer offset d, we choose

$$\tau_i = \begin{cases} t_{i+d}, & 0 \leq i < m, \\ t_i, & m \leq i < T, \end{cases} \qquad m - 1 + d < T. \tag{5}$$

The offset d indexes the native schedule rather than a fixed numerical time difference. After the initial window, conditioning resumes at t_m, not t_{m+d}. For example, two changed queries produce the sequence $t_d, t_{d+1}, t_2, \ldots, t_{T-1}$, each evaluated at its current latent (Figure 2). For a coarse native time grid, we use a finer positive displacement δ in the normalized schedule coordinate: $\tau_i = g(\sigma_i - \delta)$ for $i < m$, choosing $0 < \sigma_i - \delta \leq 1$. To inspect the changed prediction, we reconstruct its implied clean-image estimate $\widetilde{q}_i = z_i - \sigma_i \widetilde{v}_i$ using the original integration coordinate σ_i, not the noise level associated with τ_i. This extrapolates the queried velocity from the current state to $\sigma = 0$.

Only the network's time input changes; the native guidance, sampler updates, and number of network evaluations are preserved, adding no denoising computation.

4 Experiments

4.1 Experimental setup

We evaluate FLUX.1-dev, FLUX.1-schnell (Black Forest Labs, 2024), and Qwen-Image (Wu et al., 2025) on all 553 GenEval prompts (Ghosh et al., 2023) at 512^2, with eight outputs per prompt. Native and our method retain all independently generated outputs, without filtering or sample interaction. We select our configurations on 12 randomly sampled prompts outside GenEval, favoring a good tradeoff between Patch-DINO gains and HPS loss, and fix them for evaluation: Dev uses $(m, d) = (1, 8)$ or $(2, 4)$, Qwen uses $(2, 3)$, and Schnell uses $m = 1, \delta = .025$. Schnell's four-step grid is too coarse for a whole-step shift, so the numerical displacement provides finer control. Appendix A.1 gives the development protocol and executed time inputs.

We compare Group Inference (GI) (Parmar et al., 2026), SPELL (Kirchhof et al., 2025), Temporal Score Rescaling (TSR) (Xu et al., 2026), and contrastive noise optimization (CNO) (Kim et al., 2026). GI progressively prunes 64 candidates to eight using external CLIP/DINO scores; SPELL repels concurrent samples; TSR rescales the predicted score; CNO optimizes initial noise. For Qwen, we also compare guidance interval (Kynkäänniemi et al., 2024). Appendix A.3 specifies implementations and budgets. Using DINOv2 features, Patch-DINO measures spatial diversity, CLS-DINO measures global diversity, and Vendi measures effective set diversity (Oquab et al., 2024; Friedman & Dieng, 2023). CLIP measures text alignment (Radford et al., 2021); HPSv2 and PickScore estimate preference (Wu et al., 2023; Kirstain et al., 2023); MUSIQ measures technical image quality (Ke et al., 2021). Complete metrics appear in Appendix B.1.

4.2 Main results

Early time conditioning increases the variety produced by independent trajectories at the native generation budget (Table 1). All four rules improve Patch-DINO, CLS-DINO, and Vendi over Native across three models, without additional candidates or denoising evaluations. The single-query Dev rule is a mild operating point; the two-query rule characterizes the stronger-variety regime.

Table 1: **Diversity, quality, and cost.** 553 GenEval prompts, eight outputs each. Cost is runtime relative to each implementation's Native control on an RTX A6000 (three repetitions). CLS-DINO and timing details are in Appendices B.1 and E.

	Diversity		Alignment	Preference		IQA	Cost
Method	Patch ↑	Vendi ↑	CLIP ↑	HPS ↑	Pick ↑	MUSIQ ↑	×Native ↓
FLUX.1-dev (native: 28 updates, guidance 3.5)							
Native	0.5770	2.837	32.407	0.2973	23.390	73.78	1.00
Group Inference	0.6617	3.269	32.182	0.2969	23.336	74.21	1.50
SPELL	0.6103	3.380	32.266	0.2886	22.866	70.73	1.00
TSR	0.5858	2.905	32.878	0.2945	23.264	72.40	1.00
CNO	0.6188	3.238	32.292	0.2930	23.111	72.54	1.45
Ours (1 query)	0.6658	3.372	31.857	0.2970	23.270	74.15	1.01
Ours (2 queries)	0.7237	3.945	31.326	0.2899	22.685	73.54	1.01
FLUX.1-schnell (native: 4 updates, guidance 0)							
Native	0.6070	2.774	32.892	0.2966	23.189	73.61	1.00
Group Inference	0.6891	3.312	32.764	0.2949	23.065	73.94	3.39
SPELL	0.6054	2.766	32.913	0.2950	23.059	72.83	1.00
TSR	0.6078	2.795	33.118	0.2976	23.199	73.30	1.00
CNO	0.6486	3.086	32.785	0.2939	23.032	72.83	4.03
Ours	0.6983	3.586	31.931	0.2915	22.760	73.80	1.00
Qwen-Image (native: 30 updates, CFG 4)							
Native	0.4065	1.709	34.409	0.3010	23.641	75.20	1.00
Group Inference	0.4632	1.823	34.427	0.3011	23.649	75.21	1.51
Guidance interval	0.5157	2.128	34.245	0.2989	23.565	75.52	1.01
SPELL	0.4603	1.839	34.307	0.3012	23.534	75.12	1.00
TSR	0.4187	1.760	34.911	0.2925	23.321	74.16	1.00
CNO	0.4651	1.848	34.400	0.3006	23.616	74.96	1.43
Ours	0.5718	2.210	34.265	0.2996	23.510	74.96	1.01

Dev's one-query rule already reaches GI-level diversity with near-native HPS and MUSIQ, without generating a larger candidate pool. Qwen shows a larger diversity advantage: GI and SPELL improve over Native, but ETC exceeds both in all three diversity metrics. Across models, all four ETC rules exceed GI in these metrics and SPELL in Patch-DINO. Even against stronger CNO optimization on Dev, one ETC query yields higher diversity, HPS, PickScore, and MUSIQ at lower cost. CNO and TSR retain stronger text alignment; on Qwen, CNO retains higher preference but less diversity. Full-set CNO results and matched baseline parameter responses appear in Appendices B.1 and B.2.

The feature-diversity gains persist among equal-sized sets of GenEval-correct outputs (Appendix D.1), while MUSIQ stays close to Native across all four rules. Stronger interventions can reduce text alignment and task correctness, revealing a controllable variety–fidelity tradeoff. Preference-score decreases occur both in genuine content failures and in alternative viewpoints judged visually acceptable by a blinded model-based assessment (Appendix D.4). In a complementary 20-prompt Dev study, human raters favor the single-query rule for variety and visual quality, with prompt adherence close to Native (Appendix D.3). ETC also complements preference-based selection: selecting eight of twelve Dev two-query outputs by HPS/Pick ranks retains higher Patch-DINO, CLS-DINO, and Vendi than GI, with comparable preference scores and measured runtime (Appendix F.3). Table 1 reports ETC without this optional selection.

The gains also extend without retuning to 96 long-form DPG-Bench prompts (Hu et al., 2024) and a separate 32-prompt evaluation at 1024^2 (Appendix B.3).

Visually, the changes include viewpoint, subject scale, and scene context (Figure 1). In Figure 3, Native repeatedly depicts an open book on a tabletop; ETC adds closed, upright, and flat-lay views alongside different framings of the open book. In these examples, ETC's variety advantage becomes clearer through the distinct scene interpretations it adds, which feature-distance means alone do not describe. Appendix G provides complete eight-image comparisons across models, including TSR and CNO.

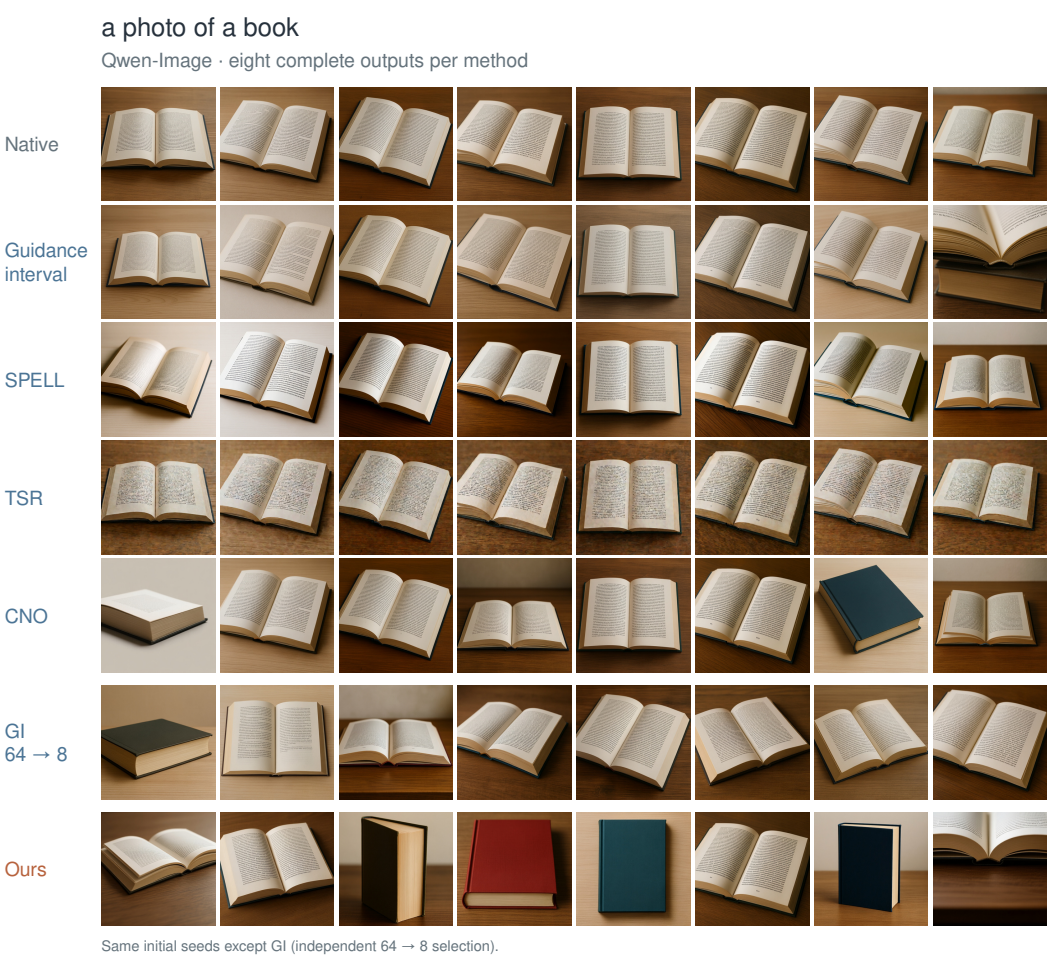

Figure 3: **Qualitative comparison on Qwen-Image.** Complete eight-image sets with Ours at the bottom. Non-GI methods start from the same seeds; CNO uses learning rate .02 and optimizes its initial noise. GI selects eight outputs from 64 candidates and is not paired by column. Ours varies the book's opening, orientation, and framing using the fixed t_3, t_4 rule.

ETC preserves Native's network-evaluation count and peak memory, with less than 0.8% measured runtime difference under matched execution (Appendix E). Unlike GI's candidate scoring and selection, CNO's optimization, or SPELL's sample interactions, ETC needs no additional stage or cross-sample coordination. Its independent trajectories retain Native's execution structure across batches and devices.

4.3 From skipping to time conditioning

The intervention changes the prediction before any state update is applied. Changing only the first time query increases first-estimate Patch-DINO on all 553 prompts for both Dev query settings and Qwen, and on 551 prompts for Schnell (Appendix C.1). We next distinguish the contribution of this changed prediction from that of the integration intervals to final variety.

Skipping one or two updates increases output variety on Dev (Table 2a), but changes the first query and the integration intervals together. Panel (b) separates these changes while holding the subsequent sampling schedule fixed.

Table 2: **From skipping to time-conditioning control.** FLUX.1-dev, 553 prompts and eight paired ablation seeds, distinct from Table 1. (a) Native and ETC retain the full schedule; Skip omits the stated prefix. (b) Every arm makes one initial write and then follows the same native suffix from index 3. Rescaled predictions use the other prediction's norm while retaining their own direction.

(a) Skipping versus conditioning only

Setting	Calls	Patch ↑	HPS ↑	Pick ↑
Native	28	0.5658	0.2970	23.402
Skip 1	27	0.6457	0.2970	23.350
Skip 2	26	0.6946	0.2942	23.107
Ours (1 query)	28	0.6606	0.2967	23.306
Ours (2 queries)	28	0.7112	0.2910	22.847

(b) Initial time query, integration interval, and norm

Velocity	Interval	Patch ↑	HPS ↑	Pick ↑
Native-time $v^{(0)}$	h_0	0.6595	0.29442	23.106
Native-time $v^{(0)}$	h_2	0.6541	0.29457	23.120
Later-stage $v^{(2)}$	h_0	0.6962	0.29409	23.096
Later-stage $v^{(2)}$	h_2	0.6946	0.29425	23.107
Rescaled $v^{(0)}$	h_2	0.6662	0.29418	23.084
Rescaled $v^{(2)}$	h_2	0.6917	0.29449	23.136

We cross predictions $v^{(0)} = v_\theta(\epsilon, t_0; c)$ and $v^{(2)} = v_\theta(\epsilon, t_2; c)$ with intervals h_0 and h_2, then use the native suffix from index 3. The $(v^{(2)}, h_2)$ cell is exactly skip-two. At either fixed interval, the later-stage query increases variety; at either fixed query, changing the interval has a much smaller effect (Table 2b). The query effect is not explained by prediction magnitude alone. The later prediction retains most of its gain when rescaled to the native norm; rescaling the native prediction to the later norm does not reproduce it. Prediction direction therefore matters beyond magnitude alone.

This query effect also survives restoring every native update. On the complete Dev schedule, changing only the first query to t_2 increases variety more than halving the original first update (Appendix C.4.1). Thus, the effect does not require omitting updates or merely weakening an unchanged prediction. Retaining these updates matters for short samplers: skipping can severely degrade four-step Schnell, whereas ETC retains coherent outputs and much higher preference (Appendix C.4). An equal-displacement control at the second query further distinguishes the two time directions: later queries increase final Patch-DINO, whereas earlier queries decrease it (Appendix C.3).

The changed predictions lead to a lasting increase in output variety. Across all 553 prompts, the decoded clean-image estimates of both Dev rules remain above Native in mean Patch-DINO throughout the 28 updates. One changed query produces an initial expansion, a contraction at the next native query, and subsequent growth; two changed queries sustain a larger early spread. The second query can also redirect a shared starting prediction: on 20 diagnostic prompts, two paths with identical first estimates and first updated states reach different final diversity when only their second time query differs (Appendix C.2). Early conditioning thus shapes how subsequent updates develop the prediction, with the feature-diversity gain persisting after native queries resume.

4.4 Intervention design

The controls support concentrating the intervention at the start: moving either Dev window to update 4 substantially weakens its diversity gain. Within this early window, strength and duration determine the response. At fixed $d = 4$, extending the window from one query to two increases variety at a greater preference cost (Appendix C.4).

Figure 4 shows a broad operating range rather than an isolated parameter setting. One query produces a rapid Patch-DINO rise followed by a plateau, with preference partly recovering at larger offsets. Two queries yield greater variety and a larger preference cost, peaking in Patch-DINO at $d = 7$. The paired images show how these choices change viewpoint, scale, and framing.

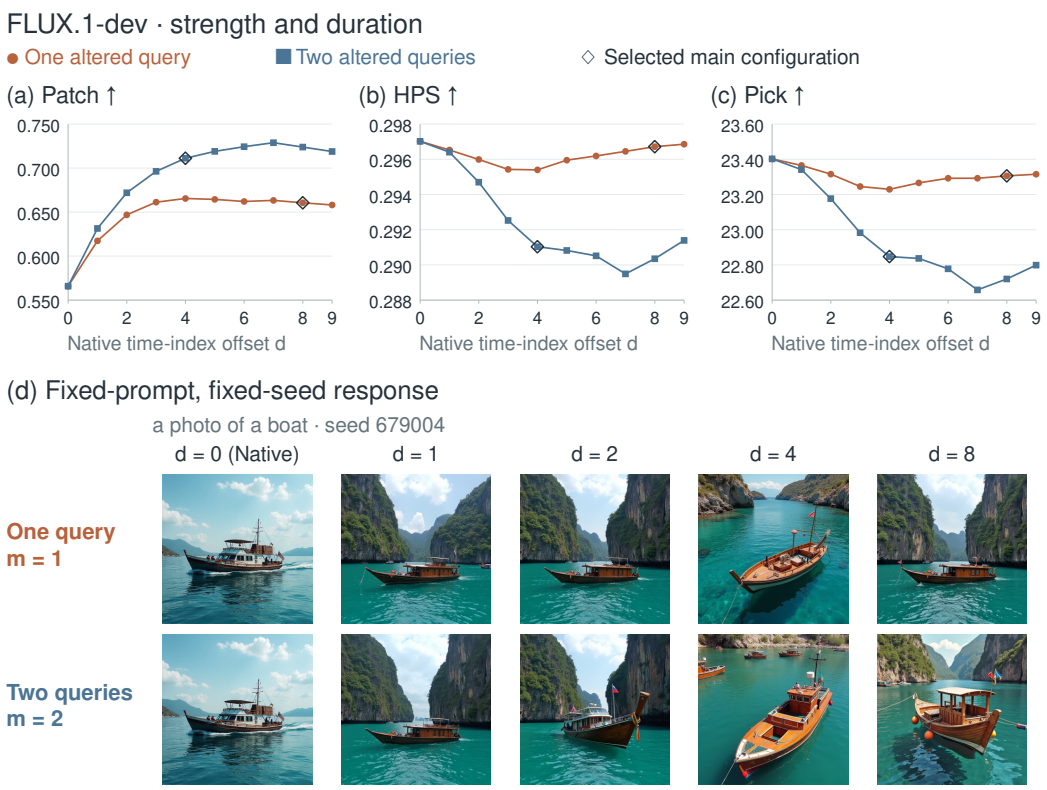

Figure 4: **Intervention strength and window length.** FLUX.1-dev. (a–c) Means over all 553 prompts, using the eight paired ablation seeds of Table 2; $d = 0$ is Native and diamonds mark the selected rules. (d) Fixed prompt and initial noise: columns vary offset d, rows vary window length m. Every image retains the native integration schedule.

The window also matters at comparable diversity. One Dev query at $d = 4$ and two at $d = 2$ give similar Patch-DINO, but the one-query path retains higher HPS, PickScore, and MUSIQ. This ordering holds on the full benchmark and independent long descriptions (Appendix D.5.1). Thus, preference is not determined by feature spread alone: the early query sequence also affects the operating point.

5 Conclusion

Our results show that image variety depends not only on the initial noise, but also on how a pretrained model turns that noise into early image hypotheses. By separating time conditioning from integration, we identify a lightweight control over early scene formation: changing a short prefix of time queries produces alternative visual interpretations that subsequent native updates can develop. Across three pretrained generators, Early Time Conditioning increases variety without retraining or additional denoising evaluations. Unlike candidate selection, sample repulsion, or noise optimization, ETC requires neither candidate management, cross-sample coordination, nor a separate optimization stage, preserving the native execution structure for large-scale generation. ETC shows that changing how a pretrained model interprets its initial noise can increase variety within the original sampling budget.

AI use statement

Generative AI tools were used as research and writing aids, supporting hypothesis refinement, experimental design, method implementation, and result analysis, as well as literature review, manuscript drafting and revision, and figure and table preparation. AI-assisted implementations and references

were checked through code tests, experimental controls, and source verification. The supplementary material separately describes the model-based visual assessment. The authors were deeply involved in all stages of the research and take responsibility for the final manuscript, results, and accompanying materials.

REFERENCES

Pradhaan S. Bhat, Rishubh Parihar, Abhijnya Bhat, and R. Venkatesh Babu. Don't settle at the mode! mitigating diversity collapse in pretrained flow models via feature self-guidance. *arXiv preprint arXiv:2606.27371*, 2026. Accepted to ECCV 2026.

Black Forest Labs. FLUX: Official inference repository. `https://github.com/black-forest-labs/flux`, 2024.

Guy E Blelloch. Programming parallel algorithms. *Communications of the ACM*, 39(3):85–97, 1996.

Chaofeng Chen and Jiadi Mo. IQA-PyTorch: PyTorch toolbox for image quality assessment. `https://github.com/chaofengc/IQA-PyTorch`, 2022.

Gabriele Corso, Yilun Xu, Valentin de Bortoli, Regina Barzilay, and Tommi Jaakkola. Particle guidance: non-i.i.d. diverse sampling with diffusion models. In *International Conference on Learning Representations*, 2024.

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 *International Conference on Machine Learning*, pp. 12606–12633, 2024. URL `https://proceedings.mlr.press/v235/esser24a.html`.

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023.

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

Anne Harrington, A. Sophia Koepke, Shyamgopal Karthik, Trevor Darrell, and Alexei A. Efros. It's never too late: Noise optimization for collapse recovery in trained diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43124–43134, 2026.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Vlad Hosu, Hanhe Lin, Tamas Sziranyi, and Dietmar Saupe. Koniq-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE Transactions on Image Processing*, 29:4041–4056, 2020.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

Hugging Face. Distributed inference. Diffusers documentation, n.d. URL `https://huggingface.co/docs/diffusers/main/en/training/distributed_inference`. Accessed September 26, 2026.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5148–5157, 2021.

Byungjun Kim, Soobin Um, and Jong Chul Ye. Diverse text-to-image generation via contrastive noise optimization. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=EVRMnAREc3.

Michael Kirchhof, James Thornton, Louis Béthune, Pierre Ablin, Eugene Ndiaye, and Marco Cuturi. Shielded diffusion: Generating novel and diverse images using sparse repellency. In *International Conference on Machine Learning*, 2025.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, 2023.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Advances in Neural Information Processing Systems*, 2024.

Marvin Li and Sitan Chen. Critical windows: non-asymptotic theory for feature emergence in diffusion models. *arXiv preprint arXiv:2403.01633*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

madebyollin. Tiny autoencoder for FLUX.1. Hugging Face model repository, 2024. URL https://huggingface.co/madebyollin/taef1.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Gaurav Parmar, Or Patashnik, Daniil Ostashev, Kuan-Chieh Wang, Kfir Aberman, Srinivasa Narasimhan, and Jun-Yan Zhu. Scaling group inference for diverse and high-quality generation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IyTNxjTuWT.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *International Conference on Computer Vision*, 2023.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Jaskirat Singh, Lindsey Li, Weijia Shi, Ranjay Krishna, Yejin Choi, Pang Wei Koh, Michael F. Cohen, Stephen Gould, Liang Zheng, and Luke Zettlemoyer. Negative token merging: Image-based adversarial feature guidance. *arXiv preprint arXiv:2412.01339*, 2024.

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

Binxu Wang and John J Vastola. Diffusion models generate images like painters: an analytical theory of outline first, details later. *arXiv preprint arXiv:2303.02490*, 2023.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, et al. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Yanbo Xu, Yu Wu, Sungjae Park, Zhizhuo Zhou, and Shubham Tulsiani. Temporal score rescaling for temperature sampling in diffusion and flow models. In *International Conference on Machine Learning*, 2026.

Guide to the supplement. Sections A–B give the reproduction settings, complete benchmark results, and transfer tests. Section C examines early predictions and controlled interventions; Section D brings together content correctness, human evaluation, and quality analyses. Sections E–F cover deployment cost and optional selection. Complete qualitative comparisons are collected in Section G.

A Experimental setup and reproduction

A.1 Sampling configurations

Table 3: Native settings shared by each model's conditioning-only variants (Black Forest Labs, 2024; Wu et al., 2025). Qwen uses a conditional and an unconditional evaluation for each update.

Model	Resolution	Updates	Guidance	Network evaluations
FLUX.1-dev	512^2	28	3.5	28
FLUX.1-schnell	512^2	4	0	4
Qwen-Image	512^2	30	4	60

Table 4: **Fixed evaluation configurations.** Executed network inputs after BF16 conversion. Only the listed initial queries change; all later queries, integration intervals, guidance settings and prediction counts remain native.

Model	Rule	Altered updates	Executed normalized inputs
FLUX.1-dev	Ours ($d = 8, m = 1$)	1	0.82421875
FLUX.1-dev	Ours ($d = 4, m = 2$)	2	0.921875, 0.89453125
FLUX.1-schnell	Ours ($\delta = .025, m = 1$)	1	0.9765625
Qwen-Image	Ours ($d = 3, m = 2$)	2	0.9375, 0.9140625

We preserve the scheduler's sigma array and update indices. The model-time argument is overridden separately. In the implementation, scheduler times are converted to the model's normalized input convention and represented at the model's working precision. Specifically, the time in 1000-unit convention is cast to BF16 before normalization by 1000. Table 4 gives the executed normalized inputs for the four main configurations, which can differ slightly from their nominal schedule coordinates. Both branches of explicit classifier-free guidance (Ho & Salimans, 2022) use the same changed time.

Operating-point development. We compare ETC operating points on 12 randomly sampled prompts outside GenEval, favoring configurations that balance Patch-DINO gains against HPS loss. The four selected configurations are fixed for evaluation on all 553 GenEval prompts. The full-set response curves characterize these configurations rather than determine them.

Discrete shifts. With offset d and window m, the query sequence is

$$t_d, t_{d+1}, \ldots, t_{d+m-1}, t_m, \ldots, t_{T-1}.$$

The integration sequence remains $h_0, h_1, \ldots, h_{T-1}$. In a repeat control with $m = 2$, the first two model queries both use t_d, but each is evaluated on its own evolving state.

Native schedules. All three models use a FlowMatchEulerDiscreteScheduler (von Platen et al., 2022). Starting from $u_i = 1 - i/T$ for $i = 0, \ldots, T-1$, Dev uses $\sigma_i = e^{\mu}/(e^{\mu} + u_i^{-1} - 1)$ with $T = 28$ and $\mu = .63$. Schnell uses $T = 4$ and $\sigma_i = u_i$ without dynamic shifting. For Qwen, first compute the same exponential shift as Dev with $T = 30$ and $\mu = .5 + .4(1024 - 256)/(8192 - 256)$, obtaining $\bar{\sigma}_i$. Its native terminal stretching gives

$$\sigma_i = 1 - \frac{.98(1 - \bar{\sigma}_i)}{1 - \bar{\sigma}_{T-1}}, \qquad i = 0, \ldots, T-1.$$

Each model appends $\sigma_T = 0$. The dynamic shifts use 1024 packed image tokens at 512^2 resolution. Native scheduler times are $1000\sigma_i$; the normalized network input follows the BF16 conversion described above. Schedule arithmetic is FP32 and network inference uses BF16.

Fractional shifts. Schnell's native normalized time sequence is $1, .75, .5, .25$. Its main $m = 1, \delta = .025$ rule produces $.975, .75, .5, .25$ before numerical conversion. A two-update control with the same displacement produces $.975, .725, .5, .25$. Every integration interval remains $-.25$. This differs from a fractional start at $.975$, whose first interval is $-.225$ and whose remaining intervals are $-.25$.

Applying ETC to another sampler. We recommend first comparing one- and two-query prefixes on a small fixed prompt and seed set, then varying the query displacement at a fixed window length. Use a fractional displacement when whole schedule steps are too coarse. Among settings reaching the desired variety, prefer those with smaller preference losses and inspect complete image sets for content and structure errors. Confirm the chosen setting on a second prompt set before freezing it for full evaluation. Throughout, change only the network time input and retain the sampler's native integration schedule.

A.2 Image-set measurements

We use `facebook/dinov2-base` (Oquab et al., 2024), with RGB images resized bilinearly to 256×256 and normalized by the ImageNet channel mean and standard deviation. From the final normalized hidden state, we extract the CLS token and spatial patch tokens and normalize each token to unit length. Patch-DINO averages corresponding-position cosine distances over all spatial tokens and all unordered image pairs. CLS-DINO instead compares the CLS features. For Vendi Score (Friedman & Dieng, 2023), the kernel is $K_{ab} = f_a^\top f_b$ for normalized CLS features f_a. For a set of B images, if λ_j are the eigenvalues of K/B, the score is $\exp(-\sum_j \lambda_j \log \lambda_j)$.

CLIP uses `openai/clip-vit-base-patch32` and its standard processor; the score is 100 times image–text cosine similarity (Radford et al., 2021). HPSv2 uses the `xswu/HPSv2` v2.0 checkpoint with its ViT-H/14 preprocessing (Wu et al., 2023). PickScore uses `yuvalkirstain/PickScore_v1` and the `laion/CLIP-ViT-H-14-laion2B-s32B-b79K` processor, reporting the checkpoint-scaled image–text score rather than a softmax over each set (Kirstain et al., 2023). MUSIQ measures no-reference technical image quality (Ke et al., 2021), using the KonIQ-trained checkpoint (Hosu et al., 2020) in PyIQA (Chen & Mo, 2022). It reads RGB images at their generated resolution and produces one score per image, separately from prompt alignment and set diversity. We first average image scores within each prompt and then average prompts.

A.3 External comparison settings

The comparisons use 553 prompts (Ghosh et al., 2023), 512^2 images, and eight returned outputs. The following settings specify candidate populations, optimization budgets, and model-specific execution paths.

Comparison coverage. GI provides the common candidate-selection comparison across all three models. SPELL couples eight trajectories through a repulsive update. TSR provides an independent-trajectory temperature-sampling comparison on all three models. Guidance interval is evaluated on Qwen, which explicitly computes conditional and unconditional predictions; changing Dev's distilled guidance embedding would be a different intervention. CNO provides an initial-noise optimization comparison on all three models.

Group Inference. The initial population has 64 candidates, progressively pruned to eight using two external pretrained scoring models: CLIP provides image–text alignment scores and DINO provides pairwise image-diversity scores (Parmar et al., 2026). These scores enter the released quadratic integer-program selection with $\lambda = 1.5$. Both scoring models are additional to the diffusion backbone; their inference and intermediate candidate decoding are included in GI's measured runtime. FLUX uses the released tiny autoencoder (madebyollin, 2024) for selection and its native VAE for final images. Dev uses 28 updates, guidance 3.5, pruning ratio .5, and the released first-prediction guidance of 1. Schnell uses four updates, guidance 0, and pruning ratio .9. Both FLUX comparisons use candidate seeds 229000–229063, with the first eight matching our independent trajectories. The Qwen adaptation uses its native VAE, 30 updates, CFG 4 with native norm rescaling, and $64 \to 32 \to 16 \to 8$ pruning. It retains the released selection-estimate convention, computed

from the post-update latent minus $\sigma_i v_i$. Dev's 28-update comparison is separate from the earlier 20-update benchmark.

SPELL. We implement the published SPELL repulsion (Kirchhof et al., 2025) on the clean-image estimates of eight concurrent trajectories and convert the resulting correction to the velocity parameterization. The radius is 160 for Dev, 279.3 for Schnell, and 117.3 for Qwen, with overcompensation 1.6 throughout. Denoiser predictions are evaluated with microbatch one; the repulsive update uses all eight samples. On Qwen it follows native classifier-free guidance (Ho & Salimans, 2022) and norm rescaling. A zero-radius control reproduces Native outputs exactly. The implementation follows the published equations through a common linear-flow interface; model-specific sampling and final decoding remain native.

Temporal Score Rescaling. TSR (Xu et al., 2026) uses the published image-generation settings $k = .93$ and $\sigma_{\mathrm{TSR}} = 3$ on all three models. The latter is TSR's rescaling parameter, not the sampler's noise schedule. We apply the released score-to-velocity conversion in FP32, retaining the native BF16 predictor, state updates, guidance, and decoder. The initial $t = 1$ prediction is unchanged; $k = 1$ reproduces Native exactly. Each prompt uses eight independent seeds, 229000–229007, with no cross-sample interaction or extra model evaluations.

Contrastive noise optimization. CNO (Kim et al., 2026) optimizes the initial noise for three Adam steps before native sampling. The main comparison uses learning rate .02 on Dev and Qwen, and .01 on Schnell. Section B.1 compares the default and stronger rates on Dev and Qwen. We retain the native BF16 predictor and sampling precision, with an FP32 optimization variable. The main comparison differentiates through the first clean-image estimate, following the released implementation. Each prompt uses the same eight initial seeds, 229000–229007, as our conditioning rules. On Schnell, we also evaluate the paper's stop-gradient formulation, which detaches the velocity when differentiating this estimate. Table 5 compares both gradient paths over all 553 prompts. The released-code path gives greater feature diversity at a higher cost; the stop-gradient path is cheaper but does not match the diversity of our conditioning-only rule.

Table 5: **CNO gradient paths on Schnell.** 553 prompts, eight outputs each. Both use three optimization steps at learning rate .01. Cost is relative to matched Native; correctness is the official GenEval image pass rate.

Gradient	Patch	CLS	Vendi	CLIP	HPS	Pick	MUSIQ	Cost	Correct (%)
Released-code gradient	0.6486	0.3465	3.086	32.785	0.2939	23.032	72.83	4.03	65.6
Stop-gradient	0.6335	0.3378	3.034	32.783	0.2933	22.952	73.18	1.72	64.4

Guidance interval. The Qwen guidance-interval control (Kynkäänniemi et al., 2024) uses the fixed interval $[.2, .9]$. The gate uses the native scheduler timestamp divided by 1000, including both endpoints. Inside the interval it uses CFG scale 4 and Qwen's native per-token norm rescaling to the conditional prediction's norm. Outside the interval it sends the conditional prediction directly to the scheduler. It retains both conditional and unconditional evaluations at every update, for 60 predictions in total; this implementation does not reduce model calls. The comparison retains the native sigma schedule and decoder.

SPELL operating-point checks. Additional checks on Dev used 24 development prompts and 24 separate confirmation prompts from the benchmark, with our main configurations fixed. Reducing overcompensation at fixed radius improves preference but brings diversity back toward Native on the confirmation set. The complete five-setting response and matched ETC, TSR, and CNO comparisons appear in Section B.2; the main table retains SPELL's default full-set comparison.

Main-comparison configurations. All four ETC rules in Table 4 use seeds 229000–229007 on each of the 553 prompts, retaining all eight outputs. The SPELL and guidance-interval comparisons use the same bank. Native sampling settings and executed time inputs are specified in Section A.1.

Seed pairing. The controlled conditioning studies pair initial noise exactly. The main comparison uses the seed banks specified above, with Native and GI summarized in Table 6. Each bank is reused for every prompt. Group selection returns a subset of a larger seed pool and therefore is not column-paired with eight independently generated outputs.

Table 6: **Seed banks for completed external rows.** GI lists all 64 candidate seeds, not eight selected output seeds. Other rows list the eight initial latent seeds.

Model	Method	Initial seed bank
FLUX.1-dev	Native (independent)	229000–229007
FLUX.1-schnell	Native	229000–229007
Qwen-Image	Native	229000–229007
Qwen-Image	Group Inference	140000–140063
FLUX.1-dev	Group Inference, native decoder	229000–229063
FLUX.1-schnell	Group Inference, native decoder	229000–229063

B Complete results and fixed-rule transfer

B.1 Complete benchmark measurements

Tables below retain all seven image metrics for the main comparison, including CLS-DINO. All methods return eight images for each of the 553 prompts. Cost is relative to the corresponding implementation's Native control; Appendix E gives the timing protocols and absolute latencies.

Table 7: **Complete diversity, quality, and cost measurements.** All 553 prompts, eight outputs each. Higher is better for image metrics; lower is better for runtime. Configurations are fixed across prompts.

Method	Patch	CLS	Vendi	CLIP	HPS	Pick	MUSIQ	Cost
FLUX.1-dev								
Native	0.5770	0.3132	2.837	32.407	0.2973	23.390	73.78	1.00
Group Inference	0.6617	0.3840	3.269	32.182	0.2969	23.336	74.21	1.50
SPELL	0.6103	0.3994	3.380	32.266	0.2886	22.866	70.73	1.00
TSR	0.5858	0.3240	2.905	32.878	0.2945	23.264	72.40	1.00
CNO	0.6188	0.3811	3.238	32.292	0.2930	23.111	72.54	1.45
Ours ($d = 8, m = 1$)	0.6658	0.3908	3.372	31.857	0.2970	23.270	74.15	1.01
Ours ($d = 4, m = 2$)	0.7237	0.4704	3.945	31.326	0.2899	22.685	73.54	1.01
FLUX.1-schnell								
Native	0.6070	0.2962	2.774	32.892	0.2966	23.189	73.61	1.00
Group Inference	0.6891	0.3783	3.312	32.764	0.2949	23.065	73.94	3.39
SPELL	0.6054	0.2950	2.766	32.913	0.2950	23.059	72.83	1.00
TSR	0.6078	0.2998	2.795	33.118	0.2976	23.199	73.30	1.00
CNO	0.6486	0.3465	3.086	32.785	0.2939	23.032	72.83	4.03
Ours ($\delta = .025, m = 1$)	0.6983	0.4181	3.586	31.931	0.2915	22.760	73.80	1.00
Qwen-Image								
Native	0.4065	0.1343	1.709	34.409	0.3010	23.641	75.20	1.00
Group Inference	0.4632	0.1566	1.823	34.427	0.3011	23.649	75.21	1.51
Guidance interval	0.5157	0.2076	2.128	34.245	0.2989	23.565	75.52	1.01
SPELL	0.4603	0.1581	1.839	34.307	0.3012	23.534	75.12	1.00
TSR	0.4187	0.1430	1.760	34.911	0.2925	23.321	74.16	1.00
CNO	0.4651	0.1617	1.848	34.400	0.3006	23.616	74.96	1.43
Ours ($d = 3, m = 2$)	0.5718	0.2261	2.210	34.265	0.2996	23.510	74.96	1.01

Stronger noise optimization. We test larger CNO learning rates on Dev and Qwen while retaining three optimization steps. Rates .015 and .02 were compared on 24 development prompts; .02 was selected for greater Patch-DINO subject to HPS and PickScore floors relative to two-query ETC, and then confirmed on 24 separate prompts before full evaluation. The floors permit HPS and

PickScore losses of at most .001 and .10, respectively. Both prompt cohorts belong to the 553-prompt benchmark; all eight outputs are retained.

Table 8: **Stronger CNO on FLUX.1-dev.** All 553 prompts, eight matched initial seeds, and three optimization steps. The main comparison uses CNO at learning rate .02; .01 is the default-rate reference. Both ETC rules remain fixed. Cost is measured relative to matched Native.

Method	Patch	CLS	Vendi	CLIP	HPS	Pick	MUSIQ	Cost
Native	0.5770	0.3132	2.837	32.407	0.2973	23.390	73.78	1.00
CNO (lr=.01)	0.5942	0.3404	3.003	32.373	0.2960	23.309	73.34	1.45
CNO (lr=.02)	0.6188	0.3811	3.238	32.292	0.2930	23.111	72.54	1.45
Ours (one query)	0.6658	0.3908	3.372	31.857	0.2970	23.270	74.15	1.01
Ours (two queries)	0.7237	0.4704	3.945	31.326	0.2899	22.685	73.54	1.01

On Dev, the larger rate increases all three diversity measures but lowers preference and MUSIQ. One ETC query exceeds this stronger CNO setting in diversity, both preference scores, and MUSIQ, with near-native rather than $1.45\times$ runtime (Table 8). CNO retains higher CLIP alignment. This comparison shows that ETC's diversity advantage need not come at a greater preference cost than noise optimization. The difference persists among equal numbers of GenEval-correct images: one-query ETC reaches Patch-DINO .6545 versus .6001 for stronger CNO over 414 jointly eligible prompts. Overall correctness is 60.6% and 61.3%, respectively (Section D.1).

Table 9: **Stronger CNO on Qwen-Image.** All 553 prompts and eight matched initial seeds. Both CNO settings use three optimization steps. The main comparison uses .02; .01 is the default-rate reference. ETC uses the fixed t_3, t_4 rule. Cost is matched runtime relative to Native (Section E); higher is better for image metrics, lower for cost.

Method	Patch	CLS	Vendi	CLIP	HPS	Pick	MUSIQ	Cost
Native	0.4065	0.1343	1.709	34.409	0.3010	23.641	75.20	1.00
CNO (lr=.01)	0.4453	0.1528	1.797	34.416	0.3009	23.625	75.12	1.42
CNO (lr=.02)	0.4651	0.1617	1.848	34.400	0.3006	23.616	74.96	1.43
Ours	0.5718	0.2261	2.210	34.265	0.2996	23.510	74.96	1.01

On Qwen, the larger learning rate increases all three diversity measures with only small changes in preference (Table 9). It narrows, but does not close, the diversity gap to ETC. CNO retains higher HPS and PickScore, while the two methods have nearly identical MUSIQ. Thus, the diversity difference is not explained by the default CNO learning rate alone. With equal numbers of GenEval-correct images (Section D.1), ETC also retains higher Patch-DINO than stronger CNO: .5601 versus .4599 over 516 jointly eligible prompts. CNO retains higher overall correctness, 85.9% versus ETC's 83.0%.

B.2 Matched operating-point comparisons

To compare parameter responses on the same inputs, we evaluate 31 Dev configurations on 24 development prompts and 24 separate confirmation prompts, using seeds 229000–229007 throughout. Each cohort contains four prompts from each GenEval category; all belong to the 553-prompt benchmark. Every configuration retains eight outputs, and the four main ETC rules remain fixed. This comparison uses the main benchmark's seed bank, separately from the full-set ablation seed bank (679000–679007).

The tested settings comprise Native; one- and two-query ETC at $d = 1, \ldots, 9$; CNO learning rates $.01, .015, .02$ with three optimization steps; TSR $(k, \sigma_{\mathrm{TSR}}) = (.93, 3), (.90, 3), (.90, 6), (.85, 6)$; and SPELL overcompensation $1.6, .8, .4, .2, .1$ at fixed radius 160. Figure 5 shows the complete parameter responses.

Reducing SPELL's repulsion restores preference, but also brings its diversity back toward Native. On confirmation prompts, the four mild settings span Patch-DINO .6019–.6067, around Native's .6043. Stronger CNO increases diversity more, whereas stronger TSR mainly lowers preference over the tested range. The fixed one-query ETC rule retains higher Patch-DINO and both preference

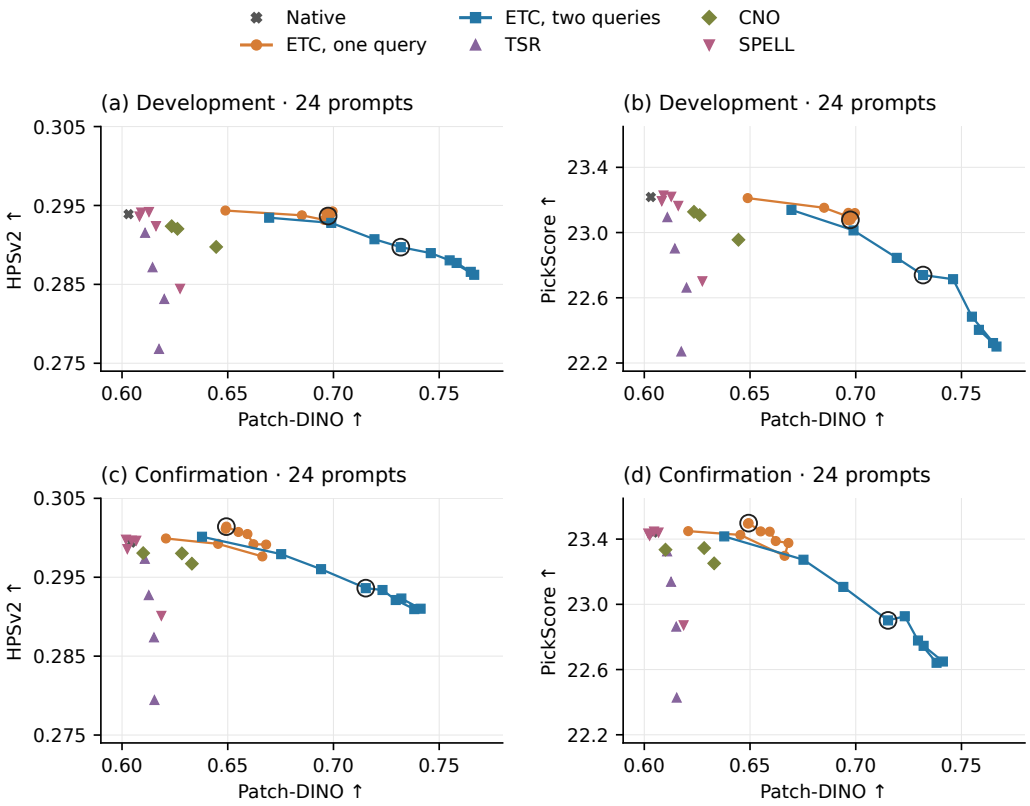

Figure 5: **Matched diversity–preference responses on Dev.** Each point is a configuration mean over 24 prompts and eight outputs per prompt. Development and confirmation cohorts use the same axes. Lines connect ETC offsets in order, not a fitted frontier; circles outline the fixed one-query $d = 8$ and two-query $d = 4$ rules. Baseline points show the parameter settings listed above. Higher and farther right indicate higher preference and Patch-DINO, respectively.

means than every tested baseline setting on the confirmation cohort. The development cohort is less uniform: some mild SPELL settings have slightly higher preference than this ETC rule, but much lower diversity. The full-set stronger-CNO comparison above provides the corresponding 553-prompt evaluation.

Table 10: **Content-correct variety on the confirmation cohort.** ETC is the fixed one-query $d = 8$ rule. Correctness uses all 24 prompts and eight outputs each. Patch-DINO uses equal numbers of correct images per method on the jointly eligible prompts (N), as in Section D.1. Each row is a separate paired comparison.

		Correct (%)		Valid-only Patch	
Comparator	N	Comparator	ETC	Comparator	ETC
TSR (default)	16	64.1	63.5	0.5959	0.6420
CNO (lr .02)	18	64.6	63.5	0.6151	0.6452
SPELL (.4)	17	64.6	63.5	0.5930	0.6435
SPELL (.1)	17	63.0	63.5	0.5926	0.6435

The diversity difference remains when comparing equal numbers of GenEval-correct images (Table 10). These comparisons include default TSR, stronger CNO, and the mild SPELL settings with the highest development-set HPS and PickScore, respectively. Correctness rates are similar on this cohort, while ETC retains higher within-valid Patch-DINO in each comparison.

B.3 FIXED-RULE TRANSFER

Transfer to long-form prompts. We separately evaluate the frozen rules on 96 preselected descriptions from DPG-Bench (Hu et al., 2024). The subset was sampled before generation from prompts of 20–90 words; all methods use eight matched seeds per prompt. Table 11 shows that Patch-DINO rises on each model. The single-query Dev rule keeps mean HPSv2 nearly unchanged, while the stronger Dev rule and the other two models incur larger preference changes.

Table 11: **Frozen-rule transfer to long-form prompts.** A preselected 96-prompt DPG-Bench subset, eight paired images per prompt at 512^2. All scores use the same complete image sets.

Model	Rule	Patch	CLS	Vendi	CLIP	HPS	Pick	MUSIQ
Dev	Native	.529	.229	2.359	34.04	.2933	22.32	73.75
	Single t_8	.590	.256	2.530	34.07	.2933	22.30	74.25
	Double t_4, t_5	.645	.291	2.761	33.62	.2886	22.00	72.85
Schnell	Native	.517	.206	2.220	34.65	.2911	21.99	74.55
	First .975	.603	.264	2.588	33.97	.2882	21.85	74.09
Qwen	Native	.436	.107	1.596	35.01	.2965	22.48	70.74
	Double t_3, t_4	.586	.205	2.154	34.73	.2923	22.18	70.85

The complete paired bus example appears in Figure 32.

Resolution transfer. Table 12 evaluates all four frozen rules at 1024^2 on 32 preselected prompts with eight paired seeds each. The model-specific native schedule is recomputed at the new resolution; the query offsets, Schnell's nominal first condition, guidance, and update count are not retuned. All four rules retain higher Patch-DINO, CLS-DINO, and Vendi than Native. Dev's single query keeps preference close to Native, while the stronger Dev rule and Schnell incur larger preference costs. The feature-diversity gain therefore transfers to higher resolution under the same query rules.

Table 12: **Fixed-rule transfer at 1024^2.** Thirty-two preselected prompts, eight paired seeds, and native update counts: Dev 28, Schnell 4, Qwen 30. All seven metrics use the same complete image sets.

Model	Rule	Patch	CLS	Vendi	CLIP	HPS	Pick	MUSIQ
FLUX.1-dev	Native	0.5534	0.2823	2.659	32.779	0.2992	23.533	73.61
FLUX.1-dev	Ours: one query	0.6451	0.3411	3.031	31.981	0.3001	23.523	73.81
FLUX.1-dev	Ours: two queries	0.6978	0.4279	3.610	31.561	0.2950	23.014	73.98
FLUX.1-schnell	Native	0.6305	0.3295	3.045	33.191	0.2969	23.203	72.37
FLUX.1-schnell	Ours	0.7323	0.4816	4.096	31.158	0.2893	22.320	74.63
Qwen-Image	Native	0.3683	0.1287	1.684	34.258	0.3024	23.698	74.57
Qwen-Image	Ours	0.6132	0.2655	2.498	34.047	0.2990	23.402	74.61

Paired resolution and update-count controls. The following smaller study compares 512^2 and 1024^2 on the same prompts and seeds; Dev additionally uses 40 rather than 28 updates at 512^2. These controls use prompts 53, 113, 193, 345, 412, and 531, seeds 339000–339003, and four outputs per prompt. Native and the altered rule share initial noise and the entire integration schedule within each setting. Native schedules are recomputed for the chosen resolution and update count; the selected query index or nominal Schnell condition is not retuned.

Dev retains its feature-diversity gain at both higher resolution and a larger update count (Table 13). At 1024^2, the single-query rule also remains closer to Native in HPS and PickScore than at 512^2. Both Schnell and Qwen retain their feature-diversity gains at 1024^2 without retuning. Schnell's larger spread comes with a greater preference cost, despite improved technical quality. Qwen increases Patch on all six prompts at this resolution, with a smaller mean PickScore reduction.

Guidance sensitivity. Dev uses guidance 2.5, 3.5, and 5.0 with the same six prompts, four seeds, 512^2 resolution and 28 updates. Both rules increase mean Patch at every guidance value. As guidance increases, Native becomes less varied and the conditioning intervention has a larger diversity

Table 13: **Paired resolution and update-count controls.** Six prompts, four paired outputs per setting. Size denotes the image side length in pixels. Dev retains offsets $d = 8/d = 4$ for one/two queries; Schnell uses first condition .975; Qwen uses t_3, t_4 with CFG 4. All metrics are higher-is-better.

Size / steps / rule	Patch	CLS	Vendi	CLIP	HPS	Pick	MUSIQ
FLUX.1-dev: resolution and update count							
512 / 28 / Native	0.4898	0.2507	1.900	33.431	0.3027	23.889	70.71
512 / 28 / Ours: one query	0.6184	0.3700	2.262	32.316	0.2976	23.345	72.92
512 / 28 / Ours: two queries	0.6710	0.4867	2.613	31.664	0.2941	23.117	73.33
512 / 40 / Native	0.4847	0.2713	1.986	33.089	0.3000	23.784	70.49
512 / 40 / Ours: one query	0.5926	0.3134	2.133	32.542	0.2984	23.427	72.65
512 / 40 / Ours: two queries	0.6481	0.4440	2.490	32.580	0.2960	23.286	71.71
1024 / 28 / Native	0.5285	0.2390	1.889	33.133	0.3050	23.867	71.67
1024 / 28 / Ours: one query	0.6109	0.3212	2.116	33.058	0.3047	23.841	72.20
1024 / 28 / Ours: two queries	0.6955	0.3888	2.348	32.827	0.3007	23.430	72.56
FLUX.1-schnell: four updates; size / rule							
512 / Native	0.5832	0.2575	1.970	33.567	0.3001	23.518	74.37
512 / Ours	0.6453	0.3067	2.149	32.477	0.2954	23.252	73.32
1024 / Native	0.6427	0.2988	2.115	33.083	0.3022	23.613	70.37
1024 / Ours	0.7356	0.4454	2.593	30.079	0.2919	22.518	74.00
Qwen-Image: thirty updates; size / rule							
512 / Native	0.3965	0.1184	1.449	34.514	0.3009	23.746	74.86
512 / Ours	0.5340	0.1803	1.655	33.964	0.3011	23.747	74.14
1024 / Native	0.3673	0.1003	1.391	34.191	0.3038	23.834	74.23
1024 / Ours	0.5869	0.2657	1.980	34.109	0.3004	23.624	74.87

effect. Two early queries yield a greater response than one, together with a larger PickScore cost at each guidance value.

Table 14: **Dev guidance sensitivity without retuning.** Every method is compared with Native at the same guidance. The 3.5 reference reuses the corresponding practical-setting images.

Guidance / rule	Patch	CLS	Vendi	CLIP	HPS	Pick	MUSIQ
2.5 / Native	0.5806	0.2999	2.116	32.650	0.3005	23.758	71.90
2.5 / Ours: one query	0.6480	0.3880	2.366	32.821	0.2978	23.439	73.52
2.5 / Ours: two queries	0.7104	0.5258	2.804	31.521	0.2929	23.046	73.75
3.5 / Native	0.4898	0.2507	1.900	33.431	0.3027	23.889	70.71
3.5 / Ours: one query	0.6184	0.3700	2.262	32.316	0.2976	23.345	72.92
3.5 / Ours: two queries	0.6710	0.4867	2.613	31.664	0.2941	23.117	73.33
5.0 / Native	0.4610	0.2050	1.770	33.559	0.3039	23.921	71.46
5.0 / Ours: one query	0.5937	0.2721	2.021	33.013	0.2994	23.467	73.05
5.0 / Ours: two queries	0.6756	0.4421	2.534	31.961	0.2955	23.065	72.08

C Early predictions and controlled interventions

C.1 First-forward response across models

We extend the first-forward comparison to all 553 GenEval prompts, with eight matched initial noise seeds per prompt. For Native, we decode $q_0 = z_0 - \sigma_0 v_\theta(z_0, t_0; c)$; for a changed time query τ_0, we decode the implied clean-image estimate $\widetilde{q}_0 = z_0 - \sigma_0 v_\theta(z_0, \tau_0; c)$. Both use the same initial state and actual integration coordinate $\sigma_0 = 1$. No subsequent sampling updates enter this comparison.

Changing only the time query increases first-estimate Patch-DINO on all 553 prompts for both Dev queries and Qwen, and on 551 prompts for Schnell. The response therefore extends beyond the illustrated cases. Native first-estimate spread differs across models: Schnell already has greater Patch-DINO dispersion than Dev or Qwen.

Table 15: **First-forward response.** All 553 prompts, eight paired seeds per setting. Metrics are computed on decoded first-pass estimates, not final images. For the two-query rules, this table measures only the first changed query.

Model	First query	Patch ↑	CLS ↑	Vendi ↑	CLIP ↑
FLUX.1-dev	Native t_0	0.2527	0.1232	1.675	24.20
	t_8	0.5791	0.5627	4.753	21.71
	t_4	0.6002	0.3935	3.484	24.24
FLUX.1-schnell	Native t_0	0.5837	0.2739	2.620	33.15
	.975	0.6853	0.4095	3.510	32.22
Qwen-Image	Native t_0	0.2784	0.1187	1.663	25.66
	t_3	0.5513	0.4228	3.651	26.15

The response evolves through subsequent updates. Dev's t_8 query gives greater first-estimate CLS-DINO and Vendi than t_4, whereas the two-query t_4, t_5 rule produces greater final-image diversity than the single-query t_8 rule. The first-pass comparison isolates the immediate response to the time input; the final-output experiments evaluate its effect together with subsequent sampling. Section C.2 follows this transition through the intermediate predictions and isolates the effect of a second changed query.

C.2 From the first estimate to the final image

We follow Native and both Dev rules through all 28 updates on all 553 prompts, using eight matched seeds per prompt. At update i, we decode $q_i = z_i - \sigma_i v_\theta(z_i, \tau_i; c)$ with the actual integration coordinate σ_i. This is the implied clean-image estimate during the changed queries and the native clean-image estimate thereafter; the last point separately measures the final decoded latent.

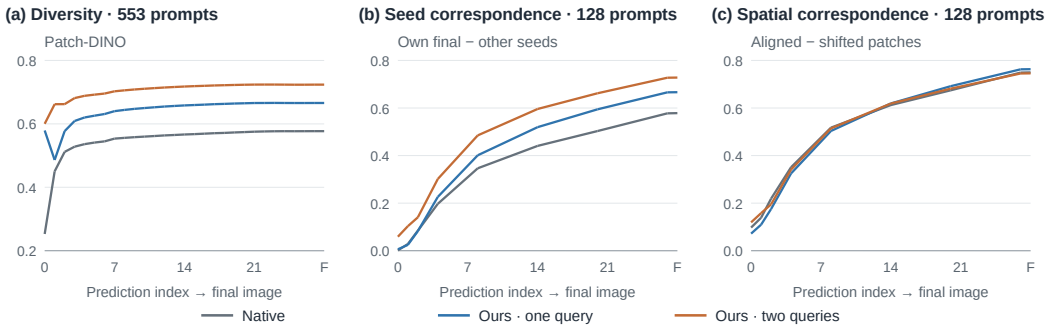

Figure 6: **Diversity and spatial correspondence along generation.** FLUX.1-dev, eight paired seeds per prompt. (a) Patch-DINO over all 553 prompts. (b–c) DINO patch correspondence with the final image on 128 fixed prompts spanning all six GenEval categories: same-seed alignment minus alignment with other seeds of the same prompt (b), or minus a spatially shifted version of the same final image (c). Points are means over prompts; F denotes the final image.

Both interventions produce an immediate increase in first-estimate variety, and their mean Patch-DINO remains above Native at every subsequent update (Figure 6a). The one-query rule shows a temporary contraction at the next, native-conditioned prediction before increasing again. The two-query rule sustains a larger spread through the early updates and reaches greater final diversity. Thus, subsequent native sampling develops the changed prediction into a more varied output set.

To examine the spatial link to the final image, we use normalized DINOv2 patch features (Oquab et al., 2024) on a fixed, category-balanced subset of 128 prompts (22 each for single- and two-object prompts, 21 each for the other four categories). For each estimate, we average cosine similarity at corresponding patch locations with its own final image. We subtract either the mean similarity to the other seven final images under the same prompt, or the similarity to its own final patch grid

cyclically shifted by four patches along both spatial axes. These controls distinguish seed-specific and spatially aligned correspondence from shared content.

The two-query rule has a clearer seed-specific association with its final image from the first estimate. The one-query rule initially increases cross-seed diversity much more than seed-specific correspondence, and its spatial-shift contrast is initially below Native's. Its correspondence strengthens during later updates. The two paths therefore reach increased final variety through different early trajectories: first-estimate spread and correspondence with the final image develop on different time courses.

Same first estimate, different second query. To isolate the effect of the second query, we compare one and two altered queries at the same offset $d = 4$ on 20 diagnostic prompts (eight randomly selected prompts and twelve preference-loss cases), with seeds 679000–679007. Both arms start from the same noise, use t_4 for their first prediction, and therefore produce exactly the same first estimate and first updated state. Only the second query differs: the one-query arm resumes at t_1, while the two-query arm uses t_5. Both subsequently follow the native queries and intervals.

Table 16: **Same first estimate, different subsequent response.** Mean Patch-DINO over 20 prompts and eight paired seeds. The difference is two queries minus one. For q_2 onward, both arms use the same native time query at their respective states.

Readout	One query	Two queries	Difference
q_0	0.607	0.607	+0.000
q_1	0.552	0.668	+0.116
q_2	0.631	0.674	+0.043
q_4	0.650	0.709	+0.059
q_8	0.670	0.726	+0.056
Final image	0.695	0.748	+0.053

The second changed query expands the feature spread immediately; the difference remains after native conditioning resumes and in the final images (Table 16). The early window offers more than one opportunity to redirect generation: subsequent queries can develop different continuations from an identical first estimate.

C.3 Which direction should the time query move?

Does any departure from the native time condition increase variety, or does the direction of the change matter? We compare equal displacements toward earlier- and later-stage time conditions at the second forward pass of FLUX.1-dev. The first update is native in all three arms, giving the same incoming state z_1 for each seed. Only the second time input changes; all 28 integration intervals and the subsequent native queries are retained. This position permits both directions within the model's time range. The executed BF16 inputs in Table 17 are equally spaced around the native input. We reconstruct the corresponding implied clean-image estimate using the unchanged σ_1.

Table 17: **Equal-displacement time-query control.** FLUX.1-dev, all 553 prompts and eight paired seeds (679000–679007). The intervention is at the second forward pass, not the first. q_1 Patch measures the decoded estimate before that update; the remaining metrics measure final images. Correct is the percentage of GenEval-correct outputs.

Query	Time input	q_1 Patch	Final Patch	HPS	Pick	Correct (%)
Native	0.9804688	0.4412	0.5658	0.2970	23.402	63.9
Later	0.9609375	0.4964	0.5885	0.2970	23.376	63.2
Earlier	1.0000000	0.2665	0.5537	0.2972	23.402	64.4

The two directions have opposite effects on Patch-DINO: a later-stage query increases feature diversity in the estimates and final images, whereas an earlier-stage query decreases it. The later-stage query increases final Patch-DINO on 451 of 553 prompts, while the earlier-stage query decreases it on 373. The effect is not confined to the initial 12-prompt direction check: the remaining 541 give a later-stage-query gain of .0227, compared with .0227 over the full set. The final preference means remain close to Native.

The direction also matters among content-correct outputs. Using the equal-count protocol of Section D.1, later-stage-query Patch-DINO increases from .5466 to .5706 over 426 jointly eligible prompts; earlier-stage-query Patch-DINO decreases from .5483 to .5361 over 424. Thus, the final diversity response distinguishes the two query directions even at the same displacement magnitude. Subsequent native updates reduce the much larger contrast in q_1, while retaining its ordering in final Patch-DINO.

C.4 Controlled ablations

The Dev decomposition and design controls, and the three-model skip comparison, use all 553 GenEval prompts. The additional cross-model placement and response tests use a twelve-prompt ablation cohort with GenEval indices 19, 168, 257, 300, 375, 460, 69, 97, 187, 312, 362, and 514. The first and last six each contain one prompt per category. These twelve-prompt tests use seeds 679000–679007 in the same order across arms; the cohort was also used to compare neighboring conditioning configurations. The ablation arms and contrasts were fixed before generating the additional outputs. The full-set studies specify their seed banks below.

C.4.1 Expanded Dev decomposition and design controls

We extend the Dev study to all 553 GenEval prompts using the same configurations as the twelve-prompt ablation. All arms retain seeds 679000–679007, eight outputs, 512^2 resolution, guidance 3.5, and the native 28-update schedule except for the explicitly omitted-prefix controls. The factorial applies one initial write, chosen from $h_0v^{(0)}$, $h_2v^{(0)}$, $h_0v^{(2)}$, or $h_2v^{(2)}$, then executes the native suffix from index 3. Each cell uses 26 network predictions. The norm controls use one additional prediction to obtain the reference magnitude. In contrast, the conditioning-only and half-write controls retain all 28 updates; halving the first write leaves its predicted clean-image estimate unchanged.

Table 18: **Full-set Dev ablations.** 553 prompts and eight matched seeds. The two groups retain either the full native schedule or the common suffix from index 3. Norm matching additionally evaluates the later-stage query to determine its scale. Late windows start at update 4.

Setting	Patch ↑	CLS ↑	Vendi ↑	CLIP ↑	HPS ↑	Pick ↑	MUSIQ ↑
Full native schedule							
Native	0.5658	0.3127	2.830	32.561	0.29702	23.402	74.122
One query t_2	0.6469	0.3750	3.212	32.348	0.29599	23.316	73.874
One query t_4	0.6654	0.3969	3.379	32.224	0.29540	23.229	74.203
One query t_8	0.6606	0.3895	3.344	32.112	0.29671	23.306	74.624
Two queries t_4, t_5	0.7112	0.4538	3.819	31.767	0.29103	22.847	74.329
Late one query ($d = 8$)	0.5790	0.3261	2.943	32.469	0.29721	23.380	74.581
Late two queries ($d = 4$)	0.5990	0.3471	3.100	32.252	0.29618	23.267	74.775
Half first write	0.5982	0.3284	2.936	32.449	0.29756	23.416	74.411
Common suffix from index 3							
$v^{(0)}, h_0$	0.6595	0.4000	3.453	32.023	0.29442	23.106	75.464
$v^{(0)}, h_2$	0.6541	0.3954	3.421	32.026	0.29457	23.120	75.414
$v^{(2)}, h_0$	0.6962	0.4350	3.675	31.858	0.29409	23.096	75.105
Skip 2: $v^{(2)}, h_2$	0.6946	0.4316	3.654	31.869	0.29425	23.107	75.090
Norm-matched $v^{(0)}, h_2$	0.6662	0.4083	3.509	31.919	0.29418	23.084	75.531

Changing the initial query increases Patch-DINO at either fixed integration interval, whereas changing the interval alone has a much smaller effect (Table 18). The reciprocal norm controls below further separate the prediction's direction from its magnitude.

Keeping every native interval and changing only the first query to t_2 increases variety more than halving the native first write. These controls distinguish changing the prediction from reducing the

influence of an unchanged prediction. At fixed $d = 4$, a second early query further increases variety at a greater preference cost. Both the one- and two-query interventions have a stronger effect at the beginning than when moved to update 4.

Table 19 gives the complementary twelve-prompt controls, including retention of the native first write followed by omission of updates 1 and 2. This control changes the subsequent trajectory without changing the first prediction.

Table 19: **First prediction versus its application.** Dev, twelve prompts and eight paired outputs. Halving the first write leaves its decoded estimate unchanged and retains all subsequent native updates. The omitted steps control retains the native first write but removes updates 1 and 2.

Control	Patch ↑	CLS ↑	Vendi ↑	CLIP ↑	HPS ↑	Pick ↑
Native	0.6022	0.4012	3.412	33.599	0.2957	23.406
Half first write; full suffix	0.6271	0.4207	3.542	32.992	0.2962	23.381
Native first write; omit steps 1/2	0.6865	0.4852	4.104	32.204	0.2935	23.002
First query t2; full schedule	0.6666	0.4250	3.626	32.986	0.2952	23.282

C.4.2 Velocity direction at matched magnitude

The reciprocal norm control completes a two-by-two comparison of velocity direction and magnitude. For each initial seed, we combine the unit direction of $v^{(0)}$ or $v^{(2)}$ with either prediction's norm. All four arms use the same interval h_2 and native suffix from index 3. The off-diagonal controls use one additional network call to obtain the reference norm.

Table 20: **Separating prediction direction and magnitude.** FLUX.1-dev, 553 prompts and eight paired seeds. Columns share the norm within each row; rows share the direction within each column group.

	Direction of $v^{(0)}$			Direction of $v^{(2)}$		
Magnitude	Patch ↑	HPS ↑	Pick ↑	Patch ↑	HPS ↑	Pick ↑
$\|v^{(0)}\|_2$	0.6541	0.29457	23.120	0.6917	0.29449	23.136
$\|v^{(2)}\|_2$	0.6662	0.29418	23.084	0.6946	0.29425	23.107

At either matched magnitude, the later-stage prediction direction yields greater Patch-DINO. Returning it to the native norm retains most of its gain; changing the norm alone has a smaller effect. Thus, changing the query affects prediction direction, not only update magnitude.

C.4.3 Skip implementations and cross-model controls

The full-set skip comparison uses seeds 679000–679007 for Dev and 229000–229007 for Schnell and Qwen, matched within each model to its conditioning-only outputs. All three models use independent single-image execution and retain eight outputs per prompt. Skip-s leaves the initial latent unchanged while bypassing the first s network predictions, then uses the original queries and integration intervals from index s. For Qwen, both CFG branches are bypassed together.

On Schnell, skipping substantially reduces MUSIQ, consistent with the degraded textures in the skip grids in Section G.5; conditioning-only generation instead improves it over Native. Dev's two rules also improve MUSIQ, while Qwen's score decreases slightly. The preference cost is therefore not simply a shared loss of technical image quality across models.

Official GenEval checks on these same outputs confirm the model-dependent skip response. On Schnell, correctness falls from 68.4% for Native to 4.5% and 0.3% after skipping one and two updates, respectively; conditioning only retains 58.3%. Thus the increased feature spread after one skipped update largely accompanies failed content generation. Qwen is less sensitive to removing updates: skip-one correctness remains close to Native, and skip-two produces a much smaller decline.

Table 21: **Complete skip and conditioning comparison.** All 553 prompts, eight paired outputs per prompt; all seven measurements use the same full sets.

Setting	Patch ↑	CLS ↑	Vendi ↑	CLIP ↑	HPS ↑	Pick ↑	MUSIQ ↑
FLUX.1-dev							
Native	0.5658	0.3127	2.830	32.561	0.2970	23.402	74.12
Skip 1	0.6457	0.3745	3.236	32.277	0.2970	23.350	74.53
Skip 2	0.6946	0.4316	3.654	31.869	0.2942	23.107	75.09
Ours ($d = 8, m = 1$)	0.6606	0.3895	3.344	32.112	0.2967	23.306	74.62
Ours ($d = 4, m = 2$)	0.7112	0.4538	3.819	31.767	0.2910	22.847	74.33
FLUX.1-schnell							
Native	0.6070	0.2962	2.774	32.892	0.2966	23.189	73.61
Skip 1	0.6774	0.6365	5.431	21.447	0.2149	17.487	60.65
Skip 2	0.2784	0.1002	1.530	20.528	0.1942	16.666	54.93
Ours ($\delta = .025, m = 1$)	0.6983	0.4181	3.586	31.931	0.2915	22.760	73.80
Qwen-Image							
Native	0.4065	0.1343	1.709	34.409	0.3010	23.641	75.20
Skip 1	0.4698	0.1636	1.862	34.464	0.3012	23.645	75.14
Skip 2	0.5419	0.2073	2.115	34.418	0.3000	23.585	74.89
Ours ($d = 3, m = 2$)	0.5718	0.2261	2.210	34.265	0.2996	23.510	74.96

To separate query and interval effects in the four-update sampler, the Schnell factorial crosses first queries 1.0 and $.975$ with intervals $-.250$ and $-.225$. All four cells retain the native suffix from index 1; the initial latent is neither rescaled nor re-noised. The nominal $.975$ query becomes $.9765625$ after the native BF16 conversion.

Table 22: **Schnell time-query and initial-interval factorial.** Twelve prompts, B8. Every cell makes four predictions and retains the same suffix after the first update. Query values are nominal, before BF16 conversion.

Query	Interval	Patch ↑	CLS ↑	Vendi ↑	CLIP ↑	HPS ↑	Pick ↑
1.0	-.250	0.6080	0.3460	3.097	33.504	0.2931	23.102
.975	-.250	0.7085	0.4877	4.071	32.049	0.2866	22.519
1.0	-.225	0.6232	0.3513	3.127	33.486	0.2929	23.053
.975	-.225	0.7169	0.4920	4.106	32.001	0.2875	22.544

The Schnell controls show the same ordering as Dev: changing the query increases variety at either interval, whereas shortening the interval alone has a much smaller effect. The query change also lowers CLIP and preference scores. Retaining all four updates provides direct control over this variety–preference response.

C.4.4 Intervention strength, duration, and placement

The following placement tests use the twelve-prompt ablation cohort. The Dev strength curve uses all 553 prompts, while Schnell and Qwen retain their twelve-prompt response studies. For a discrete index displacement d, a late window beginning at b uses $\tau_i = t_{i+d}$ for $b \le i < b + m$ and t_i elsewhere. Dev and Qwen use $b = 4$. Schnell uses $b = 1$ and constant normalized displacement $.025$. These controls match the window length and specified offset, not the norm of the realized velocity update. Duration comparisons fix the offset and compare $m = 1$ with $m = 2$ at the beginning of sampling.

C.4.5 Returning to native conditioning

Does the feature-diversity gain require jumping directly back to the native time condition? On 24 fixed Dev prompts with eight paired seeds (229000–229007), we compare the single-query sequence $t_8, t_1, t_2, \ldots$ with two three-query returns: $t_8, t_6, t_4, t_3, \ldots$ and $t_8, t_7, t_3, t_3, \ldots$. The latter two share the initial query, three-update window, and total index displacement, but distribute the return differently. Every arm retains the same 28 integration intervals and subsequent native conditions.

Table 23: **Duration and placement.** Twelve prompts and eight paired outputs. "Start" specifies the zero-based first intervened update; omitted start values mean step 0. All integration intervals and prediction counts remain native.

Model	Setting	Patch ↑	CLS ↑	Vendi ↑	CLIP ↑	HPS ↑	Pick ↑
FLUX.1-dev	Native	0.6022	0.4012	3.412	33.599	0.2957	23.406
FLUX.1-dev	$d = 4, m = 1$	0.6842	0.4629	3.833	32.834	0.2959	23.274
FLUX.1-dev	$d = 4, m = 2$	0.7272	0.5276	4.386	32.119	0.2921	22.916
FLUX.1-dev	$d = 8, m = 1$	0.6931	0.4842	4.019	32.329	0.2965	23.290
FLUX.1-dev	$d = 8, m = 1$, start 4	0.6216	0.4432	3.736	33.292	0.2967	23.360
FLUX.1-dev	$d = 4, m = 2$, start 4	0.6489	0.4692	3.985	32.698	0.2935	23.082
FLUX.1-schnell	Native	0.6080	0.3460	3.097	33.504	0.2931	23.102
FLUX.1-schnell	$\delta = .025, m = 1$	0.7085	0.4877	4.071	32.049	0.2866	22.519
FLUX.1-schnell	$\delta = .025, m = 2$	0.7150	0.5009	4.158	32.085	0.2870	22.494
FLUX.1-schnell	$\delta = .025, m = 1$, start 1	0.6218	0.3500	3.130	33.526	0.2930	23.052
Qwen-Image	Native	0.4263	0.1917	1.941	34.645	0.2996	23.722
Qwen-Image	$d = 3, m = 1$	0.5001	0.2183	2.092	34.605	0.2999	23.723
Qwen-Image	$d = 3, m = 2$	0.5561	0.2463	2.271	34.487	0.2989	23.657
Qwen-Image	$d = 3, m = 2$, start 4	0.4339	0.1990	1.990	34.765	0.2994	23.706

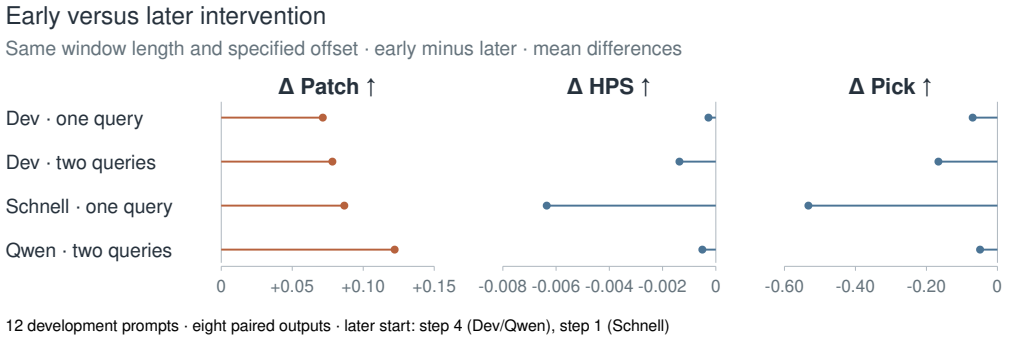

Figure 7: **Moving the intervention later.** Early-minus-late differences in the twelve-prompt means. Window length and specified offset are matched. All four early placements produce greater mean Patch-DINO; the associated preference changes depend on the model.

Table 24: **Returning to native conditioning.** FLUX.1-dev, 24 prompts and eight matched seeds. Prefixes list the changed queries; all later queries resume their native indices. All seven metrics use the same outputs.

Query prefix	Patch ↑	CLS ↑	Vendi ↑	CLIP ↑	HPS ↑	Pick ↑	MUSIQ ↑
Native	0.603	0.346	3.034	32.78	0.2939	23.22	73.40
t_8 only	0.697	0.407	3.507	32.08	0.2937	23.08	73.26
t_8, t_6, t_4	0.785	0.600	4.991	29.62	0.2787	21.72	73.35
t_8, t_7, t_3	0.787	0.608	5.115	29.36	0.2779	21.69	73.46

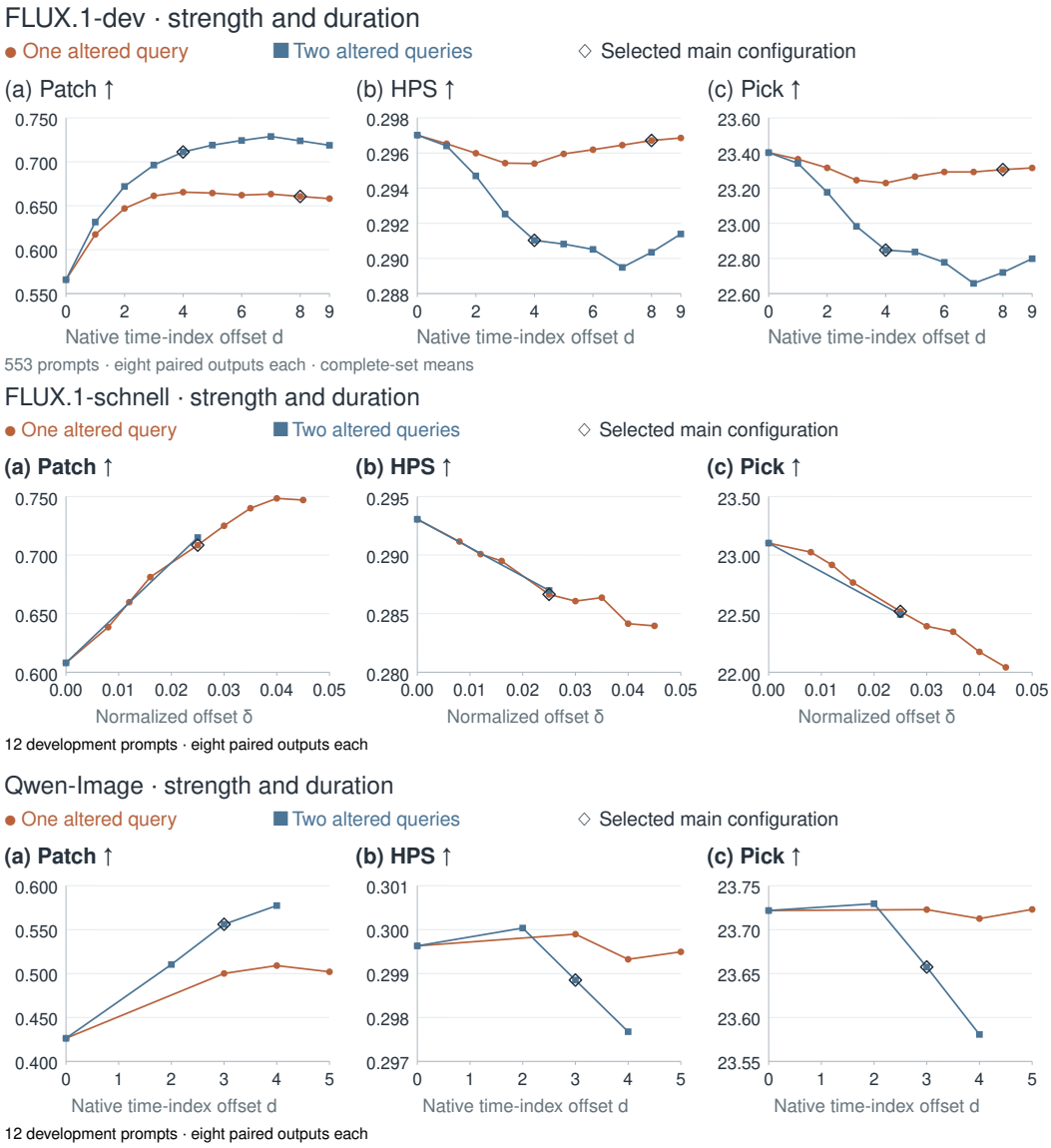

Figure 8: **Strength and duration across models.** Dev: all 553 prompts; Schnell and Qwen: twelve ablation prompts each. Every setting uses eight paired outputs per prompt. Lines connect measured means; diamonds mark the four selected configurations. The horizontal coordinate is an index displacement for Dev and Qwen and a normalized time displacement for Schnell. Metric scales are independently focused to show each model's response.

Both three-query profiles increase feature variety over Native, so a direct one-step return is not necessary for the gain. Their similar results indicate that the effect persists under different return shapes. Relative to the main single-query rule, however, both longer interventions produce greater feature spread at lower alignment and preference.

C.4.6 SKIP-TWO REPLICATION WITH BATCHED GENERATION

To test the skip observation with a different seed bank and batched execution, we skip the first two updates of FLUX.1-dev on all 553 GenEval prompts, with eight outputs per prompt and seeds 132000–132007. Resolution is 512^2, native guidance is 3.5, and the native schedule has 28 steps. Skip-two starts from the unchanged initial noise at native index 2 and executes 26 predictions. The matching Native and Skip-two sets were generated jointly in batches of eight.

Table 25: **Full-benchmark skip-two reference.** Paired changes across 553 prompts. This reference uses its own implementation-matched Native control.

Metric	Native	Skip 2	Difference
Patch	0.5693	0.6919	+0.1226
CLS	0.3062	0.4288	+0.1227
Vendi	2.784	3.625	+0.841
CLIP	32.563	31.905	-0.658
HPS	0.2971	0.2942	-0.0029
Pick	23.407	23.123	-0.284

Patch-DINO increases on 543 of 553 prompts, with a mean paired change of .1226. The gain appears in all six prompt categories. HPSv2, PickScore, and CLIP similarity decrease on average. The skip observation therefore also holds with this seed bank and batched execution, alongside alignment and preference costs.

D VARIETY AND IMAGE QUALITY

We first test feature-diversity gains among content-correct outputs and images retaining specified attributes, then report human and model-based visual assessments. The final analysis compares preference costs across query paths and composition categories.

D.1 PROMPT CORRECTNESS AND VALID-IMAGE DIVERSITY

The official GenEval evaluator (Ghosh et al., 2023) checks object identity, number, color, and spatial relations in the generated images. Table 26 reports the fraction of images it marks correct for each category. Native and Ours use the same eight seeds, 229000–229007, on each model. GI's candidate banks are listed in Table 6; GI is compared by prompt rather than by its selected seeds. Dev Native is the independent-sampling bank used in the main benchmark.

Table 26: **GenEval image-level correctness (%).** All 553 prompts, eight images per method. Columns follow the evaluator's six prompt categories; Pooled reports the pass rate over all images.

Method	Pooled	1 obj.	2 obj.	Count	Colors	Color attr.	Position
FLUX.1-dev							
Native	63.2	98.8	79.0	67.5	78.2	43.5	21.1
Group Inference	62.7	98.1	79.8	66.1	77.0	45.9	17.9
SPELL	60.4	95.3	70.7	71.6	75.3	41.5	18.5
TSR	63.7	99.4	79.4	66.7	79.9	44.5	20.9
CNO	61.3	95.8	76.9	68.6	73.5	43.0	19.2
Ours, one query	60.6	99.2	76.8	61.6	74.1	43.4	17.4
Ours, two queries	55.9	96.1	71.5	55.5	66.9	39.6	14.9
FLUX.1-schnell							
Native	68.4	99.2	90.7	65.9	79.0	51.4	30.8
Group Inference	67.4	98.8	87.6	65.5	78.9	52.5	28.1
SPELL	67.7	99.1	88.9	67.0	77.9	50.9	29.2
TSR	68.4	98.9	90.4	65.8	79.3	52.0	30.8
CNO	65.6	98.3	86.7	63.4	73.1	48.1	30.5
Ours	58.3	97.3	80.7	51.4	67.2	36.8	23.4
Qwen-Image							
Native	86.5	99.4	93.4	86.7	89.2	74.5	78.4
Group Inference	85.9	98.9	94.3	86.9	89.8	74.9	74.0
Guidance interval	84.7	98.9	94.1	82.0	89.4	73.4	73.1
SPELL	85.5	99.2	93.3	85.0	88.3	74.4	75.6
TSR	86.0	99.7	93.9	85.9	90.0	71.2	78.0
CNO	85.9	99.1	94.4	86.9	89.1	73.9	75.4
Ours	83.0	98.8	94.4	80.5	88.7	72.2	66.5

To measure feature diversity among content-correct outputs, Table 27 retains only prompts for which both methods have at least two correct images. At each such prompt, it compares the same number k of correct images on both sides, with $k = \min(n_{\text{ref}}, n_{\text{ours}})$; the table reports the number of eligible prompts. Pairwise Patch-DINO averaged over all correct pairs equals its expected value over uniformly selected k-subsets. The gains persist within this conditional comparison, although the difference from GI is small for Dev's one-query rule and Schnell. Both Dev rules, Schnell, and Qwen also retain greater Patch-DINO than TSR and CNO among the jointly eligible prompts.

Table 27: **Feature variety among GenEval-correct images.** Equal-count within-prompt Patch-DINO on the common eligible prompts. The correctness rates for all images, including ineligible prompts, are in Table 26. GI is matched by prompt, not by selected seed.

Model	Reference → Ours	Prompts	Ref. Patch	Ours Patch
Dev	Native → one query	415/553	.563	.655
	GI → one query	422/553	.648	.653
	Native → two queries	404/553	.560	.708
	GI → two queries	407/553	.646	.708
	TSR → one query	415/553	.573	.655
	CNO → one query	414/553	.600	.655
	TSR → two queries	399/553	.570	.708
	CNO → two queries	405/553	.595	.707
Schnell	Native → Ours	431/553	.599	.687
	GI → Ours	433/553	.680	.687
	TSR → Ours	433/553	.601	.686
	CNO → Ours	426/553	.639	.686
Qwen	Native → Ours	512/553	.402	.559
	GI → Ours	516/553	.456	.560
	Interval → Ours	516/553	.507	.559
	TSR → Ours	510/553	.414	.559
	CNO → Ours	516/553	.460	.560

GenEval checks requested content; Section G shows the corresponding types of visual change through complete image sets. Section D.4 separately evaluates visible defects and acceptable-image diversity using a blinded multimodal-model assessment.

D.2 PHOTOGRAPHIC APPEARANCE AND COMPOSITIONAL ATTRIBUTES

The GenEval-correct comparison in Section D.1 tests whether the feature-diversity gains remain among outputs satisfying the requested content. Here we additionally check photographic appearance and explicit compositional attributes on twelve constrained prompts per model. These specify attributes such as viewpoint, subject position, color, count, or an action. We apply the frozen time-query rules with seeds 881000–881007, paired between Native and ETC. Nonblind model-assisted inspection labels photographic appearance and the targeted attribute separately, retaining uncertain cases as a third category.

Table 28 keeps only seeds for which both Native and ETC pass both checks. Feature-diversity gains remain for every configuration, so they are not entirely explained by photographic-style changes or these attribute failures. Preference losses still occur for some configurations within the retained subset. The blinded assessment in Section D.4 further compares preference scores with judgments of faithfulness, visible defects, and aesthetics. Section D.5 examines how preference varies with the intervention path and with the compositions produced.

Table 28: **Paired images retaining style and the specified attribute.** ETC minus Native, averaged over eligible prompts with at least two surviving seed pairs. Prompts and Pairs report coverage out of twelve constrained prompts and 96 possible pairs per configuration; subsets differ between configurations.

Rule	Prompts	Pairs	ΔPatch	ΔHPS	ΔPick	ΔMUSIQ
Dev, one query	10/12	59	+0.074	+0.0002	+0.014	+0.51
Dev, two queries	7/12	33	+0.203	-0.0072	-0.593	+0.11
Schnell	10/12	55	+0.080	-0.0030	-0.120	-1.62
Qwen	11/12	78	+0.125	-0.0007	-0.092	+0.36

D.3 HUMAN SET-LEVEL EVALUATION

We compare perceived variety, visual quality, and prompt adherence using sets of eight FLUX.1-dev outputs from Native, GI64-to-8, and our single-query t_8 rule. The study complements the image-level model assessments in Appendix D.4 with direct human judgments of complete image sets.

Images and presentation. The evaluation contains 20 samples, with eight outputs per method. All participants view the same image sets. Native and Ours share initial seeds, while GI returns eight selected candidates from its population of 64. Rows are labeled A, B, and C rather than with method names. Participants can enlarge individual images.

Questions and aggregation. For each prompt, participants answer three separate questions: which group has greater variety, which has higher visual quality, and which follows the prompt better. They may select multiple groups as tied best, or choose "unable to judge." We received 12 complete responses. For each question and prompt, a method's top-choice rate is the fraction of non-abstaining participants selecting it, including ties. We then average these rates equally across the 20 prompts. The respective numbers of non-abstaining answers are 226, 210, and 200 out of 240 for variety, visual quality, and prompt adherence.

Participants favor ETC in perceived variety and visual quality on the evaluated image sets. Prompt adherence is close to Native.

Table 29: **Human evaluation on 20 prompts with 12 participants.** Top-choice rates include ties, so percentages need not sum to 100. Each prompt receives equal weight; "unable to judge" responses are excluded from its denominator.

	Top-choice rate (%) ↑		
Method	Variety	Visual quality	Prompt adherence
Native	6.0	33.9	**65.0**
GI	28.3	32.7	37.6
Ours (single query)	**86.6**	**70.3**	63.6

D.4 Model-based visual quality and accepted-image diversity

We examine whether individual images satisfy the prompt, contain visible defects, and retain diversity after unacceptable outputs are removed. This blinded multimodal-model assessment complements the set-level human evaluation above and is separate from the HPS and PickScore models used for optional selection (Section F.3).

D.4.1 Blinded assessment protocol

Image cohort. We uniformly sample 24 of the 553 Dev prompts without replacement, using random seed 2026092131 before collecting judgments. For each prompt, we evaluate eight outputs from Native, GI64-to-8, the unfiltered two-query t_4, t_5 rule, and its 12-to-8 joint-rank selection. Native and unfiltered Ours share seeds 229000–229007. The selected rule uses the first twelve candidates from the same conditioning-only bank. GI returns its selected eight outputs from seeds 229000–229063; its columns are not treated as seed-matched to independent generation. The cohort thus contains 96 sets and 768 method–image memberships. Shared images are judged once, giving 653 unique prompt–image pairs.

Judge and presentation. We use the `gpt-6-astra` model with medium reasoning effort in three fresh evaluation sessions. Images are randomly assigned to sessions and shown under anonymous identifiers in mixed-method 2×2 panels, with each original 512×512 image and its prompt separately identified. The judge has no method labels, scalar evaluation scores, or paper context, and assesses each image independently. Each unique image receives one judgment. The rubric, cohort, and analysis rules are fixed before inspecting the judgments.

Rubric. The judge records three separate attributes and a short explanation tied to visible evidence:

- **Prompt faithfulness:** yes, no, or unclear. All explicit objects, counts, colors, and spatial relations must be satisfied. An unspecified background, viewpoint, or setting is not a content error.
- **Visible defects:** none, minor, major, or unclear. This captures malformed objects and rendering artifacts separately from prompt correctness; an unusual but plausible view is not itself a defect.
- **Aesthetic quality:** an integer from 1 to 5, with 3 denoting ordinary acceptable quality, 4 good quality, and 5 excellent quality. Novelty and background complexity are not automatically rewarded.

An image is accepted if faithfulness is yes, defects are none or minor, and aesthetic quality is at least 3. An unclear faithfulness or defect judgment does not pass. We average within each prompt and then give all 24 prompts equal weight.

Consistency controls. We interleave eight repeated images and eight images paired with unrelated prompt text, giving 669 presentations in total. Controls do not contribute to method scores. All eight repetitions agree on faithfulness, aesthetic score, and acceptance; seven agree on defect severity. All eight prompt-mismatch controls are identified as incorrect, supporting repeat consistency and sensitivity to content mismatch.

D.4.2 QUALITY AND ACCEPTED-IMAGE DIVERSITY

Table 30 separates content fidelity from rendering quality. Major visual defects are rare in this cohort, whereas prompt faithfulness changes appreciably. Joint-rank selection raises the accepted fraction of the two-query outputs from 70.3% to 79.7%, approaching GI's 81.2%. Mean aesthetic quality also improves, although it remains below GI. Selection improves both acceptance and aesthetic quality.

Table 30: **Blinded model-based assessment on 24 randomly sampled Dev prompts.** Each method contributes 192 image memberships. Faithfulness, major defects, and acceptance are percentages; aesthetic quality is rated from 1 to 5. Shared outputs reuse the same judgment.

Method	Faithful ↑	Major defects ↓	Aesthetic ↑	Accepted ↑
Native	85.9	0.0	3.89	85.9
GI64-to-8	81.2	0.0	3.86	81.2
Ours: first 8	70.3	0.0	3.68	70.3
Ours: 12-to-8	79.7	0.5	3.76	79.7

Matching the number of accepted images. For prompt p, let $A_{p,m}$ be method m's accepted image set and set $k_p = \min_m |A_{p,m}|$. We evaluate prompts with $k_p \geq 2$, drawing k_p images uniformly without replacement from each method's accepted set. Metrics are averaged over 200 draws and then equally over prompts. For an unfiltered comparison, we draw the same k_p from each original eight-image set on exactly the same prompts. This prevents a difference in retained set size from driving the comparison, particularly Vendi. We reuse the original-space DINO pairwise matrices; filtering does not refit the image representation.

This leaves 23 of 24 prompts, with k_p between 2 and 8 (mean 5.65). The remaining prompt has no accepted output under any method and remains included in Table 30. Table 31 shows that the diversity advantage persists among accepted images. The 12-to-8 rule exceeds GI by 5.0% in Patch-DINO, 6.0% in CLS-DINO, and 5.2% in Vendi after filtering. The unfiltered two-query rule retains still greater diversity, but with a lower acceptance rate. Thus, its increased feature spread is not explained solely by incorrect outputs; quality selection provides a more balanced returned set.

Table 31: **Diversity before and after model-based visual filtering.** Both panels use the same 23 prompts and prompt-specific returned count k_p for every method. Accepted-only outputs satisfy the frozen visual rubric; unfiltered outputs are subsampled to the same counts.

Image pool	Method	Patch ↑	CLS ↑	Vendi ↑
Unfiltered	Native	0.5493	0.2854	2.2119
	GI64-to-8	0.6325	0.3648	2.5902
	Ours: first 8	0.6932	0.4564	2.9877
	Ours: 12-to-8	0.6667	0.3876	2.7271
Accepted only	Native	0.5437	0.2729	2.1906
	GI64-to-8	0.6263	0.3516	2.5595
	Ours: first 8	0.6671	0.4145	2.8649
	Ours: 12-to-8	0.6576	0.3727	2.6931

D.4.3 WHEN PREFERENCE SCORES AND VISUAL JUDGMENTS DISAGREE

We also compare Native and unfiltered Ours on the 192 shared prompt–seed pairs. We call the judged quality not lower when both images are accepted, Ours has at least the same aesthetic score, and its defect severity is no worse. This rule is fixed before examining the scores.

Lower preference scores do not consistently correspond to lower judged visual quality. Of the 145 pairs with lower HPS, 70 have no reduction under the visual-quality rule; the corresponding PickScore count is 68 of 143 (Table 32). Conversely, score decreases also occur on outputs that fail the rubric. The lower scores thus include both acceptable alternatives judged no worse and genuine content failures. Figure 9 illustrates accepted alternatives with lower preference scores. Section D.4.4 examines the framing changes within these disagreements.

Table 32: **Disagreement between scalar scores and model-based judgments.** Counts among 192 Native–Ours prompt–seed pairs. Columns after the first count refer only to pairs in which Ours has a lower scalar score. The "quality not lower" column is a subset of "both accepted."

Score	Score lower	Both accepted	Quality not lower	Ours not accepted
CLIP	109	72	56	36
HPS	145	94	70	47
Pick	143	92	68	48

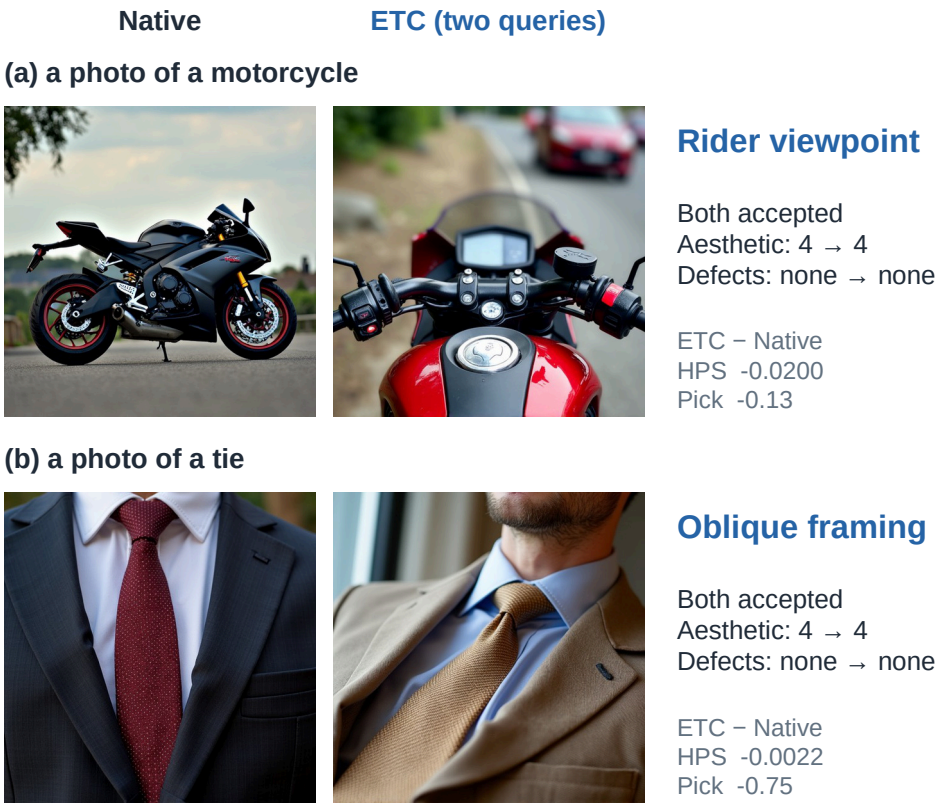

Figure 9: **Alternative viewpoints with lower preference scores.** Native and unfiltered ETC use the same seed within each row. (a) A complete motorcycle side view becomes a rider-view close-up. (b) A frontal tie view becomes an oblique view with more of the wearer visible. All four images pass the blinded model's visual rubric with aesthetic score 4, despite lower HPS and PickScore for ETC. These pairs are selected for visible viewpoint changes among judged-no-worse score disagreements. Aggregate disagreement counts appear in Table 32.

Reproduction details. The sampled prompt indices are 9, 18, 21, 28, 33, 38, 68, 75, 136, 158, 173, 182, 212, 220, 248, 258, 324, 392, 393, 399, 407, 462, 504, and 545. Prompt 407 is the sole prompt excluded from accepted-only diversity. Subset sampling uses random seed 2026092132. The saved evaluation record contains the full rubric, anonymous image manifests and hashes, per-image judgments and explanations, control responses, and aggregation code. The model identifier records the session model alias rather than a separately pinned checkpoint.

D.4.4 Reasonable framing changes and preference scores

The disagreements above include recognizable changes in how the subject is shown, not only variations in surface appearance. To examine this across the cohort, we separately annotate all 384 Native and unfiltered ETC images from the same 24 prompts, without selecting by score or quality outcome. Three fresh model sessions receive shuffled, anonymous images and their original prompts, but no method identities, preference scores, or previous quality judgments. The original quality judgments remain fixed.

The descriptive labels record whole versus frame-cropped subjects; wide, medium, or close framing; camera view; subject placement; and minimal versus recognizable scene context. Camera views distinguish frontal, side, oblique, overhead, low-angle, and first-person views, with other and unclear options. Missing objects are not automatically labeled as crops, and unrecognizable geometry remains unclear. We repeat 24 randomly selected images in different sessions: agreement is 24/24 for whole/partial framing and context, 22/24 for view, and 21/24 for scale and placement. Repeats do not enter the population statistics.

Reasonable alternatives with lower scores. The fixed quality-not-lower rule retains 100 of the 192 pairs. Among these, 71 pairs change at least one of whole/partial framing, subject scale, or camera view. HPS decreases in 46 of these pairs and PickScore in 49 (Table 33). Thus, score reductions can accompany changes in how a faithful, visually acceptable subject is presented, even without lower model-judged quality.

Table 33: **Framing changes among pairs judged no worse.** The 100 qualifying Native–ETC pairs come from the original random 24-prompt cohort. Lower columns count pairs with a score decrease; differences are ETC minus Native, averaged within each represented prompt and then equally across prompts. Labels describe framing categories. All 100 pairs have a determinate camera/framing label.

Camera/framing	Prompts	Pairs	HPS lower	Pick lower	ΔHPS	ΔPick
All qualifying pairs	23	100	70	68	-0.0049	-0.333
Camera/framing changed	22	71	46	49	-0.0049	-0.272
Same annotated camera/framing	14	29	24	19	-0.0084	-0.522

The directional breakdown in Table 34 makes the visual changes more concrete. Whole-subject views become partial views in 35 qualifying pairs, of which 26 lose HPS and 27 lose PickScore. Close views and different camera angles also occur among the score decreases. These observations extend the rider-view example in Figure 9 beyond a single selected image pair.

Table 34: **Direction of framing and context changes.** The same quality-not-lower subset and aggregation as Table 33. Rows can overlap because an image can change both scale and view. Whole/partial refers to the image boundary, not natural occlusion. All score directions, including increases, remain in the means.

Native to ETC	Prompts	Pairs	HPS lower	Pick lower	ΔHPS	ΔPick
Whole to partial subject	12	35	26	27	-0.0068	-0.365
Partial to whole subject	1	1	0	0	+0.0027	+0.095
Becomes a close view	12	21	14	15	-0.0065	-0.374
Leaves a close view	7	12	8	6	-0.0050	-0.194
Different camera view	15	35	21	26	-0.0038	-0.297
Minimal to scene context	4	10	7	5	-0.0023	-0.290
Scene to minimal context	4	4	4	4	-0.0167	-1.391

Interpreting the score changes. Figure 10 shows lower-scoring cake and donut close views, alongside higher-scoring changes in cup framing and tie presentation. Preference responses vary: the changed group contains 25 HPS increases and 22 PickScore increases, as well as the decreases reported above. Its mean loss is no larger than that of the unchanged-label group, also when paired within the 13 prompts containing both groups. The key finding is that lower scores can accompany reasonable alternative views with no reduction in model-judged quality. These disagreements co-exist with the content failures reported in Table 32; together, they show why preference scores and visual assessments provide complementary information.

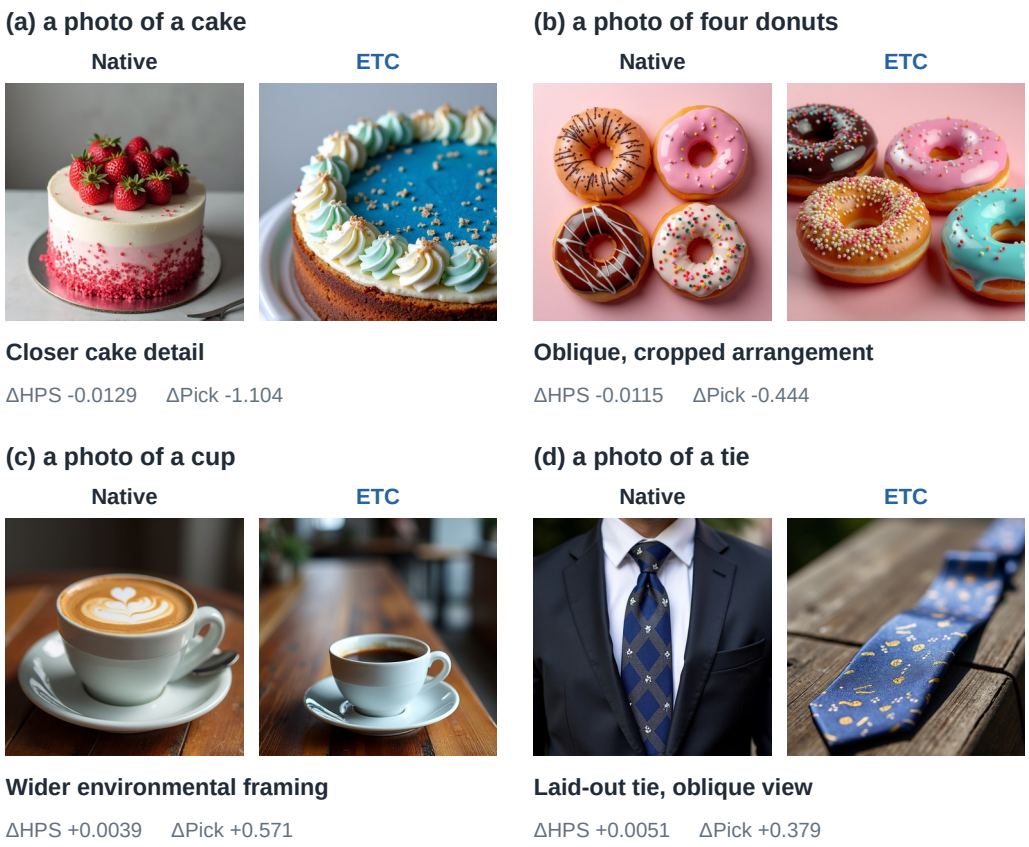

Figure 10: **Reasonable framing changes with different preference responses.** Each pair uses the same prompt and seed, with unfiltered Dev t_4, t_5 ETC. Both images pass the original blinded model rubric and ETC is judged no worse. Cake detail and an oblique donut arrangement receive lower HPS and PickScore; a wider cup scene and a laid-out tie receive higher scores. Examples illustrate both score directions. Images are shown uncropped; partial subjects are part of the generated outputs. Aggregate counts include all pairs in Tables 33–34.

D.4.5 Illustrative score responses on Qwen

Figure 11 examines the same umbrella prompt under Qwen's Native and fixed two-query rule. Native largely repeats a black umbrella against a plain background. The changed conditions produce oblique views, colored umbrellas, and outdoor scenes with people. Mean CLIP and PickScore decrease, while HPS increases. Even the two preference metrics respond differently to these changes, so the scores should be read alongside the visible compositions rather than as a single quality axis.

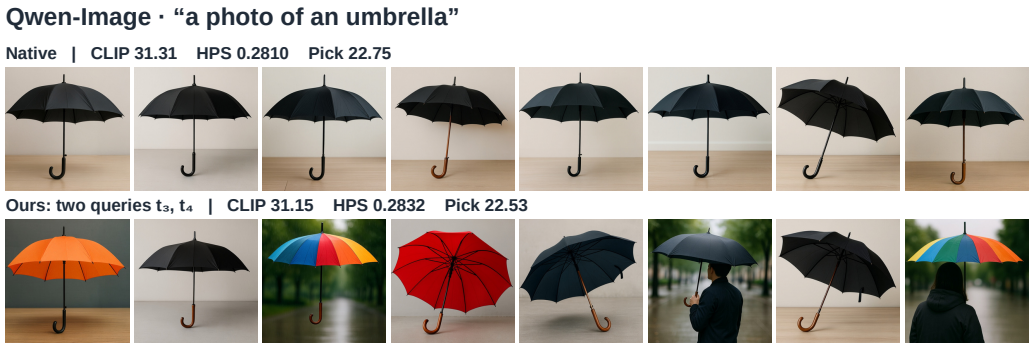

Figure 11: **Different scores, different visual changes.** Qwen-Image, “a photo of an umbrella.” Both rows retain all eight outputs in the same seed order. Ours changes only the first two conditions to t_3, t_4. Set-mean scores accompany the complete images: CLIP and PickScore fall, whereas HPS rises as the views, colors, and scene contexts become more varied.

D.5 Sources of preference differences

We compare preference costs across query paths, then separate changes in composition frequencies from changes in scores within those categories.

D.5.1 Query paths at similar feature diversity

Similar feature diversity can accompany different preference costs. We compare one Dev query at $d = 4$ with two queries at $d = 2$. Both use all 28 native integration intervals and seeds 679000–679007 per prompt.

Table 35: **Different paths at similar feature diversity.** All 553 GenEval prompts. Correctness is the official evaluator’s image-level pass rate. The two-query $d = 4$ rule provides a stronger-intervention reference.

Rule	Patch	HPS	Pick	MUSIQ	Correct (%)
One query, $d = 4$	0.665	0.2954	23.23	74.20	61.4
Two queries, $d = 2$	0.672	0.2947	23.18	73.86	61.7
Two queries, $d = 4$	0.711	0.2910	22.85	74.33	58.0

The one-query and two-query $d = 2$ rules have close Patch-DINO and correctness rates, but the one-query rule has higher HPS, PickScore, and MUSIQ (Table 35). Among the 422 prompts with at least two correct images in both arms, their valid-image Patch-DINO remains close at .651 and .656. The path therefore matters beyond the amount of feature spread. To check whether this comparison extends beyond GenEval, we keep both rules fixed and evaluate long-form descriptions.

We first compare the rules on 24 fixed random descriptions from the 96-prompt long-form set, then evaluate the remaining 72 without retuning. All use seeds 229000–229007 paired across rules. Table 36 separates the confirmation set from the pilot and pooled results. On the 72 confirmation prompts, Patch-DINO is nearly identical, while the one-query rule retains higher mean HPS, PickScore, and MUSIQ. HPS is higher on 55 prompts and PickScore on 45. This confirms the preference ordering beyond the initial descriptions.

Inspection of all 24 pilot pairs of eight-image sets shows substantial overlap in scene types, alongside changes in viewpoint and content accuracy. For example, the two-query rule corrects the requested golf-ball count for one seed but breaks it for another; it also introduces full-body views in a raccoon scene. The path change can therefore affect both the alternatives represented and how accurately they are rendered. Preference-based output selection provides a separate way to improve returned-set scores without changing the query path (Section F.3).

Table 36: **Path comparison on natural descriptions.** Eight paired outputs per rule, with the native 28-update schedule. The remaining 72 descriptions confirm the frozen comparison after the 24-prompt pilot; All pools both sets without changing the rules.

Set	Rule	Patch	HPS	Pick	MUSIQ
Pilot (24)	One query, $d = 4$	0.604	0.2957	22.29	73.17
Pilot (24)	Two queries, $d = 2$	0.607	0.2942	22.19	72.76
Confirmation (72)	One query, $d = 4$	0.602	0.2911	22.22	73.70
Confirmation (72)	Two queries, $d = 2$	0.603	0.2897	22.18	73.05
All (96)	One query, $d = 4$	0.603	0.2922	22.24	73.57
All (96)	Two queries, $d = 2$	0.604	0.2908	22.18	72.98

D.5.2 Composition frequency and within-class preference

Do preference differences reflect which compositions appear, or scores within those categories? We inspect complete Native and single-query t_8 ETC pools for 20 fixed prompts: eight randomly selected prompts and twelve preference-loss cases. We use seeds 770000–770127 paired across arms (5,120 images). Nonblind model-assisted inspection records prompt-specific scene or view categories, explicit content/style failures, and uncertain cases before joining preference scores. The categories complement the official content checks and the blinded model-based assessment in Section D.4.

For each of the 18 prompts where both methods have valid images in shared categories, let p_k be a category's frequency and R_k its mean score. Subscripts E and N denote ETC and Native. Frequencies are normalized within that shared category set for each method. The score difference admits the exact decomposition

$$\overline{R}_E - \overline{R}_N = \sum_k (p_{E,k} - p_{N,k}) \frac{R_{E,k} + R_{N,k}}{2} + \sum_k \frac{p_{E,k} + p_{N,k}}{2} (R_{E,k} - R_{N,k}). \tag{6}$$

The first term measures the contribution of changed category frequencies; the second measures changed scores within the shared categories. These categories group related compositions rather than holding composition fixed.

Table 37: **Frequency and within-category contributions.** ETC minus Native, averaged equally over the 18 eligible prompts. This table concerns shared valid categories; failures and categories observed in only one arm remain in the complete-pool analysis.

Metric	Frequency	Within class	Total
HPS	+0.00007	-0.00012	-0.00006
Pick	+0.019	-0.073	-0.054
MUSIQ	+0.000	+0.260	+0.260

For PickScore, the positive frequency contribution is outweighed by a negative within-category contribution. A change in category frequencies alone therefore does not explain the observed loss. Responses differ by prompt: the shared suitcase categories improve, whereas the three-truck categories lose preference with little change in their relative frequencies. MUSIQ can increase while preference falls, further separating technical image quality from prompt-conditioned preference. Section D.4.4 examines a complementary question: whether preference scores can fall for alternative views judged visually no worse, rather than how category frequencies contribute to the mean.

The Native pools already contain some of the target alternatives, including full-body bears and suitcases in rooms, with high scores under more than one quality proxy. These larger pools demonstrate that Native can produce high-scoring examples of some ETC alternatives. Other target categories have lower preference, highlighting the role of both sampling frequency and quality within each category.

E COMPUTATIONAL COST AND DEPLOYMENT

To distinguish batching from sampling work, a denoiser sample-evaluation counts one latent processed by one network evaluation. For a returned set, the total is the sum of batch sizes over all denoiser calls, with explicit conditional and unconditional branches counted separately. Generating eight images independently therefore costs $8T$ sample-evaluations for Dev and Schnell and $16T$ for the Qwen implementation. Changing time inputs adds no calls or backward passes. Group Inference's total instead depends on its candidate population at each update; its 64 initial candidates are not 64 completed trajectories.

The matched GI runs process populations $64, 32, 16, 8, \ldots, 8$ on Dev and Qwen, and $64, 8, 8, 8$ on Schnell. Summing over 28, 30, and four updates, respectively, gives 312, 656, and 88 sample-evaluations, including both Qwen guidance branches. These totals agree with the recorded network calls in all completed confirmation runs. They exclude GI's intermediate decoding, feature scoring, and discrete selection, which add work outside the denoiser.

Implications for large-scale deployment. ETC preserves not only Native's denoiser work but also its independent-task structure. In the work–depth model of parallel computation, total work and the longest dependency chain jointly determine available parallelism (Blelloch, 1996). At the trajectory level, generating N images with T updates entails NT denoising stages arranged into N independent chains of length T for both Native and ETC; each stage retains the model's native guidance evaluations. ETC changes time-input values, not these dependencies or the network's tensor shapes. Thus, with P identical model replicas, equal per-image service time ℓ, and negligible dispatch overhead, both admit the same ideal completion time $\lceil N/P \rceil \ell$. This is output-level parallelism: a replica may itself span multiple devices. Independent prompt/seed tasks can be assigned to available replicas without exchanging intermediate sample states, following the distributed-inference pattern supported by Diffusers (Hugging Face, n.d.). At fixed local batch size, ETC requires neither an additional model nor group-level feature storage; increasing the total output count therefore does not introduce a new per-worker inference-memory requirement. In contrast, GI requires additional candidates and group-wise scoring and selection (Parmar et al., 2026), SPELL requires interactions among concurrent samples (Kirchhof et al., 2025), and CNO requires a noise-optimization stage before sampling (Kim et al., 2026). These operations are integral to their diversity mechanisms: removing them to recover Native's independent sampling workflow also removes the corresponding intervention. ETC obtains its variety gains without imposing any of these additional execution requirements. The unchanged peak memory and near-native latency measured below support this structural analysis. ETC therefore offers a variety control that fits existing generation pipelines without adding candidate management, an external scorer, or cross-sample synchronization. Optional preference-based selection adds a separate stage, whose cost is reported in Section F.3.

Timing protocol. We use an RTX A6000 (48 GB), BF16, GenEval prompt 53, and seeds 339000–339007 for each eight-output set. Native and the four evaluated rules generate independently at batch one, with resident denoiser and VAE and CPU-resident text encoders. Diffusion prompt embeddings are cached. Timing includes noise and schedule construction, denoising, native decoding, and pixel conversion; loading, diagnostic traces and file writes are excluded. After exact cached-output replay and warm-up, we synchronize CUDA around each timed set and rotate the execution order over three repetitions. Table 38 reports mean latency and maximum peak allocated memory, including all resident models and prompt tensors. Table 40 repeats these rules on a second prompt and seed bank.

Selection and batch-based execution. GI extends the same seed bank to 64 candidates, with batch-one denoiser calls. Its timing additionally includes candidate decoding, CLIP/DINO scoring and scoring-text encoding, candidate management, and solver startup and selection. Reporting-only scores and a redundant tiny-decoder final output are omitted. Peak memory includes the selection networks and the additional tiny VAE used for FLUX selection.

Qwen guidance interval retains both CFG predictions at batch one, using the same cached-prompt boundary and three alternating repetitions.

The conditioning-only rules retain Native's denoiser work and peak memory, with near-identical latency. GI processes additional candidates and adds scoring and selection. The main comparison

reports paired runtime ratios; absolute latency and the implementation-specific controls are given below.

Table 38: **Denoiser work and runtime comparison.** Eight native-decoded outputs at 512^2 on an RTX A6000, with cached diffusion prompts. Sample-evaluations count processed latents and CFG branches; seconds/set include required decoding, transfers, and selection. Guidance interval retains its matched Native control. Peak allocated memory includes all required resident models. None of these samplers uses backward evaluations.

Method	Initial pool	Sample-evals	Seconds/set	Peak GiB
FLUX.1-dev				
Native	8	224	72.12	22.94
Ours ($d = 8, m = 1$)	8	224	72.53	22.94
Ours ($d = 4, m = 2$)	8	224	72.69	22.94
Group Inference	64	312	107.91	24.25
FLUX.1-schnell				
Native	8	32	10.35	22.92
Ours ($\delta = .025, m = 1$)	8	32	10.35	22.92
Group Inference	64	88	35.07	24.23
Qwen-Image				
Native	8	480	124.54	39.34
Ours ($d = 3, m = 2$)	8	480	125.20	39.34
Group Inference	64	656	188.07	40.41
Native (interval B1)	8	480	126.90	39.34
Guidance interval (B1)	8	480	128.22	39.34

Other baseline costs. Table 39 uses seeds 229000–229007, cached text embeddings, native decoding, and three repetitions after warm-up on the same RTX A6000. SPELL uses prompt 19; TSR and CNO use prompt 53. Loading, warm-up, and image-file writes are excluded.

SPELL evaluates each eight-image group at microbatch one. Its Native control disables repulsion within the same instrumented sampling path; both retain identical diagnostics. TSR and CNO instead exclude diagnostics and rotate their execution order with Native. CNO timing includes initial-noise optimization, required VAE transfers, and final native sampling; its Native control omits optimization. The learning-rate .02 settings on Dev and Qwen each use their own matched Native measurement.

SPELL and TSR have near-native latency. SPELL nevertheless accesses the group's states for repulsion, unlike independent sampling; CNO adds the cost of noise optimization.

Table 39: **Matched baseline runtime.** Mean seconds per eight outputs at 512^2 on an RTX A6000. Each row uses its own implementation-matched Native control. CNO follows the released gradient path; both learning rates in Tables 8 and 9 are timed separately.

Model / setting	Native (s)	Method (s)	Ratio
SPELL			
FLUX.1-dev	72.43	72.43	1.00
FLUX.1-schnell	10.59	10.61	1.00
Qwen-Image	127.60	127.52	1.00
TSR			
FLUX.1-dev	72.18	72.22	1.00
FLUX.1-schnell	10.84	10.85	1.00
Qwen-Image	126.30	126.50	1.00
CNO			
FLUX.1-dev (lr=.01)	72.18	104.54	1.45
FLUX.1-dev (lr=.02)	72.26	104.78	1.45
FLUX.1-schnell	10.69	43.09	4.03
Qwen-Image (lr=.01)	126.30	179.60	1.42
Qwen-Image (lr=.02)	126.23	179.91	1.43

Main-configuration runtime. The following measurements repeat the four main configurations on prompt 19 and seeds 679000–679007, separately from the prompt 53 comparison. Each set contains eight sequential batch-one outputs with cached text embeddings and native decoding. The device is an RTX A6000. Three repetitions alternate the order of Native and intervention settings; each setting is replayed against its stored images before timing. Peak allocated memory includes the resident denoiser and decoder. The ratio uses the Native measurement from the same experiment.

Table 40: **Main-configuration cost on a second workload.** Predictions are per image; latency is per eight-output set. Loading, warm-up, diagnostic traces and file writes are excluded.

Model	Setting	Predictions	Seconds / set	Ratio	Peak GiB
FLUX.1-dev	Native	28	71.58	1.000	22.94
FLUX.1-dev	Ours ($d = 8, m = 1$)	28	71.66	1.001	22.94
FLUX.1-dev	Ours ($d = 4, m = 2$)	28	71.73	1.002	22.94
FLUX.1-schnell	Native	4	10.50	1.000	22.92
FLUX.1-schnell	Ours ($\delta = .025, m = 1$)	4	10.51	1.001	22.92
Qwen-Image	Native	60	124.46	1.000	39.34
Qwen-Image	Ours ($d = 3, m = 2$)	60	124.64	1.001	39.34

F CANDIDATE BUDGETS AND OPTIONAL QUALITY SELECTION

Time conditioning changes the candidates a generator produces; selection changes which candidates are retained. These operations are complementary. We first compare ETC with GI's progressive pruning and external CLIP/DINO scoring (Parmar et al., 2026) as the returned-set size grows. We then apply HPS/Pick selection to completed ETC outputs, testing whether their diversity advantage remains at preference scores and serving cost comparable to GI.

F.1 RETURNED-SET SIZE AT FIXED AND PROPORTIONAL CANDIDATE BUDGETS

Fixed candidate pool on Dev. On six GenEval prompts (53, 113, 193, 345, 412, 531), we vary the returned set size K from 8 to 64, holding GI's initial population at $M = 64$. Native and both conditioning-only rules use the first K independent seeds from the same 64-seed bank, beginning at 339000. GI is run for each requested K; a larger returned set is not obtained by subsampling a different GI run. The released Dev GI implementation uses guidance 1 for its first prediction and 3.5 thereafter. Consequently, its $M = K$ limit is an unselected GI backbone, not Native with guidance 3.5 throughout.

Table 41: **Dev returned-set size with a fixed GI pool.** Mean Patch-DINO over six prompts. All methods return the same K within each column. The unselected backbone preserves GI's first-step guidance policy.

Returned images K	8	16	32	64
Native	.5002	.5180	.5202	.5275
Ours: one query t_8	.6406	.6280	.6313	.6251
Ours: two queries t_4, t_5	.6584	.6532	.6668	.6684
GI, $M = 64$	.6170	.5915	.5773	.5782
GI's unselected backbone	.5861	.5820	.5744	.5782

GI's selection gain over its own backbone shrinks as a larger fraction of the fixed pool must be returned; at $K = M$, no selection remains. Both conditioning-only rules retain a Patch-DINO gain over Native as K grows. Increasing GI's pool to 128 for $K = 16$ instead raises Patch-DINO from .5915 to .6295. The relevant constraint is therefore the available selection budget, not the returned-set size alone.

Proportional candidate budget on Schnell. We therefore also hold $M/K = 8$ rather than M fixed. On six development prompts (19, 168, 257, 300, 375, 460), seeds begin at 679000 and the conditioning-only rule sets the first time input to .975. The completed comparisons at $K = 8$ and

$K = 16$ are shown in Table 42. GI retains its released .9 pruning ratio: it evaluates $8K$ candidates once and K on each of the remaining three updates. Its denoiser work is therefore $11K$ sample-evaluations, versus $4K$ for independent generation, in addition to its decoding, scoring, and selection operations. An eightfold candidate population does not mean eightfold full-generation work.

Table 42: **Schnell with a fixed eightfold GI candidate ratio.** Six-prompt means; no quality-based output filtering is applied to Ours. The candidate ratio is fixed, unlike Table 41.

K	Method	Candidates	Patch ↑	HPS ↑	Pick ↑
8	Native	8	.6653	.29002	22.850
8	Ours	8	.7336	.28549	22.394
8	GI	64	.7193	.28924	22.767
16	Native	16	.6543	.29172	22.884
16	Ours	16	.7274	.28661	22.462
16	GI	128	.7168	.28873	22.767

With the candidate ratio held fixed, GI retains its advantage over Native as the returned set grows, while our method remains higher in Patch-DINO with lower preference scores. This differs from the decreasing GI selection gain at a fixed candidate budget: returning more images does not by itself eliminate the benefit of selection.

F.2 Output distributions at matched return count

We compare 64 returned images per method on the six Dev prompts in Table 41. Native and the two query rules reuse their complete independent 64-image sets. For GI, we pool eight independent 64-to-8 selections from disjoint candidate banks. This matches the number of returned images; GI processes 512 initial candidates per prompt. Each image is represented by its unit-normalized DINOv2-base CLS feature. We fit one PCA jointly to all four methods within each prompt and display its first two coordinates in Figure 13.

The two-query rule has the largest mean CLS distance and Vendi score, while GI and the single-query rule have similar mean CLS distances (Table 43). The projected sets overlap rather than forming disjoint method-specific clusters. Repeated small-group selection retains a feature-diversity gain even when its outputs are pooled into a larger set. Figure 12 connects the point representation to actual images. For the white-bottle-and-blue-sheep prompt, the ETC representatives vary the sheep's pose, its scale relative to the bottle, and the placement of the two subjects. These image-space differences make the feature spread interpretable alongside changes in appearance and scene context.

Table 43: **Matching 64 returned images.** Six-prompt means. GI pools eight independent 64-to-8 groups; the other methods generate 64 independent outputs.

Setting	Patch ↑	CLS ↑	Vendi ↑	CLIP ↑	HPS ↑	Pick ↑
Native	0.5275	0.2466	3.705	33.063	0.30278	23.852
Ours: one query t_8	0.6251	0.3096	4.902	33.012	0.30181	23.724
Ours: two queries t_4, t_5	0.6685	0.3669	6.255	32.759	0.29686	23.325
GI: pooled selections	0.6290	0.3098	4.772	33.047	0.30115	23.754

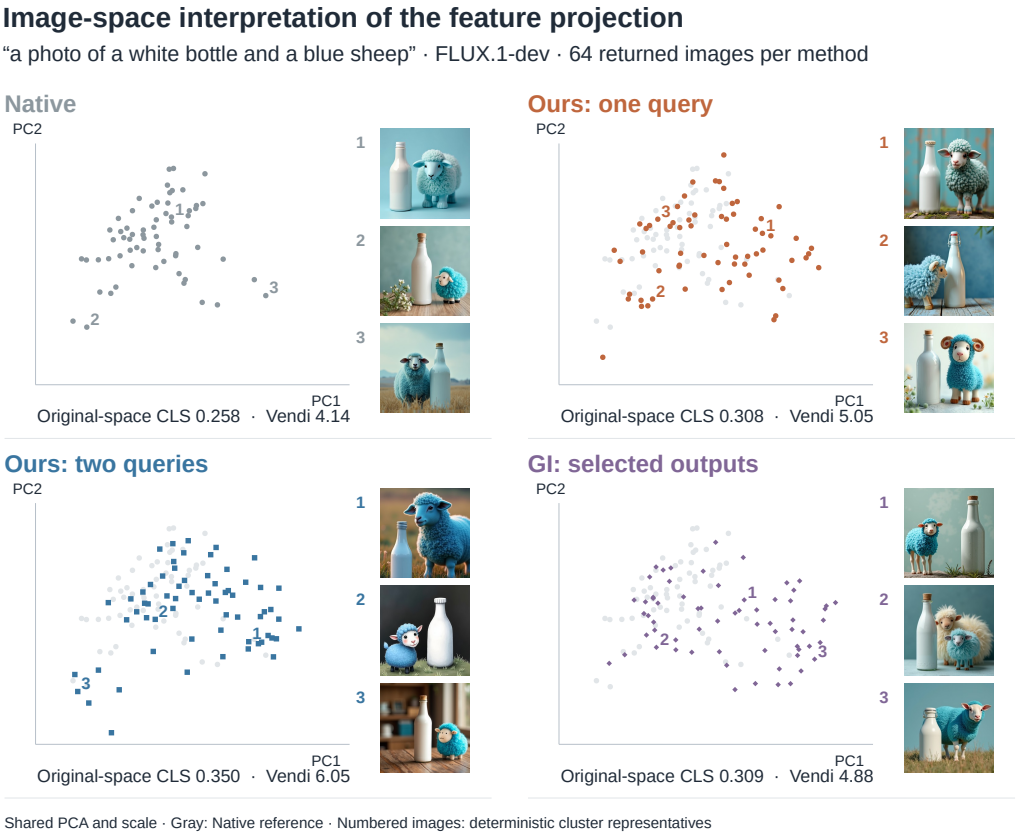

Figure 12: **Connecting feature points to images.** One Dev prompt, 64 returned images per method. All panels use the same PCA coordinates and scale; gray points repeat the Native reference. Numbered images are three deterministically selected feature-cluster representatives within each method’s full set. GI pools eight independent 64-to-8 selections. Original-space CLS and Vendi accompany the projection. The common view reveals overlap and changes in appearance, arrangement, and content.

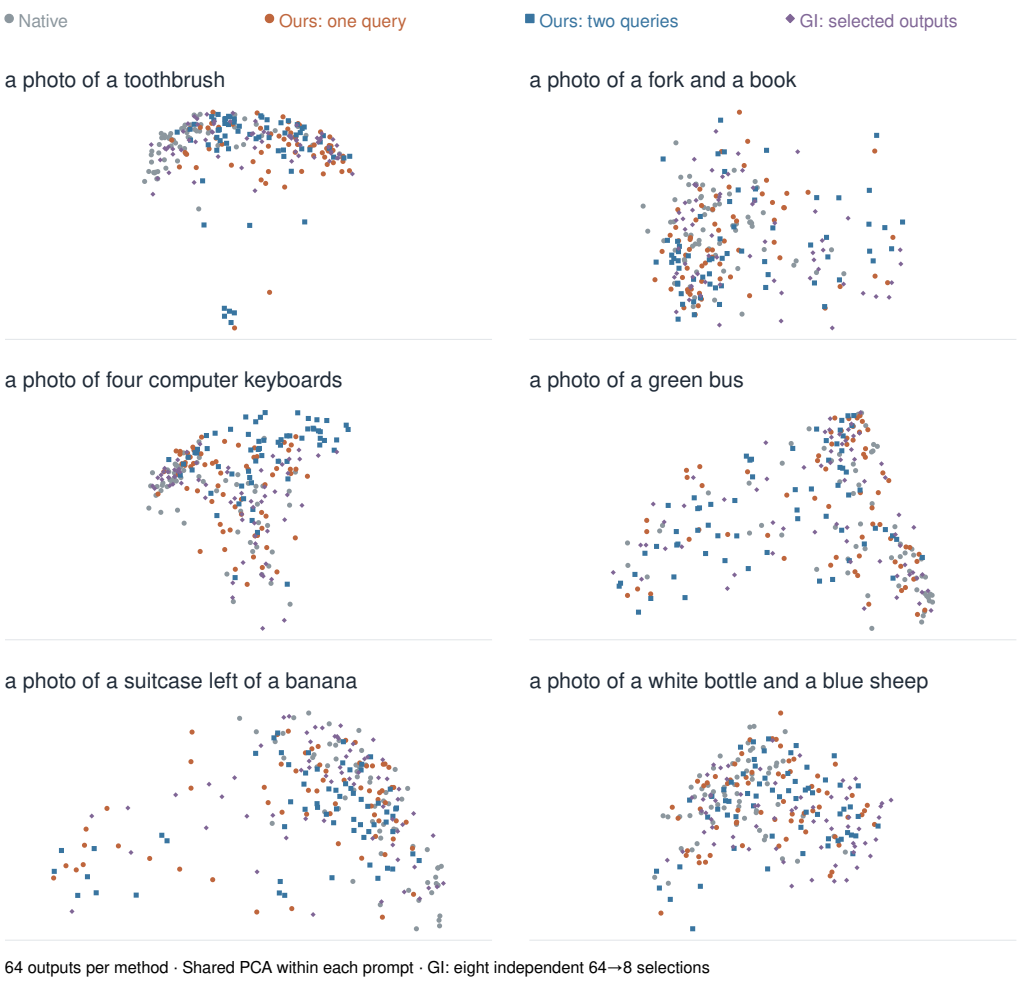

Figure 13: **Each image as a point.** All six Dev scaling prompts, with 64 outputs per method. Native and both query rules share a seed bank; GI uses eight disjoint candidate banks. Each panel uses a shared PCA fit to all four output sets; all points are retained. The two displayed coordinates preserve 28–37% of feature variance. These are global appearance features, while the image sets show which differences involve composition or content. Each prompt has its own projection axes.

F.3 VARIETY AT COMPARABLE PREFERENCE AND COMPUTATIONAL COST

We test whether the diversity advantage persists when output preference and serving cost approach those of GI. On all 553 Dev prompts, selecting eight of twelve conditioning-only outputs retains higher Patch-DINO, CLS-DINO, and Vendi with similar HPS, PickScore, and runtime (Table 44). ETC first produces a more varied pool; preference-based selection then raises its returned-set scores without eliminating the diversity advantage. We vary only the candidate budget of a fixed selection rule, not the underlying ETC configuration.

For each prompt, we use a prefix of $M \in \{10, 11, 12, 16\}$ independent candidates generated by the unchanged two-query t_4, t_5 rule. Let $r_H(i)$ and $r_P(i)$ be the descending-score ranks of candidate i under HPS and PickScore, respectively, with average ranks assigned to ties. We return the eight candidates with the smallest $r_H(i) + r_P(i)$. Equal combined ranks are resolved by higher HPS, then higher PickScore, then the initial seed order. Patch-DINO does not enter the selection rule. This is an optional output-selection wrapper, not a change to the time-conditioning intervention. The ranks are recomputed within the M available candidates, so the smaller budgets never access the remaining images in the stored 16-image pool.

Preference and cost near GI. The full-set comparison uses seeds 229000–229015 and the compatible Native8 reference. Each candidate budget accesses only its own prefix of this bank. For cost, we generate each pool afresh on an RTX A6000 alongside Native8 and GI64-to-8, using five cyclically ordered repetitions of prompt 53 with seeds beginning at 339000. Timings include generation, native final decoding, and required scoring and selection, but exclude loading, warm-up, and file writes. Preference scoring retains batch size 16 by repeating the last in-pool image for padding, then discarding the padded scores; no out-of-pool candidate is read, and padding cost is included.

Table 44: **Variety at comparable preference and runtime.** Quality and diversity are means over 553 prompts; time is the mean of five same-GPU repetitions on one fixed prompt, normalized to GI. All methods return eight images; Ours selects by HPS/Pick ranks, without using diversity scores.

	Diversity ↑			Alignment / preference ↑			Cost ↓
Setting	Patch	CLS	Vendi	CLIP	HPS	Pick	Time / GI
Native8	0.5770	0.3132	2.837	32.407	0.29731	23.390	0.670
GI64-to-8	0.6617	0.3840	3.269	32.182	0.29691	23.336	1.000
Ours: 10-to-8	0.7071	0.4268	3.649	31.922	0.29471	23.096	0.844
Ours: 11-to-8	0.7013	0.4160	3.573	32.046	0.29586	23.180	0.927
Ours: 12-to-8	0.6968	0.4070	3.516	32.077	0.29673	23.240	1.011

Twelve candidates give the closest comparison to GI in both preference and runtime. HPS and PickScore approach GI's values, while Patch-DINO, CLS-DINO, and Vendi remain 5.3%, 6.0%, and 7.6% higher, respectively, at a runtime difference of 1.1%. Eleven candidates offer a lower-cost point, with slightly lower preference and 7.3% less runtime than GI. Increasing the pool to sixteen recovers Native-level preference at greater cost.

GI prunes partial trajectories; our wrapper completes a smaller pool before selection. Twelve candidates require 336 denoiser sample-evaluations compared with GI's 312, while their measured end-to-end runtimes are close. The additional computation belongs to the optional selection wrapper; the conditioning intervention itself adds no model evaluations.

Selection criteria and independent quality checks. Using the same 16-image pools (8,848 images in total), we compare joint ranking with HPS-only, Pick-only, and no-selection rules on all 553 prompts. HPS and PickScore are optimized selection criteria here, rather than independent quality judgments.

Both joint and Pick-only selection exceed Native in the two preference means while remaining above GI in Patch-DINO (Table 45); joint ranking favors HPS at a small PickScore cost. MUSIQ, which is not used for selection, also improves relative to the first eight outputs and slightly exceeds Native under joint selection (Table 46). Section D.4 adds blinded model-based judgments and diversity measured only among accepted images.

Table 45: **Full 553-prompt quality-selection study.** All Ours rows use Dev's two-query t_4, t_5 configuration and the same 16-image pools; each returns eight images. HPS and PickScore participate in the selection rules and are reported as optimized preference scores.

Setting	Patch ↑	CLS ↑	Vendi ↑	CLIP ↑	HPS ↑	Pick ↑
Native8	.5770	.3132	2.8371	32.407	.29731	23.390
GI64-to-8	.6617	.3840	3.2688	32.182	.29691	23.336
Ours: first 8	.7237	.4704	3.9446	31.326	.28988	22.685
Ours: HPS top 8 of 16	.6917	.3912	3.4070	32.283	.29959	23.363
Ours: Pick top 8 of 16	.6904	.3872	3.3729	32.339	.29813	23.476
Ours: joint rank top 8 of 16	.6904	.3876	3.3740	32.345	.29922	23.439

Table 46: **Technical quality after preference-based selection.** All 553 prompts, eight returned outputs each. MUSIQ is not used by either selection criterion.

Setting	MUSIQ ↑	Difference from Native
Native8	73.776	–
Ours: first 8	73.541	-0.235
Ours: HPS selection	74.157	$+0.382$
Ours: Pick selection	73.810	$+0.035$
Ours: joint selection	73.987	$+0.211$

End-to-end selection cost. Generating 16 Dev candidates requires 448 denoiser sample-evaluations, compared with 224 for eight independent outputs and 312 for matched GI64-to-8. We time these workloads on an RTX A6000 with prompt 53 and seeds beginning at 339000, using three counterbalanced warm-resident repetitions. Timings include native final decoding and each method's required scoring and selection, but exclude loading, warm-up, and file writes.

Table 47: **Measured cost of the optional selection wrapper.** Eight returned images per method. GI progressively prunes its initial 64 candidates rather than completing 64 independent trajectories.

Setting	Denoiser sample-evaluations	Seconds/set
Native8	224	71.46
Ours: joint 16-to-8	448	143.73
GI64-to-8	312	107.03

The 16-to-8 wrapper takes $1.343\times$ GI's runtime on this workload. Quality scoring and selection take approximately .57 seconds; generating the 16 complete candidates dominates its cost. The conditioning-only intervention and its optional output-selection wrapper therefore have different computational profiles.

Development comparison. The selection rule was developed on the same six Dev prompts and seed bank as Table 41, using the first 16 candidates from each pool. We compare selected-set HPS and PickScore against unfiltered Native8 and Patch-DINO against GI64-to-8. These targets apply to prompt-averaged scores, not separate per-image acceptance thresholds.

Joint selection after two queries exceeds Native's mean HPS and PickScore while retaining higher Patch-DINO than GI. Applying the same selection to one query raises both preference scores, but its Patch-DINO falls below GI. The two-query rule therefore provides more diversity for selection to consume while retaining the target advantage.

Table 48: **Selection-rule ablation on Dev.** Six development prompts; all sets contain eight returned images. The three quality-selection rules operate on the same 16 candidates for each generator. Native and GI are not subjected to an additional quality filter.

Generator	Output selection	Patch ↑	HPS ↑	Pick ↑
Native	First 8	.5002	.30058	23.707
GI	64-to-8	.6170	.30128	23.840
One query t_8	First 8	.6406	.30024	23.491
One query t_8	HPS top 8 of 16	.6192	.30633	23.851
One query t_8	Pick top 8 of 16	.5932	.30447	24.004
One query t_8	Joint rank top 8 of 16	.6134	.30580	23.972
Two queries t_4, t_5	First 8	.6584	.29607	23.231
Two queries t_4, t_5	HPS top 8 of 16	.6274	.30386	23.752
Two queries t_4, t_5	Pick top 8 of 16	.6283	.30261	23.867
Two queries t_4, t_5	Joint rank top 8 of 16	.6341	.30347	23.853

G Complete qualitative comparisons

The gallery collects complete image sets for the main-paper examples, open-ended prompts, baseline comparisons, and controlled interventions. Each caption specifies the configuration and pairing used in that comparison.

G.1 Complete main-paper Figure 1 sets and first-forward examples



Figure 14: **Main-paper Figure 1(a), complete bicycle set.** Qwen-Image, eight paired seeds. Ours queries t_3, t_4 for the first two updates.



Figure 15: **Main-paper Figure 1(a), complete record-shop set.** FLUX.1-dev, eight paired seeds. The one-query rule adds interior views to the largely storefront-focused Native set. Two queries also produce close views of records, turntables, and customers. The two rules differ in both offset and window length.

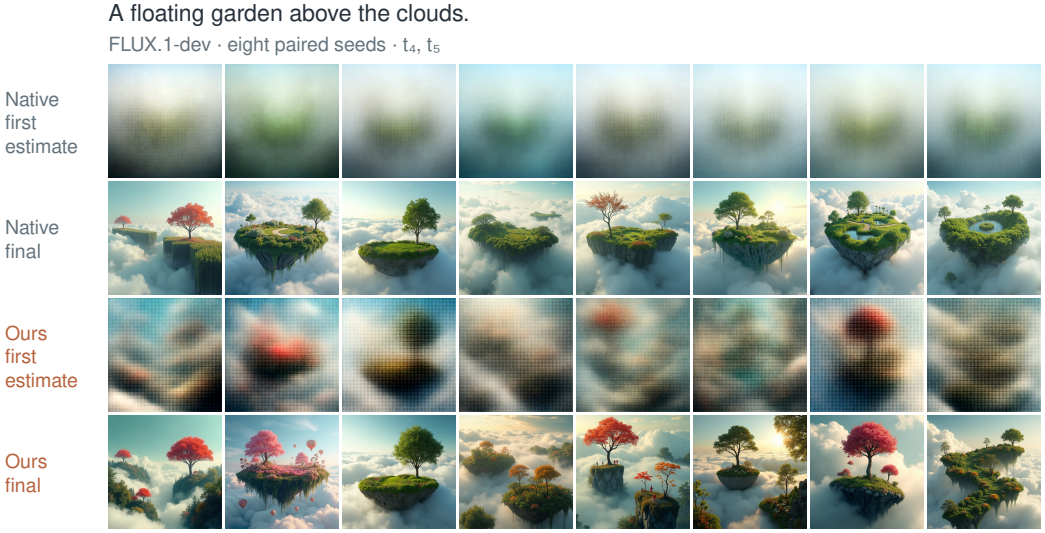

Figure 16: **Main-paper Figure 1(b), complete floating-garden set.** FLUX.1-dev, eight paired seeds. First-forward estimates are reconstructed at $\sigma_0 = 1$; final Ours images use the t_4, t_5 rule.

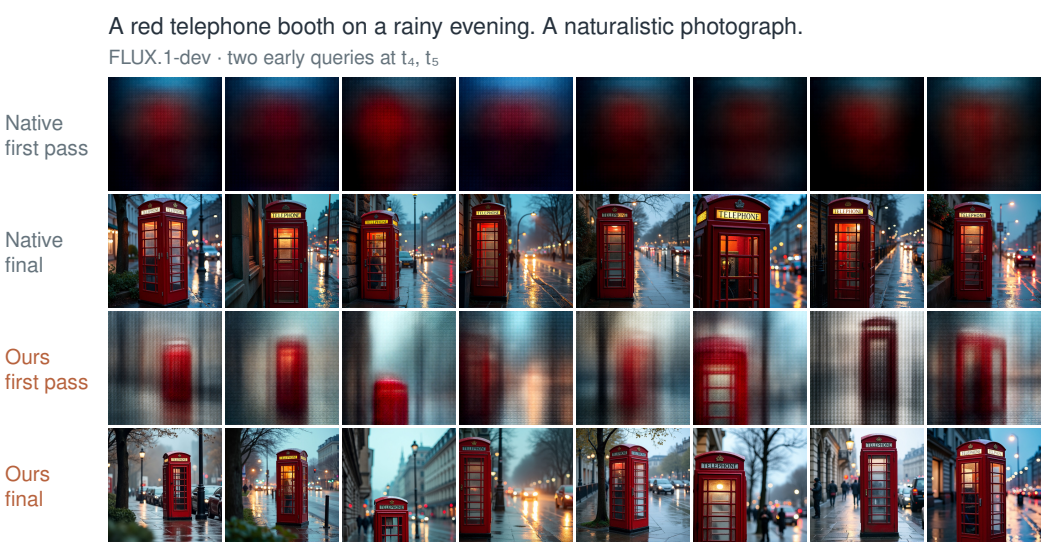

Figure 17: **First-forward structure and final framing.** FLUX.1-dev, eight paired seeds. Native first estimates share a similar red-booth silhouette. Using t_4 changes the estimated booth's scale and position; the final Ours row uses the two-query t_4, t_5 rule. Both first-pass estimates are computed as $z_0 - v$ at the actual $\sigma_0 = 1$.

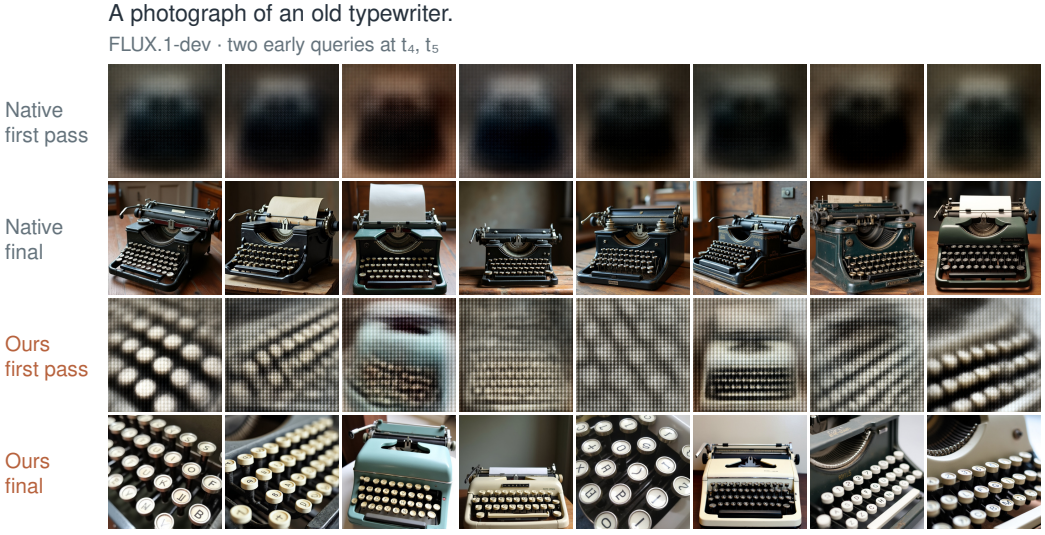

Figure 18: **First-forward structure and final framing: typewriter.** All eight paired seeds, using the same Dev t_4, t_5 rule. The changed first estimates include keyboard close-ups, which also appear in the final outputs. Both first estimates are reconstructed at $\sigma_0 = 1$.

G.2 VISUAL VARIETY ACROSS MODELS

We use prompts that leave viewpoint and layout unspecified. Every displayed row retains all eight outputs, with seeds 691000–691007 paired across Native and our method. Dev uses one query at t_8 or two at t_4, t_5; Qwen uses two at t_3, t_4; Schnell changes its first condition to .975. All other sampling settings remain native. These illustrative prompts are separate from the quantitative evaluation set.



Figure 19: **FLUX.1-dev: a rooftop robot gardener.** Eight paired seeds under both fixed rules. The altered queries introduce closer robot views alongside wider rooftop framing, changing the relative prominence of the robot, vegetables, and city. The two-query set also varies illustration style.



Figure 20: **FLUX.1-dev: a street food stall.** Both rules add closer views of the people and food preparation, while retaining wider stall views in other seeds.



Figure 21: **FLUX.1-schnell: an airship arriving at a mountain settlement.** ETC adds closer frontal and underside views alongside distant airships, varying the vessel's prominence relative to the surrounding settlement.



Figure 22: **FLUX.1-schnell: an old typewriter.** Native outputs concentrate on frontal views. ETC introduces oblique views and closer framing while retaining frontal views in other seeds. All eight outputs remain in their original paired-seed order.

G.3 ACTIVITY-BASED COMPARISON

The activity-based comparison below tests how methods vary the framing of a person and their work. Section G.4 then compares object views and use scenes, including TSR and CNO. Both show complete eight-image sets. GI progressively selects eight outputs from 64 candidates; its returned order is not paired by column with the other methods.



Figure 23: **An activity-based Qwen comparison.** All eight outputs per method for a craftsperson shaping a clay bowl. Native, guidance interval, SPELL, and Ours share seeds; GI retains its selected order from 64 candidates and is not paired by column. Ours is the bottom row and uses the fixed t_3, t_4 rule. Guidance interval and Ours include wider views of the craftsperson alongside close views of the hands and bowl.

G.4 BENCHMARK COMPARISONS

The following Qwen-Image and FLUX.1-schnell sets show all eight outputs for each method on selected prompts from the 553-prompt evaluation. Non-GI rows start from the same seeds, 229000–229007; CNO optimizes that initial noise with learning rate .02 on Qwen and .01 on Schnell, matching the main comparison. GI selects eight outputs from a 64-candidate pool and is not paired by column. Ours uses two queries at t_3, t_4 on Qwen and the first condition .975 on Schnell. These sets extend the main qualitative comparison across object categories using these fixed configurations. The comparisons make the types of added variation explicit: ETC adds changes in viewpoint, subject scale, and use context to changes in appearance. Such distinctions are visible in the complete sets, whereas a mean feature distance summarizes their combined effect in one number. For example, the skateboard comparison includes both isolated boards and skating scenes, and the car comparison includes side, rear, and detail views.

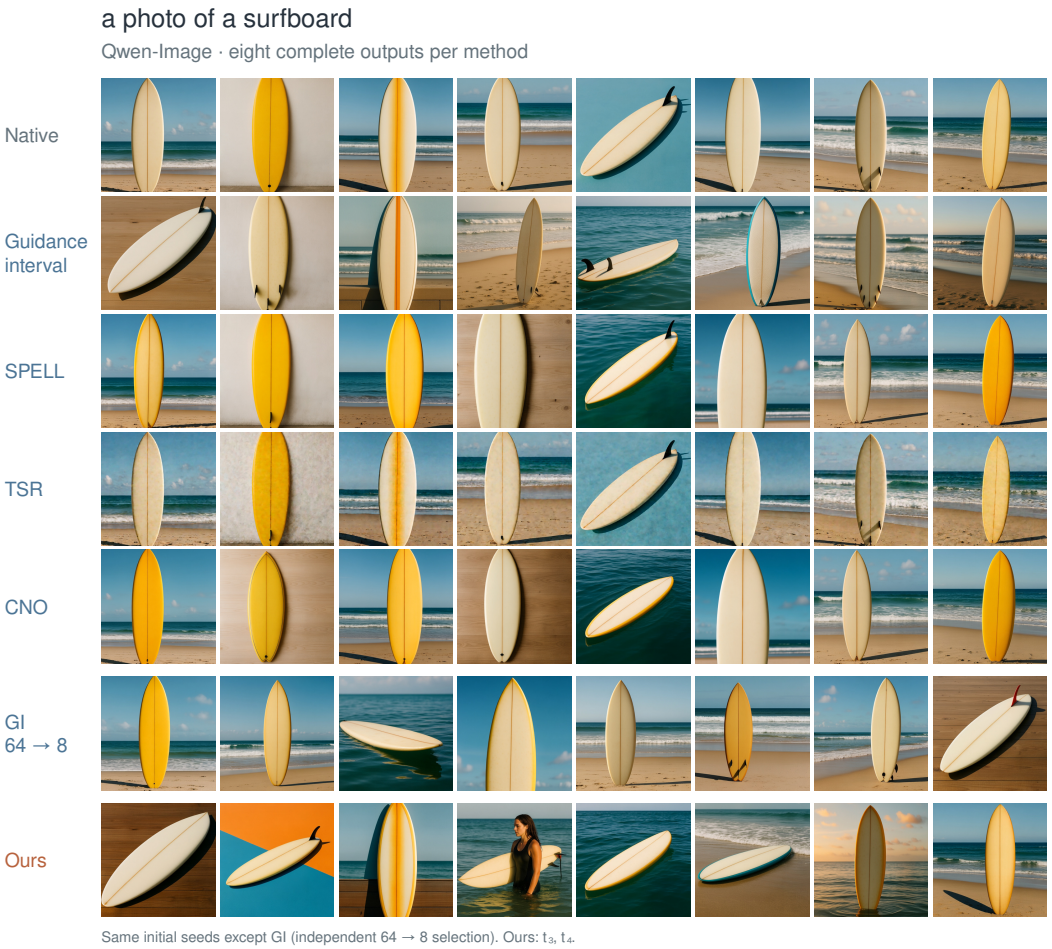

Figure 24: **Surfboard: orientation and scene context.** Complete comparison including TSR and CNO. Most Native outputs place an upright board near the center. Ours also includes diagonal and horizontal views, a close-up, and a person carrying the board.

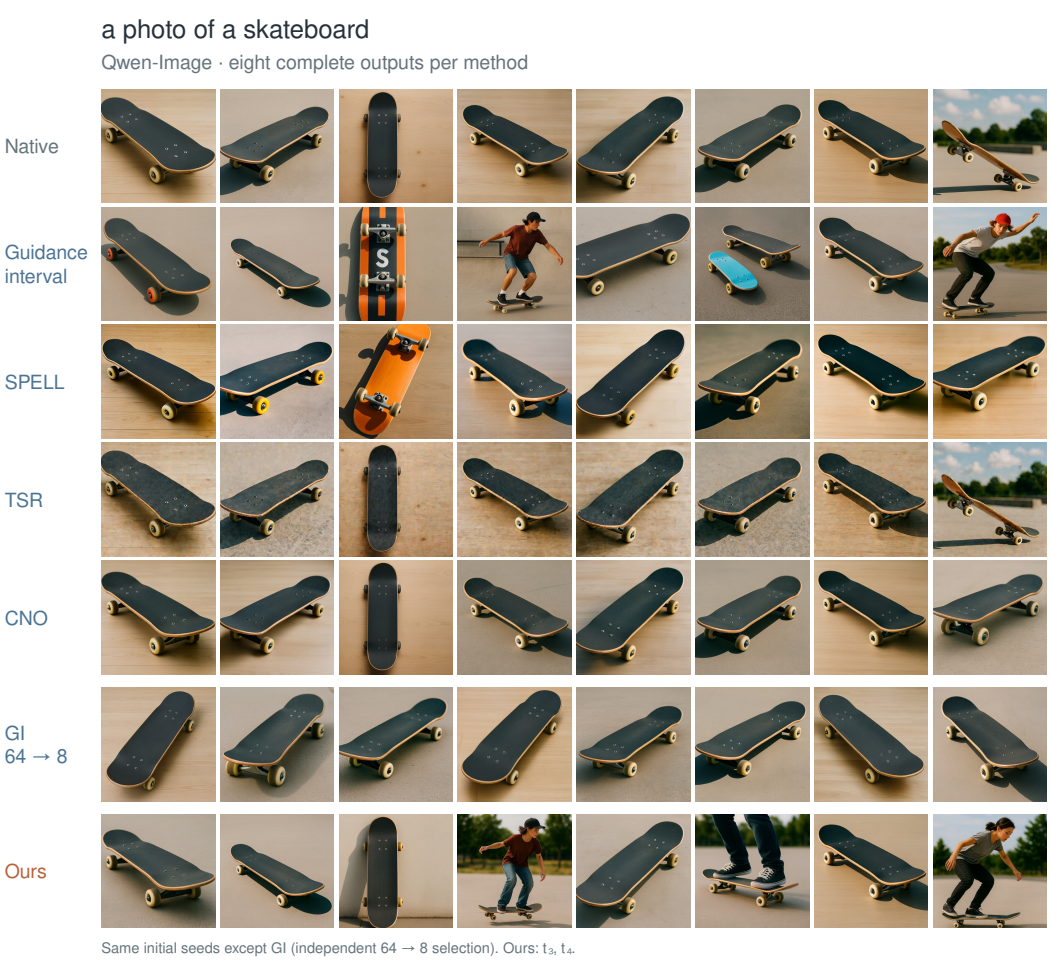

Figure 25: **Skateboard: from object views to use scenes.** Native mostly produces isolated board views. Both guidance interval and Ours introduce skating scenes; Ours also includes a close view of feet on the board. All eight outputs are retained for every method.

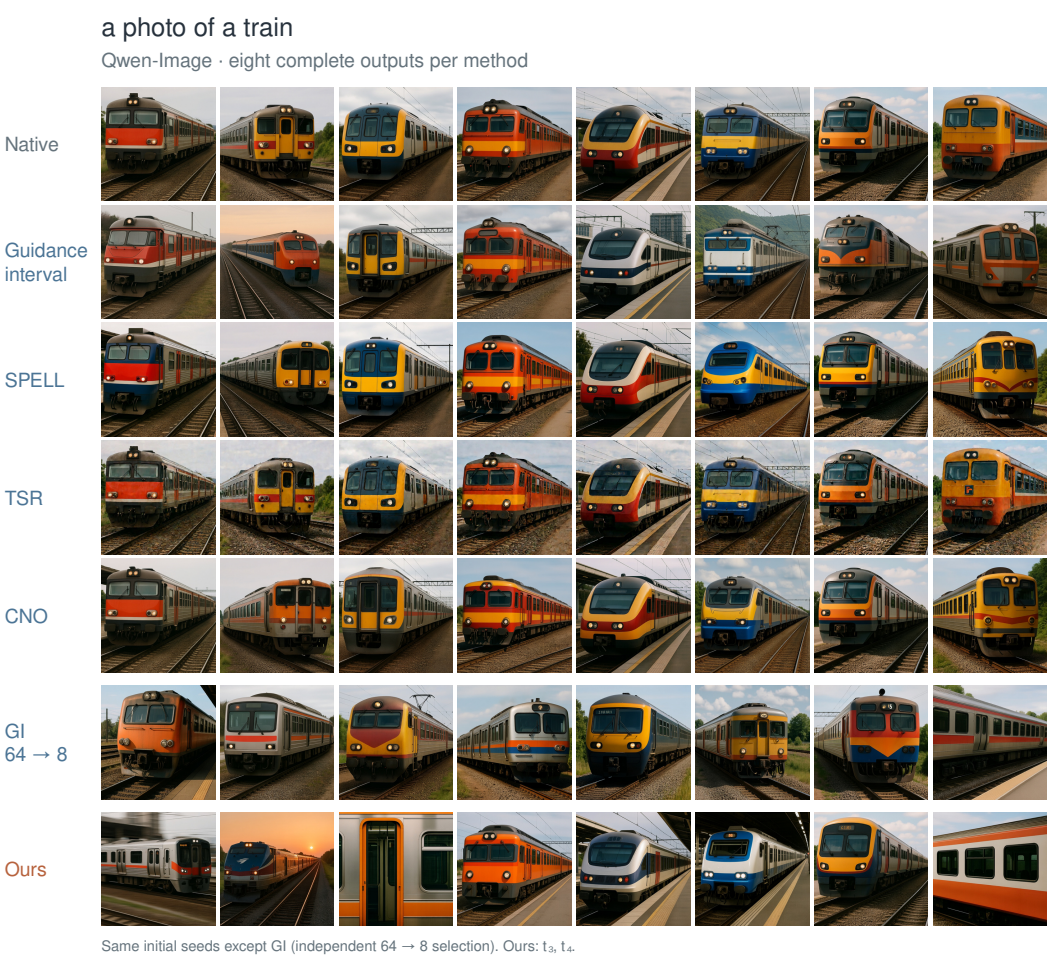

Figure 26: **Train: viewpoint and scale.** The Native set largely shares a front-oblique viewpoint. Ours adds side-on and door-detail views alongside wider train views, changing framing as well as train appearance.

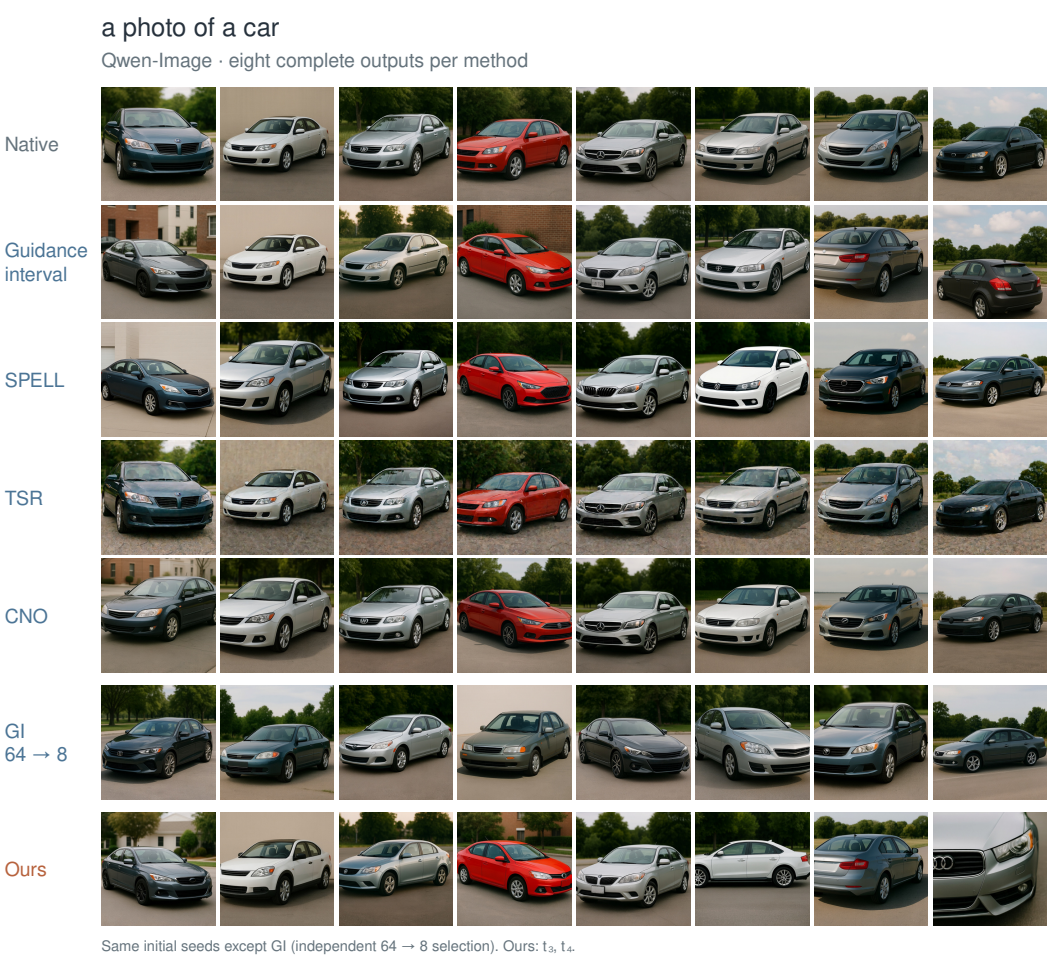

Figure 27: **Qwen-Image: car viewpoints.** Native, SPELL, TSR, CNO, and GI largely retain front-oblique views. Guidance interval also introduces rear views; Ours includes side-on, rear-oblique, and close-detail framing. All eight outputs are shown, with Ours at the bottom.



Figure 28: **FLUX.1-schnell: backpacks in use.** Native, SPELL, TSR, and GI mainly show standalone bags. Ours adds worn backpacks at different subject scales, from close views to a distant walker, while retaining standalone views. CNO also introduces a person in one output. GI retains all eight selected outputs from its 64-candidate pool.

G.5 SKIPPING VERSUS CONDITIONING-ONLY GENERATION

The following plates show all eight outputs for the same single-object prompt and seed order. They expose the model-dependent effect of removing early updates alongside conditioning-only generation.



Figure 29: **Dev: skipping and conditioning only.** Complete paired eight-output sets for "a photo of a traffic light." The two conditioning rows use the fixed t_8 and t_4, t_5 rules. Differences include framing and viewpoint, as well as occasional changes in rendering style.

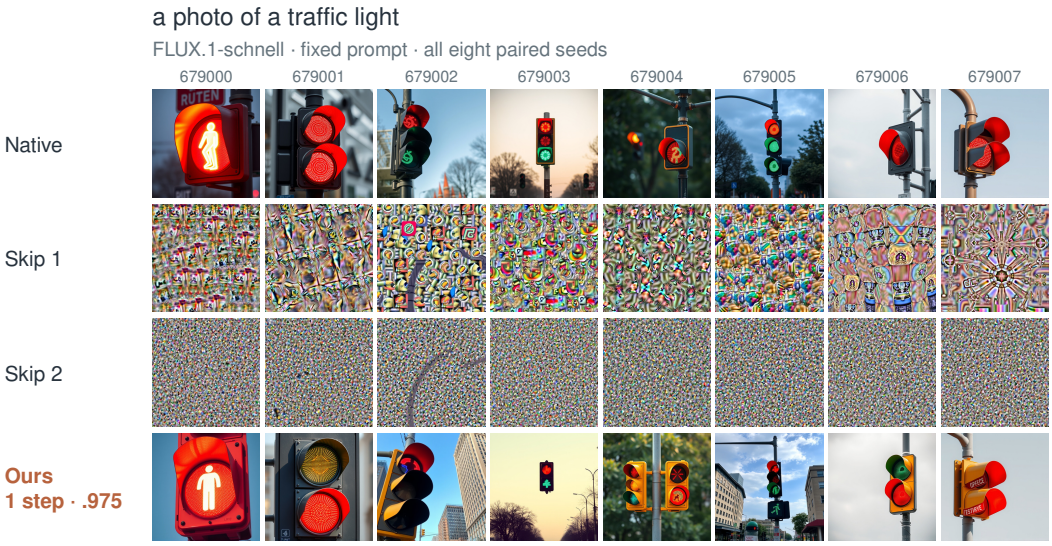

Figure 30: **Schnell: the cost of dropping early updates.** Same prompt and seeds as the other skip plates. Skipped outputs can become repetitive textures, whereas the .975 conditioning-only rule retains all four updates and generates recognizable traffic-light scenes.



Figure 31: **Qwen: skipping and conditioning only.** Complete paired eight-output sets; the conditioning rule uses t_3, t_4. Changes in this example are mainly in viewpoint and crop.

G.6 LONG-FORM PROMPT COMPARISON

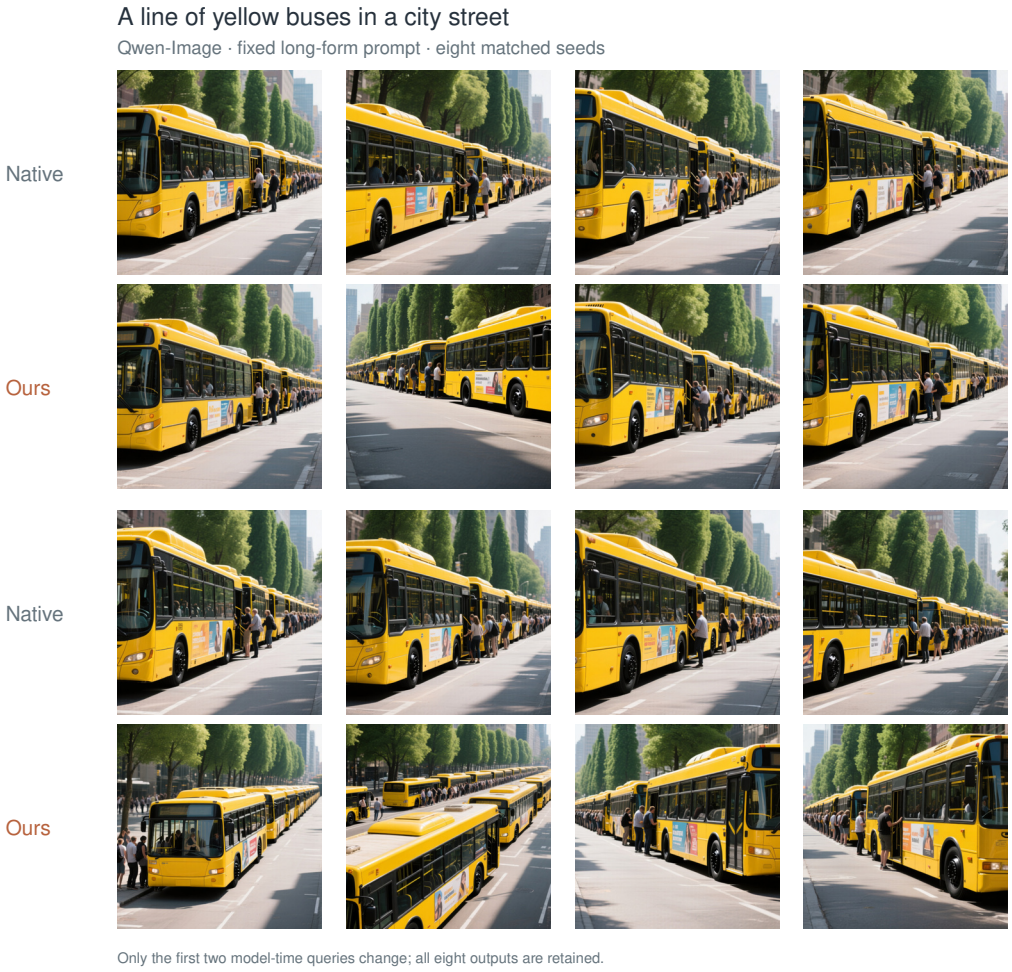

Figure 32: **Complete long-prompt comparison.** Qwen-Image on a frozen DPG-Bench description of a yellow lead bus, queued buses, boarding passengers, and trees (source row 999). Native and Ours share all eight initial seeds; Ours changes only the first two time queries to t_3, t_4. Native repeatedly uses a similar roadside camera, while Ours includes central, side-on, and higher views of the bus line. Table 11 gives the aggregate 96-prompt result.