RELOAD: PACKED FUTURE-STATE RE-ENCODING FOR LOW-LATENCY EXACT SPECULATIVE DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Exact speculative diffusion preserves a designated discretized target distribution, but autoregressive drafting leaves sequential neural calls on the critical path. Predicting a whole horizon from one anchor removes this rollout yet leaves later predictions without representations of their random future source states. We call this mismatch *future-state representation staleness*. RELOAD, a two-pass drafter, first constructs causal future sources from parallel coarse predictions. A packed second pass re-encodes these sources, and the scheduler rebuilds proposals from the same anchor and innovations. Rank-eight Role-LoRA specializes this second role with 267,264 trainable parameters. We prove that the resulting proposals retain prefix causality, allowing the unchanged reflection verifier to preserve the target-chain law in exact arithmetic under matched-covariance assumptions. At horizon eight, re-encoding raises mean accepted length from 2.02 to 4.52, and role adaptation further raises it to 4.77 in a one-GPU component study. On 256 paired ImageNet-256 cases, RELOAD reduces mean sampling latency by 10.0% relative to reproduced strict FREE, a 1.111× speedup under the same eight-rank allocation. Eight-way verification yields a 2.286× latency speedup over one-active-rank TARGET.

1 INTRODUCTION

Sequential neural evaluations bottleneck diffusion sampling (Ho et al., 2020; Song et al., 2021). Exact speculative diffusion drafts several transitions, verifies them in parallel, and preserves the target-chain law through acceptance and correction (De Bortoli et al., 2025; Wen et al., 2025).

Autoregressive drafters such as FREE condition on each current state but require L dependent neural forwards. One-pass language-model drafters instead predict future positions from shared context (Chen et al., 2026; Liu et al., 2026). In diffusion, later transitions act on random states produced by earlier updates: knowing the future timestep does not provide the future state. We call this *future-state representation staleness*: predictions reuse the anchor representation as source states evolve.

Re-encoding each rebuilt state would restore recurrent neural drafting. We use a causal coarse trajectory to supply different predictor inputs in parallel, excluding current and future innovations. These predictions then rebuild a trajectory from the same anchor. Neural passes need not follow the rebuilt trajectory step by step; verification uses rebuilt states and recorded proposal densities.

We develop RELOAD, a two-pass drafter, with the following contributions:

- **Source-state diagnosis.** Fixed-weight interventions isolate the benefit of coarse over anchor inputs, before and after adaptation (Section 4.3).
- **Two-pass drafting.** We restore local state information through packed re-encoding without recurrent neural calls; Role-LoRA adapts the second pass to future source states (Hu et al., 2022) (Section 3).
- **Causality and acceptance.** We prove target-law preservation under matched Gaussian covariances in exact arithmetic and bound acceptance through prediction error and source displacement (Section 3.5).

On 256 paired ImageNet-256 cases with DiT-XL/2 and 1,000-step DDPM, RELOAD reduces latency by 10.0% over reproduced strict FREE at the same eight-active-rank allocation (Table 1). We isolate

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.

the effects of re-encoding and shared noise, then evaluate guidance, verification allocation, draft horizon, and agreement with 50K target-generated samples.

2 Background and related work

2.1 Speculative decoding for large language models

Speculative decoding drafts future tokens, scores them with the target in parallel, and preserves the target law through acceptance and correction (Leviathan et al., 2023; Chen et al., 2023). Efficiency depends on drafting cost and accepted-prefix length.

Sequential drafters condition on preceding proposals (Li et al., 2024). Parallel predictors reduce neural-call depth but can lose this information (Stern et al., 2018; Xiao et al., 2024; Chen et al., 2026; Liu et al., 2026); predecessor-conditioned corrections (Cheng et al., 2026) and xPress's fixed-feature Jacobi updates (Wang et al., 2026) restore intra-block dependence. Diffusion adds a continuous random source state: knowing the timestep does not reveal the state produced by earlier stochastic transitions.

2.2 Exact speculative diffusion and prefix causality

Exact speculative diffusion preserves a designated chain through acceptance and correction (De Bortoli et al., 2025). Autospeculative decoding (ASD) exploits reparameterized DDPM exchangeability to speculate without an auxiliary drafter while preserving the discretized sampler's law (Hu et al., 2025). Appendix A.8 contrasts iterative solvers and stale-state pipelining.

Prefix causality allows a transition's mean to use past noise, but not current or future noise. Condition on the round-start anchor x_{anc}, class/guidance condition c, and fixed target-feature cache a. Draw mutually independent standard Gaussian innovations $\xi_{1:L}$, independent of this history. The information after k innovations is $\mathcal{F}_k = \sigma(x_{\mathrm{anc}}, c, a, \xi_1, \dots, \xi_k)$. For $\sigma_k > 0$, target and proposal kernels share the scheduler-prescribed covariance:

$$p_k(\mathrm{d}x \mid x_{k-1}, c) = \mathcal{N}\big(x; \mu_k^p(x_{k-1}, c), \sigma_k^2 I\big)\,\mathrm{d}x, \tag{1}$$

$$q_k(\mathrm{d}x \mid \mathcal{F}_{k-1}) = \mathcal{N}\big(x; \mu_k^q, \sigma_k^2 I\big)\,\mathrm{d}x. \tag{2}$$

With $\mathcal{F}_{k-1}$-measurable μ_k^q, independence of ξ_k makes $x_k^q = \mu_k^q + \sigma_k \xi_k$ follow equation 2, the generating density used by verification. Cache a conditions only the drafter; inaccurate anchor-only predictions can still be causal. Strict FREE combines autoregressive drafting and packed target evaluation (Wen et al., 2025).

2.3 Exact verification and reflection correction

For a causal chain $x_{0:L}^q$ with $x_0^q = x_{\mathrm{anc}}$, the target scores $x_{0:L-1}^q$ itemwise in parallel (De Bortoli et al., 2025; Wen et al., 2025). Conditional on accepting positions $1, \dots, k-1$, it accepts x_k^q with probability

$$a_k(x_k^q) = \min\left\{1, \frac{p_k(x_k^q \mid x_{k-1}^q, c)}{q_k(x_k^q \mid \mathcal{F}_{k-1})}\right\}. \tag{3}$$

The ratio uses the actual source and recorded generating density, with fresh uniforms independent of innovations, previous uniforms, and round-start history.

At the first rejection, reflection coupling samples the normalized residual $[p_k - q_k]_+$ from the rejected innovation (De Bortoli et al., 2025) (Algorithm 2). The verifier commits the accepted prefix plus correction, discards the suffix, and restarts with fresh randomness. Fully accepted blocks are committed in full; zero-variance steps use the target update. Accepted length $\mathrm{AL} = \mathbb{E}[A]$ counts accepted proposals per round, excluding correction. Block verification instead uses joint prefix decisions and a weighted Decomposition residual (Soen et al., 2026). ReLoaD changes the drafter; combining it with block verification requires new conditional-density and correction arguments, and joint gains remain untested (Appendix A.8).

3 METHOD

RELOAD constructs coarse source states, re-encodes them in parallel (the rerun), then rebuilds proposals with the updated predictions for target verification (Figure 1; Algorithm 1). The two trajectories share their anchor and per-step noise. Each prediction pass uses one neural invocation; scheduler updates remain sequential.

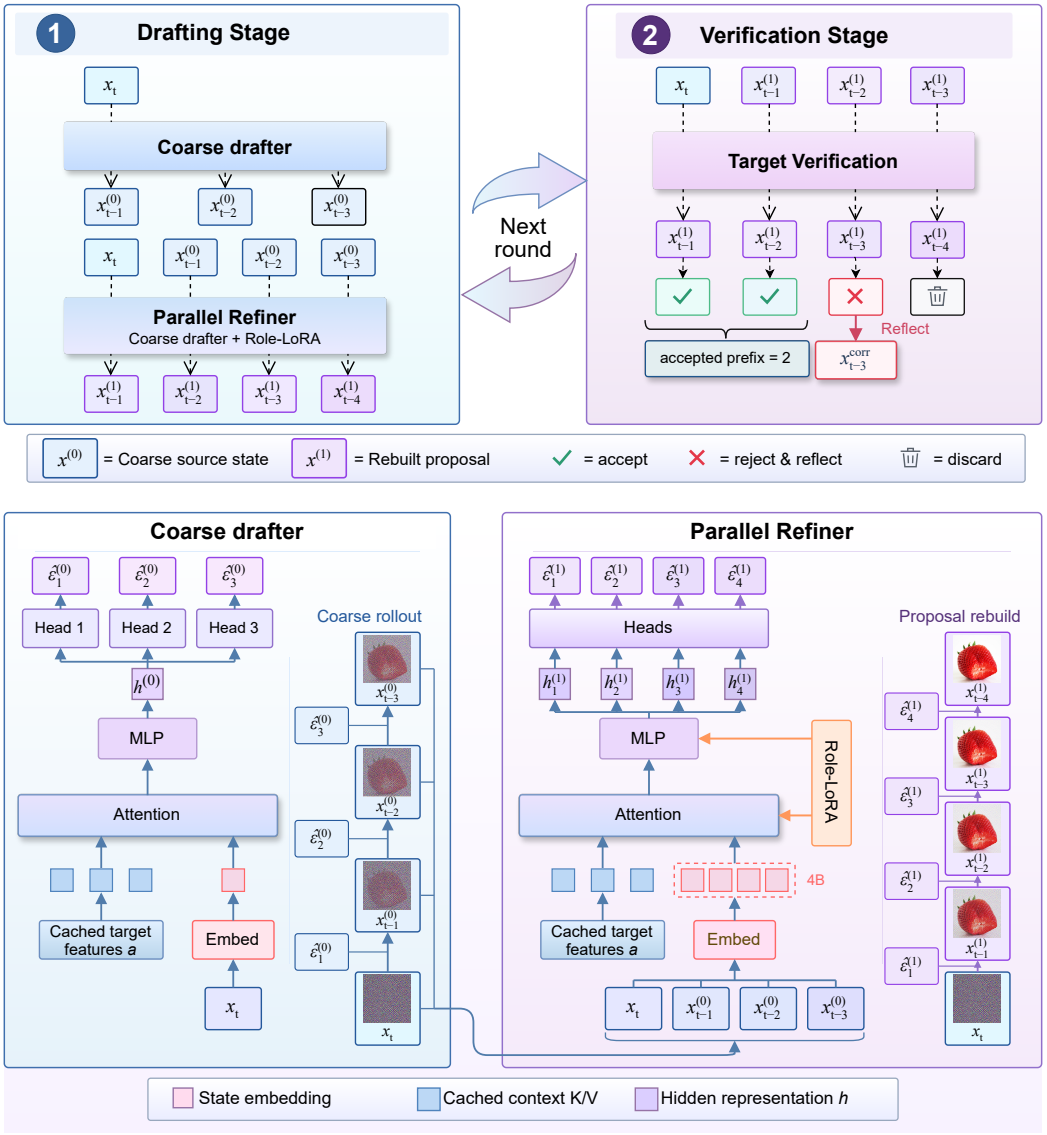


Figure 1: RELOAD at $L = 4$ (main experiments: $L = 8$). **Top:** two accepted proposals and one reflected correction are committed before the next target anchor step; the suffix is discarded. **Bottom:** coarse drafting (left) and packed re-encoding (right) share the raw target cache. Coarse sources form $4B$ rerun items; predictions rebuild proposals from the same anchor and innovations. Images are schematic illustrations.

3.1 HORIZON-PARALLEL COARSE DRAFTING

Each round obtains anchor x_{anc} by committing one target step and caches features a from its parent evaluation. Horizon L is capped by the remaining transitions; a terminal anchor ends sampling.

Adapting DFLASH's one-pass interface (Chen et al., 2026), one backbone forward encodes the anchor at t_1 and absolute-timestep residual heads predict $\widehat{\epsilon}^0_{1:L-1} = h_W(x_{\text{anc}}, t_{1:L-1}, c, a)$ (Figure 1, bottom left). The anchor supplies the first source, leaving $L-1$ predictions to construct the rest.

Let t_k denote the absolute timestep and $\mathcal{M}_k$ its fixed scheduler mean map. From $x^0_0 = x_{\text{anc}}$, construct

$$x^0_k = \mathcal{M}_k(x^0_{k-1}, \widehat{\epsilon}^0_k) + \sigma_k \xi_k, \qquad s_{k+1} = x^0_k, \quad k = 1, \ldots, L-1, \tag{4}$$

with $s_1 = x_{\text{anc}}$. These rerun sources depend only on $\xi_{1:k-1}$ and are not final proposals.

Both drafter roles share raw cache a but build separate context K/V once per round. Re-encoding refreshes local tokens and queries (Appendices B and D.2).

Guidance convention. We follow the released DiT implementation (Peebles & Xie, 2023). For $c = (y, w)$, target anchor and verification use $\epsilon^p_w = \Gamma_w(\epsilon^p_{\text{cond}}, \epsilon^p_{\text{uncond}})$, with $\Gamma_w(u, v) = u + (w-1)\Pi_3(u - v)$ and $w = 1.5$ by default. Π_3 applies the guidance increment to the first three latent channels; the fourth retains its conditional prediction. Both drafter roles use raw conditional predictions and the conditional-branch cache, without an unconditional pass or CFG mixing. Their recorded proposal densities and the guided target density define verification (Appendix D.3); Table 26 compares target guidance conventions.

3.2 PACKED FUTURE-STATE RE-ENCODING

We **re-encode all L causal sources in parallel**, packing the source–timestep pairs (s_k, t_k) into LB aligned items for one rerun invocation:

$$\widehat{\epsilon}^1_k = h_{W+\Delta W_{\text{role}}}(s_k, t_k, c, a), \qquad k = 1, \ldots, L. \tag{5}$$

Each item attends to its own source tokens and the fixed rerun K/V for that image. LayerNorm and RMSNorm reduce feature channels; AdaLN uses local conditioning, and dropout is disabled. Packing preserves these per-item operations even when histories share past innovations (Appendix D.2). Call depth is one for all L positions, while arithmetic scales in LB (Appendix B.1).

Because re-encoded predictions differ from coarse predictions, we rebuild from $x^q_0 = x_{\text{anc}}$ using the same innovations:

$$\mu^q_k = \mathcal{M}_k(x^q_{k-1}, \widehat{\epsilon}^1_k), \qquad x^q_k = \mu^q_k + \sigma_k \xi_k. \tag{6}$$

The rerun reads coarse s_k; the target scores rebuilt x^q_{k-1} independently of the drafter cache and Role-LoRA. Reflection uses the recorded generating mean and residual $x^q_k - \mu^q_k = \sigma_k \xi_k$, so differing sources preserve the verification interface. Shared innovations cancel additive noise in their displacement (Appendix A.6), improving acceptance (Table 21). Exactness also permits independent auxiliary coarse noise included in the initial augmented history (Appendix A.1).

Algorithm 1 RELOAD: one drafting round from an anchor.

Input: anchor x_{anc}, cache a, condition c, horizon L, timesteps $t_{1:L}$, scheduler $(\mathcal{M}, \sigma)$, two drafter roles, and target.

1. **Cache** each role's context K/V from the same raw a, once at batch B.
2. **Sample** independent $\xi_{1:L} \sim \mathcal{N}(0, I)$ once for both chains.
3. **Predict** $\widehat{\epsilon}^0 \leftarrow h_W(x_{\text{anc}}, t_{1:\max(1,L-1)}, c, a)$ in one forward.
4. **Construct** $s_1 = x_{\text{anc}}$ and s_{k+1} using only $\xi_{1:k}$ in equation 4.
5. **Pack** (s_k, t_k, c) into LB aligned items; expand rerun K/V across L.
6. **Re-encode** each item separately in one packed forward equation 5; no item mixing.
7. **Rebuild** from $x^q_0 = x_{\text{anc}}$ with the same ξ; retain $(x^q_k, \mu^q_k, \sigma^2_k)$ in equation 6.
8. $(x_{\text{next}}, A) \leftarrow$ STRICTVERIFY$(x^q_{0:L}, \mu^q, t, c; \mathcal{M}, \sigma,$ TARGET$)$.
9. **Commit** A accepted proposals and, if $A < L$, the correction x_{next}.
10. **Return** the endpoint; advance $A + \mathbb{I}[A < L]$ proposal transitions.

3.3 LOW-RANK ROLE SPECIALIZATION

We specialize only the rerun role with LoRA, $W_{\text{rerun}} = W + (\alpha/r)UV$ (Hu et al., 2022) (Figure 1, bottom right). It encodes the anchor at position one and coarse future sources thereafter, each at its local timestep.

The primary configuration uses $r = \alpha = 8$ on six context and transformer maps, with 267,264 trainable parameters initialized by $U = 0$ and random V (Appendix B). Updates are merged before timing; deployment keeps separate dense coarse and rerun copies. Table 20 evaluates smaller placements.

3.4 TWO-STAGE DRAFTER TRAINING

Stage I: base drafter. The joint predictions $\widehat{\epsilon}_k^{\text{base}} = [h_W(x_{\text{anc}}, t_{1:L}, c, a)]_k$ construct on-policy sources s_k through equation 4. The frozen target supplies noise references at these sources. We detach the prediction copies used for rollout and apply the regression loss to the original outputs. We train the draft block, context path, and timestep residual heads. Target-shared embeddings and the base output projection stay frozen.

Stage II: rerun role. We freeze the Stage-I EMA in both copies. The coarse copy constructs detached sources s_k; the rerun predicts $\widehat{\epsilon}_k^{\text{role}}$ by equation 5, updating only Role-LoRA. Both stages minimize

$$\mathcal{L}_v = \sum_{k=1}^{L} \omega_k \, \mathbb{E}\left[\frac{\|\widehat{\epsilon}_k^v - \epsilon_k^p(s_k, t_k, c)\|_2^2}{d}\right], \quad v \in \{\text{base}, \text{role}\}, \quad \sum_k \omega_k = 1, \tag{7}$$

where d counts latent scalars and $\omega_k \geq 0$ are fixed early-heavy weights emphasizing positions reached earlier by prefix verification. Source construction and target references receive no gradients. All training calls use $w = 1$ and the same retained or dropped class label; the reference is the frozen target's raw noise prediction at s_k, not innovation ξ_k. Training supervises all L outputs, whereas inference omits the unused terminal coarse output for $L > 1$. Table 7 and Appendix D.4 give the recipe and execution details; Table 19 compares alternative weights.

3.5 CAUSALITY AND ACCEPTANCE ANALYSIS

Since s_k reads only $\xi_{1:k-1}$, itemwise re-encoding with fixed context makes $\widehat{\epsilon}_k^1$ and rebuilt mean μ_k^q $\mathcal{F}_{k-1}$-measurable (Lemma 1, Appendix A.1). Packing target scores preserves these dependencies.

Proposition 1 (Target-law preservation)**.** *Assume causal proposals, itemwise target scoring, and proposal/target transitions matching their recorded Gaussian densities with common covariance $\sigma_k^2 I$, $\sigma_k > 0$. Draw mutually independent proposal innovations and verifier uniforms, independent of round-start history. With exact reflection correction, the concatenated committed states have the designated target-chain joint law at every fixed set of diffusion indices, in exact arithmetic.*

The common covariance may vary with timestep but is state-independent. Zero-variance terminal steps commit the target update directly. Learned or mismatched covariances and ODE steps require a different verification construction (Appendix A.8). Appendix A.3 proves the proposition for Algorithm 2; finite-precision checks appear in Table 32 and Appendix E.9.2.

From source-state error to acceptance. Exactness permits a predictor to read a different causal source from the one scored by the target. Efficiency depends on how that difference affects the proposal mean. For the guided target field $f_k(x) = \epsilon_w^p(x, t_k, y)$, separate prediction error at the encoded coarse source from displacement to the rebuilt verification source:

$$\eta_k = \|\widehat{\epsilon}_k^1 - f_k(s_k)\|_2, \qquad \delta_k = \|s_k - x_{k-1}^q\|_2. \tag{8}$$

Training at $w = 1$ supervises raw conditional prediction error, whereas η_k compares the rerun prediction with the deployed guided target. It therefore includes the CFG discrepancy as well as prediction error (Appendix A.6).

Proposition 2 (Source-error bound on conditional acceptance)**.** *Under Proposition 1, condition on the legal history at a reached stochastic position k. Suppose f_k is K_k-Lipschitz between the two sources and $\mathcal{M}_k(x_{k-1}^q, \cdot)$ is B_k-Lipschitz on the relevant noise predictions, in Euclidean norm.*

$$\|\mu_k^q - \mu_k^p\|_2 \leq B_k(\eta_k + K_k\delta_k), \tag{9}$$

$$o_k^{\text{live}} = 2\Phi\left(-\frac{\|\mu_k^q - \mu_k^p\|_2}{2\sigma_k}\right) \geq 2\Phi\left(-\frac{B_k(\eta_k + K_k\delta_k)}{2\sigma_k}\right), \tag{10}$$

where o_k^{live} is acceptance probability averaged over the next proposal and verifier uniform, conditional on that history, and Φ is the standard normal CDF.

Proof sketch. Add and subtract $f_k(s_k)$, apply the Lipschitz bounds, and use the Gaussian overlap identity. For the affine scheduler $\mathcal{M}_k(x, e) = a_k^{\mathrm{sch}} x + b_k^{\mathrm{sch}} e$, $B_k = |b_k^{\mathrm{sch}}|$ is known; target regularity K_k is assumed. The bound identifies two sources of acceptance loss: prediction error and source displacement. Numerical calibration would also require estimating K_k. Appendix A.6 gives the proof, accepted-length consequence, and shared-noise displacement recursion.

4 EXPERIMENTS

4.1 SETUP AND EVALUATION PROTOCOL

We evaluate class-conditional DiT-XL/2 (Peebles & Xie, 2023) on ImageNet-256 (Russakovsky et al., 2015), using 1,000-step fixed-posterior DDPM (Ho et al., 2020) and CFG 1.5 (Ho & Salimans, 2022) unless specified. With AdamW (Loshchilov & Hutter, 2019), Stage I uses 180,120 updates on four A100s (batch 256); Stage II uses up to 8,000 on one A100 (batch 16), with step-7000 EMA preselected (Appendix C.2). Appendix C.3 reports the logged Stage-II cost (1.001 A100 GPU-hours) and training-cost amortization.

The primary $L = 8$ study pairs RELOAD, reproduced strict FREE (Wen et al., 2025), and sequential TARGET on 256 locked cases on one host. All allocate eight ranks; TARGET activates one, speculative methods eight. We report mean sampler seconds per image, ratios of means, and strict AL (accepted drafts per round, averaged over cases). Calls count logical target invocations. Appendix C.5 defines paired-bootstrap 95% intervals and GPU-second accounting; Table 9 gives acceptance and calls.

Our FREE reproduces the published algorithmic contract and performance scale; an independent rerun agrees within 0.3% on algorithmic metrics (Table 10). Speedups use matched hosts and case banks (Table 8).

The main latency, source-input, and core-component studies use Primary Role7000 for the adapted re-encoder. Follow-up Rank8 is a different step-7000 EMA used in the supplementary adaptation, shared-noise, and CFG-channel studies. Table 6 maps these studies to their checkpoints, cases, and timing status.

4.2 MAIN LATENCY RESULTS

Table 1: Primary comparison: 256 paired cases, $L = 8$, eight allocated ranks. Active-rank GPU-s is latency times ranks with a target invocation, including waiting, not FLOPs. A dash means not applicable.

Method	Time (s/image)↓	Speedup vs. Target↑	AL↑	Active ranks	Active-rank GPU-s/image
TARGET	12.535	1.000×	–	1	12.535
Strict FREE	6.092	2.058×	4.5363	8	48.737
RELOAD	**5.482**	**2.286**×	**4.7645**	8	43.859

Relative to strict FREE, RELOAD reduces latency and both GPU-second measures by 10.0%, yielding a 1.111× latency speedup (95% CI [1.105,1.117]). It is faster on 254/256 paired cases. Mean AL increases by 0.2281 ([0.1919,0.2648]), with 9.21 fewer logical target calls ($[7.54, 10.91]$). Eight-way verification gives 2.286× speedup over one-active-rank TARGET ([2.275,2.298]), at the higher active-rank GPU-second cost in Table 1.

Training-free score reuse. Our carried-score control removes drafter calls but lowers acceptance under the same sample-level verifier; its previous-source noise reuse differs from ASD (Hu et al., 2025) (Appendix E.2). On 128 paired cases, RELOAD is $1.322\times$ faster than this control and $1.100\times$ faster than strict FREE in the same run ($[1.092, 1.108]$; Table 11).

4.3 DIAGNOSING FUTURE-STATE REPRESENTATION STALENESS

To isolate source information, we fix predictor weights and vary inputs on 1,024 paired target trajectories disjoint from training: repeated anchors (B0), causal coarse states (B1), or target future states (B2). We test base weights and the Primary Role7000 re-encoder. B0 uses local predictions rather than the deployment one-pass interface; B2 is a non-deployable reference.

All predictions are scored at target-trajectory states: $\mu^q_{i,k} = \mathcal{M}_k(x^p_{i,k-1}, \widehat{\epsilon}_{i,k})$ and $\mu^p_{i,k} = \mathcal{M}_k(x^p_{i,k-1}, \epsilon^p_{i,k})$. Gaussian overlap $o_{i,k}$ (De Bortoli et al., 2025) measures transition agreement and yields the prefix proxy $\widetilde{A}_i$:

$$o_{i,k} = 2\Phi\left(-\frac{\|\mu^q_{i,k} - \mu^p_{i,k}\|_2}{2\sigma_k}\right), \qquad \widetilde{A}_i = \sum_{k=1}^{L}\prod_{j=1}^{k} o_{i,j}. \tag{11}$$

Live AL also depends on rebuilt histories and prefix survival (Appendix A.5).

Source information before adaptation. With base weights frozen, changing anchor inputs to coarse inputs raises the mean prefix proxy from 2.0946 to 5.0220; target-source inputs yield 5.0119. Coarse inputs improve the proxy on all 1,024 cases, with mean gain 2.9273 ($[2.8360, 3.0192]$), before Role-LoRA training (Table 12).

Position-wise behavior after adaptation. With adapted re-encoder weights fixed, repeated-anchor overlap deteriorates at later positions, whereas coarse-source inputs track the target-source reference (Figure 2). At position eight, B0/B1/B2 overlap is 0.2849, 0.8264, and 0.8384. Their prefix proxies are 2.1239, 5.3674, and 5.4246; B1 recovers 98.27% of the B0-to-B2 gain.

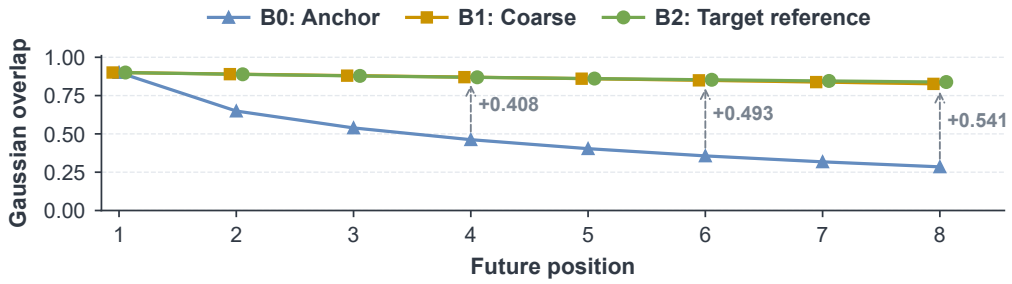


Figure 2: Source-input diagnostic with Primary Role7000 on 1,024 disjoint target trajectories (95% bootstrap bands). Coarse sources (B1) recover late-position overlap lost by anchor inputs (B0), approaching target reference (B2). Arrows: B1-minus-B0 overlap; B1/B2 markers offset horizontally for visibility.

4.4 COMPONENT ABLATIONS AND TRAJECTORY COUPLING

The core ablation separates source conditioning, execution layout, and adaptation in the deployed sampler (Table 2). It uses the same base, Primary Role7000, and 256 cases as the headline comparison, with each case sampled on one GPU.

Table 2: Core components: 256 paired cases, $L = 8$, one-GPU sampling with strict verification. Full acceptance/call statistics and the separate packed-refiner pilot are in Table 14.

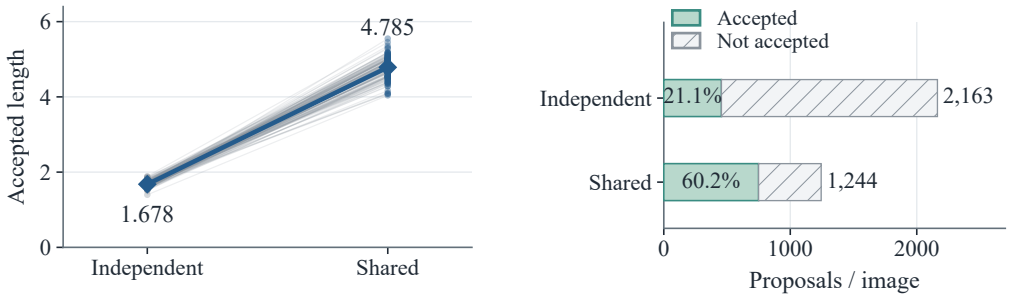
Variant	Rerun weights	AL↑	Time (s)↓
Anchor one-pass	No rerun	2.0155	20.3256
Serial re-encoding	Frozen base	4.5284	15.5544
Packed re-encoding	Frozen base	4.5240	13.8242
RELOAD	Primary Role7000	**4.7703**	**13.3846**

Frozen re-encoding supplies most of the AL gain. Packing reduces latency with an AL change of -0.0044; Role-LoRA adds 0.2463 AL and saves 0.440 seconds. On a disjoint 64-case holdout, changing only the adapters gives $+0.2331$ AL ($[0.1547, 0.3085]$) and 9.80 fewer calls ($[6.16, 13.27]$; Table 15).

Choice of adaptation. On a separate 128-case bank, adaptation methods share the base and checkpoint-selection budget (Table 16). Role-LoRA (Follow-up Rank8) raises frozen-rerun AL from 4.564 to 4.818 with a $1.035\times$ speedup, also exceeding the selected fully fine-tuned rerun's AL. Across three training seeds, Role-LoRA averages 4.802 AL versus 4.771 for conditioning-only tuning of the context projection and AdaLN (Table 17). All three differences favor Role-LoRA, although each paired 95% interval includes zero. Role-LoRA uses 267,264 trainable parameters versus 14,605,056 for conditioning-only. Overall, Role-LoRA provides an effective, parameter-efficient specialization of the re-encoding role.

Supplementary adaptation studies. In the Final128 comparison, rank four retains the Follow-up Rank8 gain with half the parameters. Neither rank sixteen nor uniform weighting significantly improves AL; reversed weighting and the selected Packed-FREE checkpoint yield lower AL (Table 18). A separate placement run finds that removing conditioning adapters improves AL by 0.0463 (pointwise interval $[0.0051, 0.0863]$), while removing attention or MLP adapters has no significant effect (Table 20). Appendix E.6 places the full sweeps alongside their training, selection, and timing protocols.

Shared causal innovations. The shared-noise study holds Follow-up Rank8 fixed throughout. Sharing innovations improves AL on all 128 paired cases, raising its mean from 1.6782 to 4.7851, pooled acceptance from 21.07% to 60.23%, and reducing calls from 544.31 to 314.45 (Figure 3; Table 21). Both designs admit exact verification under Section 3.2's conditions. This acceptance gain is consistent with the displacement analysis in Section 3.5: shared innovations cancel additive noise in the difference between coarse and rebuilt states.



(a) Paired acceptance gains

(b) Proposal utilization

Figure 3: Shared versus independent auxiliary noise: fixed Follow-up Rank8, $L = 8$, CFG 1.5, strict verification. **(a)** 128 paired AL values and their means (diamonds). **(b)** Mean accepted/not-accepted counts, including rejected and discarded proposals; labels give total counts and pooled acceptance.

4.5 Efficiency and robustness

Horizon and packed work. In the four-rank sweep, increasing L from 8 to 16 raises ReLoaD AL from 4.7672 to 6.0362 but latency from 6.243 to 7.046 seconds (Table 22). Both methods select $L = 8$ by latency on 64 cases. On 192 disjoint test cases, ReLoaD is $1.143\times$ faster than strict FREE ($[1.135, 1.150]$; Table 23). An untuned online pilot improves AL and calls but increases instrumented elapsed time (Appendix E.8.5); formal timing retains fixed horizons.

At $B = 1, L = 8$, packing cuts re-encoder-only latency from 9.778 to 2.210 ms while increasing additional peak allocation from 19.5 to 120.6 MiB (Table 24). Full-sampler coordinator peaks are 2.574, 2.711, and 2.939 GiB for Target, FREE, and ReLoaD (Table 28).

Verification allocation. Across $G \in \{1, 2, 4, 8\}$ ranks on 32 paired cases, speedup grows from $1.198\times$ to $2.200\times$ over one-rank TARGET and $1.072\times$ to $1.116\times$ over FREE. Moving from four to eight ranks cuts latency from 7.415 to 6.985 seconds but raises GPU-seconds from 29.66 to 55.88 (Table 25).

Guidance. With checkpoints and $L = 8$ fixed, RELOAD improves AL and latency over strict FREE at all three CFG scales (Table 3).

Table 3: Guidance robustness: 64 paired cases per CFG, fixed checkpoints, $L = 8$, four ranks. Speedup is mean FREE time divided by mean ReLoaD time, with paired-bootstrap 95% intervals.

CFG	Target (s)	FREE (s)	ReLoaD (s)	FREE AL↑	ReLoaD AL↑	Speedup vs. FREE↑	95% CI
1.0	11.801	5.914	5.017	4.838	5.122	$1.179\times$	$[1.169, 1.189]$
1.5	12.178	7.179	6.203	4.479	4.764	$1.157\times$	$[1.143, 1.171]$
3.0	12.214	8.813	7.741	3.187	3.370	$1.138\times$	$[1.120, 1.157]$

At CFG above one, DiT's three-channel convention yields higher acceptance than four-channel target guidance. Both variants apply guidance to anchor and verification while retaining raw conditional drafters; at CFG 1, final samples and acceptance counts coincide (Table 26).

Cross-system repeatability and implementation. On two servers, the same checkpoints and 32 paired cases give $1.146\times$ and $1.178\times$ within-server speedups over strict FREE, with identical per-case AL and calls (Table 27). A full-trajectory audit verifies the parent-feature inputs and delivered rerun K/V in all 151 rounds (Appendix D.2). Future-noise interventions test causality; layout and precision replays test numerical sensitivity, with one offline decision changing under a precision toggle (Table 32; Appendix E.9.2).

4.6 FINITE-SAMPLE AGREEMENT WITH THE TARGET SAMPLER

With 50 generated images per ImageNet class, the RELOAD–TARGET 50K comparison gives FID 1.0851 and near-zero KID (Table 4). Banks share labels and initial noise, with method-dependent later randomness (Appendix E.10).

Table 4: Generated target-relative quality, with 50K class-balanced samples per method and a separate 25K target split. KID: mean ± subset standard deviation, scaled by 10^3, not a confidence interval. Precision/recall use $k = 3$; dashes mean unmeasured.

Generated sets	Samples	FID↓	KID $\times 10^3$↓ mean ± std.	Prec.↑	Rec.↑
RELOAD–TARGET	50K/50K	1.0851	-0.00025 ± 0.05740	0.72294	0.72890
Target even–odd	25K/25K	2.8089	0.00884 ± 0.07368	—	—

The independent 10K banks cover all 1,000 classes with unequal counts. RELOAD–Target-B FID is 5.860, versus 5.831 for Target-A–Target-B (Table 36). Table 4 also reports a separate, smaller 25K target split.

5 CONCLUSION

RELOAD re-encodes causal future states in a packed pass with rerun-only Role-LoRA; parallel coarse prediction supplies those states. The scheduler uses the updated predictions to rebuild a separate proposal trajectory. Under matched Gaussian covariances, reflection verification preserves the target-chain law in exact arithmetic. Coarse inputs recover late-position overlap in the source-input diagnostic, while shared innovations increase acceptance in the deployed sampler. On ImageNet-256 with DiT-XL/2 and a 1,000-step DDPM sampler, this two-pass design reduces latency by 10.0% over reproduced strict FREE at the same eight-active-rank allocation, without recurrent neural drafting.

AI USE STATEMENT

We used generative AI tools to draft and edit manuscript sections, translate text, retrieve and review relevant literature, and prepare LaTeX and figures. These tools also provided feedback on experimental design, assisted with code implementation and audits, helped interpret experimental results, and assisted in constructing, writing, and checking mathematical arguments and proofs. Reported measurements were computed from experimental outputs, including images generated by the diffusion samplers under study. The authors have reviewed all AI-assisted content and independently checked the cited sources, mathematical arguments and proofs, code, and numerical results. The authors take responsibility for the final content of the paper.

ETHICS STATEMENT

This work introduces no human-subject data. Lower sampling latency can facilitate both beneficial uses and misuse; it does not necessarily reduce total compute. Deployment inherits the target model's capabilities, biases, licensing and data constraints, and therefore requires provenance and misuse evaluation.

REPRODUCIBILITY STATEMENT

Section 4 defines the paired protocol; Appendices A–E provide proofs, parameter counts, execution checks, and experimental protocols. The anonymous supplementary package contains the primary implementation, training and evaluation configurations, the locked case bank, raw paired measurements, and scripts for recomputing core statistics. Upon acceptance, we will open-source the code and release the Stage-I base, Primary Role7000, and reproduced strict FREE checkpoints used in the main experiment.

REFERENCES

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Jian Chen, Yesheng Liang, and Zhijian Liu. DFlash: Block diffusion for flash speculative decoding. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026.

Zigeng Chen, Xinyin Ma, Gongfan Fang, Zhenxiong Tan, and Xinchao Wang. AsyncDiff: Parallelizing diffusion models by asynchronous denoising. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95170–95197, 2024. doi: 10.52202/079017-3015.

Xin Cheng, Xingkai Yu, Chenze Shao, Jiashi Li, Yunfan Xiong, Yi Qian, Jiaqi Zhu, Shirong Ma, Xiaokang Zhang, Jiasheng Ye, Qinyu Chen, Chengqi Deng, Jiping Yu, Damai Dai, Zhengyan Zhang, Yixuan Wei, Yixuan Tan, Wenkai Yang, Runxin Xu, Yu Wu, Zhean Xu, Xuanyu Wang, Muyang Chen, Rui Tian, Xiao Bi, Zhewen Hao, Shaoyuan Chen, Huanqi Cao, Wentao Zhang, Anyi Xu, Huishuai Zhang, Dongyan Zhao, and Wenfeng Liang. DSpark: Confidence-scheduled speculative decoding with semi-autoregressive generation. *arXiv preprint arXiv:2607.05147*, 2026.

Valentin De Bortoli, Alexandre Galashov, Arthur Gretton, and Arnaud Doucet. Accelerated diffusion models via speculative sampling. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 12590–12631. PMLR, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. doi: 10.48550/arXiv.2207.12598.

Jonathan Ho, Ajay N. Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Hengyuan Hu, Aniket Das, Dorsa Sadigh, and Nima Anari. Diffusion models are secretly exchangeable: Parallelizing DDPMs via auto speculation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24270–24289. PMLR, 2025.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 2023.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28935–28948. PMLR, 2024.

Fuliang Liu, Xue Li, Ketai Zhao, Yinxi Gao, Ziyan Zhou, Zhonghui Zhang, Zhibin Wang, Wanchun Dou, Sheng Zhong, and Chen Tian. DART: Diffusion-inspired speculative decoding for fast LLM inference. *arXiv preprint arXiv:2601.19278*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Nikil Roashan Selvam, Amil Merchant, and Stefano Ermon. Self-refining diffusion samplers: Enabling parallelization via parareal iterations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 5429–5453. Curran Associates, Inc., 2024. doi: 10.52202/079017-0176.

Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 4263–4276, 2023.

Alexander Soen, Hisham Husain, Valentin De Bortoli, and Arnaud Doucet. Accelerating speculative diffusions via block verification. *arXiv preprint arXiv:2606.13426*, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. In *Advances in Neural Information Processing Systems*, volume 31, pp. 10086–10095, 2018.

Zheng Wang, Davis Wertheimer, Yu Chin Fabian Lim, Mudhakar Srivatsa, Raghu K. Ganti, Minjia Zhang, and Naigang Wang. xPress: Parallel refinement for diffusion drafters in speculative decoding. *arXiv preprint arXiv:2608.02438*, 2026.

Xinwan Wen, Bowen Li, Jiajun Luo, Ye Li, and Zhi Wang. FREE: Uncertainty-aware autoregression for parallel diffusion transformers. *arXiv preprint arXiv:2511.20390*, 2025.

Zilin Xiao, Hongming Zhang, Tao Ge, Siru Ouyang, Vicente Ordonez, and Dong Yu. ParallelSpec: Parallel drafter for efficient speculative decoding. *arXiv preprint arXiv:2410.05589*, 2024. doi: 10.48550/arXiv.2410.05589.

Beier Zhu, Ruoyu Wang, Tong Zhao, Hanwang Zhang, and Chi Zhang. Distilling parallel gradients for fast ODE solvers of diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19557–19566, 2025.

A EXACTNESS AND ACCEPTANCE ANALYSIS

A.1 PROPOSAL FILTRATION

Lemma 1 (Causal validity of packed re-encoding)**.** *For every proposal position k, the refined prediction $\widehat{\epsilon}_k^1$ in equation 5 is $\mathcal{F}_{k-1}$-measurable. Consequently the proposal mean μ_k^q in equation 6 is a function of the legal prefix and does not read ξ_k.*

Condition on the complete history at the start of a round, including anchor x_{anc}, condition c, feature cache a, weights, and schedule. Draw independent standard Gaussian innovations $\xi_{1:L}$ and independent uniform variables $u_{1:L}$, with the two collections independent of each other and of that history. Use $\mathcal{F}_k$ defined in Section 2.2 for proposal dependencies.

The anchor-conditioned coarse outputs $\widehat{\epsilon}_j^0$ are $\mathcal{F}_0$-measurable. By induction in equation 4, x_j^0 is $\mathcal{F}_j$-measurable: its scheduler mean reads x_{j-1}^0 and a fixed coarse prediction, then the transition adds only ξ_j. Hence source $s_k = x_{k-1}^0$ is $\mathcal{F}_{k-1}$-measurable.

The packed rerun is a product map across packed items,

$$H(s_1, \dots, s_L) = \big(h_{W+\Delta W_{\mathrm{role}}}(s_1, t_1, c, a), \dots, h_{W+\Delta W_{\mathrm{role}}}(s_L, t_L, c, a)\big). \tag{12}$$

Its attention and normalization operate within each packed item, and context-cache expansion repeats fixed a; there is no cross-item batch statistic. With both models in evaluation mode, component $\widehat{\epsilon}_k^1$ is a deterministic function of s_k, c, a and is $\mathcal{F}_{k-1}$-measurable even though all components execute together.

For the rebuilt chain, assume x_{k-1}^q is $\mathcal{F}_{k-1}$-measurable. Equation 6 then makes μ_k^q $\mathcal{F}_{k-1}$-measurable and $x_k^q = \mu_k^q + \sigma_k \xi_k$ $\mathcal{F}_k$-measurable. The base case is $x_0^q = x_{\mathrm{anc}}$. This proves Lemma 1. Drawing the full innovation tensor before neural execution is immaterial: measurability concerns which components a map reads, not when storage is allocated.

Our evaluated construction shares innovations between the coarse and rebuilt chains. The measurability argument also accommodates independent auxiliary coarse randomness by including it in an augmented initial history: rebuilt innovations and verifier uniforms must then be mutually independent and independent of that history, and each proposal mean must be measurable from that history and preceding rebuilt innovations. This defines a distinct proposal construction with the same density-recording requirements. In either construction, reflection uses the actual candidate's innovation and recorded generating mean.

A.2 ONE-STEP COUPLING IDENTITY

Fix the complete legal internal history before a stochastic transition and abbreviate the conditional proposal and target densities by q and p. Maximal coupling contributes accepted mass $\min\{p(x), q(x)\}\mathrm{d}x$. The rejection probability is

$$\rho = 1 - \int \min\{p(x), q(x)\}\mathrm{d}x = \int (p(x) - q(x))^+ \mathrm{d}x, \tag{13}$$

and the residual density is $(p-q)^+/\rho$. Multiplying that density by the rejection probability cancels the normalizer. Therefore, for any measurable set A,

$$\Pr[X \in A] = \int_A \min\{p(x), q(x)\}\mathrm{d}x + \int_A (p(x) - q(x))^+ \mathrm{d}x \tag{14}$$

$$= \int_A p(x)\mathrm{d}x. \tag{15}$$

For equal-covariance Gaussians, sample-level reflection coupling realizes this residual without generic rejection sampling (De Bortoli et al., 2025).

A.3 SEQUENTIAL INDUCTION

To include earlier verifier decisions, define $\mathcal{G}_{k-1} = \mathcal{F}_{k-1} \vee \sigma(u_1, \dots, u_{k-1})$. Survival to position k is measurable with respect to this history. The independence assumption leaves ξ_k standard Gaussian

and u_k uniform after conditioning on $\mathcal{G}_{k-1}$ and survival. Thus the recorded q_k is still the proposal density used in the one-step identity.

Packed target scoring computes $\mu_k^p(x_{k-1}^q, c)$ itemwise and does not feed these values into the proposal. At every reached position, the accepted mass and reflected residual together give p_k. Conditional on rejection alone, the corrected state has residual density $(p_k - q_k)^+/\rho_k$, not density p_k. Discarding the suffix and restarting with fresh randomness preserves the next-step target kernel. The outer loop's anchor transition is itself a target step. Induction over the concatenated diffusion indices therefore proves equality of every fixed-length path prefix; it makes no claim of an unchanged path law after conditioning on a random accepted length.

The proof requires the densities in equation 3 to match the transitions actually sampled. Relaxation, mismatched variance, cross-item neural operations, current/future-innovation access, or an approximate residual sampler each changes the argument and requires a separate bound or theorem.

A.4 Zero-variance terminal transition

When $\sigma_k = 0$, the Gaussian density ratio with respect to Lebesgue measure is not defined. The terminal target kernel is instead the point mass $\delta_{\mu_k^p(x_{k-1},c)}$. The implementation treats this transition separately; target-law preservation requires committing the target point-mass update rather than applying the positive-variance reflection formula. Proposition 1 covers the stochastic transitions and includes this deterministic target update by composition.

A.5 From local overlap to realized acceptance

Consider a round containing L stochastic proposal transitions with $\sigma_k > 0$. At a reached position, condition on its legal history $\mathcal{G}_{k-1}$. Integrating the sample-level acceptance rule over the next proposal gives

$$o_k^{\text{live}} = \int q_k(x) \min\{1, p_k(x)/q_k(x)\} \, \mathrm{d}x = 2\Phi\left(-\frac{\|\mu_k^q - \mu_k^p\|_2}{2\sigma_k}\right). \tag{16}$$

Here the target mean is evaluated at the rebuilt proposal source, not a teacher-trajectory source. Let J_k indicate acceptance of the first k proposals, with $J_0 = 1$. For a round of fixed horizon L, the tower property gives

$$\mathbb{E}[A] = \sum_{k=1}^{L} \Pr(J_k = 1) = \sum_{k=1}^{L} \mathbb{E}[J_{k-1} o_k^{\text{live}}]. \tag{17}$$

Equation 17 accounts jointly for local overlap and prefix survival on the random visited histories. In contrast, equation 11 evaluates source interventions on fixed target trajectories. The latter isolates input-state sensitivity; deployed AL measures the full proposal process, including its visited states and survival events. A live calibration would pair logged o_k^{live} with acceptance indicators at reached positions and cluster uncertainty by sampling case; the reported source-intervention diagnostic uses the fixed-trajectory estimand.

A.6 Source-state error and conditional acceptance

Proof of Proposition 2. Fix a reached position and condition on $\mathcal{G}_{k-1}$ from Appendix A.3. This history contains the preceding innovations and verifier decisions, but not ξ_k or u_k. Lemma 1 makes s_k, x_{k-1}^q, and $\widehat{\epsilon}_k^1$ fixed under this conditioning. The next innovation remains standard Gaussian and the verifier uniform remains independent. Thus q_k and p_k have the recorded means and common covariance $\sigma_k^2 I$, with $\sigma_k > 0$.

Write $z = x_{k-1}^q$. The required regularity inequalities are $\|f_k(s_k) - f_k(z)\|_2 \le K_k \|s_k - z\|_2$ and $\|\mathcal{M}_k(z,u) - \mathcal{M}_k(z,v)\|_2 \le B_k \|u - v\|_2$ for the predictions used below. The two means use the

same rebuilt source z, so no scheduler state-error term is needed at this step:

$$\begin{aligned}\|\mu_k^q - \mu_k^p\|_2 &= \|\mathcal{M}_k(z, \widehat{\epsilon}_k^1) - \mathcal{M}_k(z, f_k(z))\|_2 \\ &\le B_k \|\widehat{\epsilon}_k^1 - f_k(z)\|_2 \\ &\le B_k\big(\|\widehat{\epsilon}_k^1 - f_k(s_k)\|_2 + \|f_k(s_k) - f_k(z)\|_2\big) \\ &\le B_k(\eta_k + K_k\delta_k). \end{aligned} \tag{18}$$

For completeness, let $r = \|\mu_k^q - \mu_k^p\|_2/\sigma_k$. If $r > 0$, the half-space $\{p_k \ge q_k\}$ has q_k-probability $\Phi(-r/2)$, and its complement has the same p_k-probability. Hence integrating $\min\{p_k, q_k\}$ gives $2\Phi(-r/2)$, as in equation 16; when $r = 0$, both densities coincide and the overlap is one. Since this expression decreases with r, substituting equation 18 proves equation 10. The result concerns the conditional mean of the acceptance rule, not a pointwise lower bound on $a_k(x_k^q)$ for every sampled candidate.

Displacement under shared innovations. Assume additionally that $\mathcal{M}_k(\cdot, e)$ is A_k-Lipschitz in its state argument and that the preceding B_k bound covers the coarse and refined predictions. For $k < L$, subtracting equation 4 and equation 6 cancels the identical $\sigma_k\xi_k$ terms. Adding and subtracting $\mathcal{M}_k(x_{k-1}^q, \widehat{\epsilon}_k^0)$ gives

$$\delta_1 = 0, \qquad \delta_{k+1} \le A_k\delta_k + B_k\|\widehat{\epsilon}_k^0 - \widehat{\epsilon}_k^1\|_2, \quad k = 1, \dots, L-1. \tag{19}$$

This stability recursion separates propagated state displacement from the coarse/refined prediction difference. For an affine scheduler $\mathcal{M}_k(x, e) = a_k^{\mathrm{sch}}x + b_k^{\mathrm{sch}}e$, one may take $A_k = |a_k^{\mathrm{sch}}|$ and $B_k = |b_k^{\mathrm{sch}}|$. The target regularity constant K_k remains a theoretical assumption; contraction would additionally require control of both terms in equation 19.

Consequence for accepted length. Fix the round-start history and a horizon of L positive-variance transitions. If constants $r_k \ge 0$, fixed given that initial history, satisfy

$$\frac{B_k(\eta_k + K_k\delta_k)}{\sigma_k} \le r_k \quad \text{almost surely on every surviving prefix at position } k, \tag{20}$$

then $\ell_k = 2\Phi(-r_k/2)$ is a uniform conditional acceptance lower bound. With J_k from equation 17, the tower property gives $\mathbb{E}[J_k] = \mathbb{E}[J_{k-1}o_k^{\mathrm{live}}] \ge \ell_k\mathbb{E}[J_{k-1}]$. Induction yields

$$\mathbb{E}[A] \ge \sum_{k=1}^{L}\prod_{j=1}^{k} 2\Phi(-r_j/2), \tag{21}$$

where expectations remain conditional on the round-start history. No independence of acceptance events is required. The uniform condition cannot be replaced by an average training MSE, unconditional marginal acceptance rates, or overlaps on fixed teacher trajectories.

Guidance and interpretation. All bounds use $f_k = \epsilon_w^p$, the noise field actually used by target verification. If f_k^{raw} denotes the target's raw conditional prediction, then

$$\eta_k \le \|\widehat{\epsilon}_k^1 - f_k^{\mathrm{raw}}(s_k)\|_2 + \|f_k^{\mathrm{raw}}(s_k) - f_k(s_k)\|_2. \tag{22}$$

The $w = 1$ loss supervises the raw prediction term in Equation 22 on training rollouts. At a fixed encoded source and rerun prediction, write $e_k = \widehat{\epsilon}_k^1 - f_k^{\mathrm{raw}}(s_k)$ and $g_k = \Pi_3(\epsilon_{\mathrm{cond}}^p(s_k, t_k, y) - \epsilon_{\mathrm{uncond}}^p(s_k, t_k))$. Then $\eta_k = \|e_k - (w-1)g_k\|_2$, and the guidance-discrepancy term is $|w-1|\,\|g_k\|_2$. The two error vectors can reinforce or cancel; training at $w = 1$ alone therefore does not determine η_k at deployment. Together with source displacement and the surviving-history condition in equation 20, this decomposition connects the training objective to the acceptance bound. The verifier still targets the deployed guided transition law under Proposition 1. The bounds characterize a given proposal construction; comparisons between trained drafters are evaluated empirically. Zero-variance terminal updates are handled separately as specified above.

A.7 UNCHANGED STRICT VERIFIER

At the first rejection, reflection coupling samples the normalized residual $[p_k - q_k]_+$, where $[z]_+ = \max\{z, 0\}$, using the rejected proposal's innovation (De Bortoli et al., 2025):

$$x_k^{\mathrm{corr}} = \mu_k^p + \sigma_k(I - 2e_k e_k^\top)\xi_k, \tag{23}$$

Here $e_k = (\mu_k^q - \mu_k^p)/\|\mu_k^q - \mu_k^p\|_2$. Equal means imply certain acceptance. A zero-variance terminal transition instead commits the deterministic target update.

Algorithm 2 Strict sample/reflection verification, adapted from De Bortoli et al. (2025); packed target scoring follows Wen et al. (2025).

Input: proposal states $x_{0:L}^q$, recorded proposal means $\mu_{1:L}^q$, timesteps $t_{1:L}$, condition c, variances $\sigma_{1:L}^2$, scheduler $\mathcal{M}_{1:L}$, and target model.

1 **Set** $\widehat{\epsilon}_{1:L}^p \leftarrow \textsc{Target}(x_{0:L-1}^q, t_{1:L}, c)$ by itemwise packed scoring of rebuilt sources, not coarse $s_{1:L}$.
2 **for** $k = 1, \ldots, L$ **do**
3 **Set** $\mu_k^p \leftarrow \mathcal{M}_k(x_{k-1}^q, \widehat{\epsilon}_k^p)$.
4 **if** $\mu_k^q = \mu_k^p$ **then continue** to the next position.
5 **if** $\sigma_k = 0$ **then**
6 **return** $(\mu_k^p, k-1)$ using the deterministic target point-mass update.
7 **end if**
8 **Sample** $u_k \sim \mathrm{Uniform}(0,1)$ independently of all innovations and earlier uniforms.
9 **Set** accepted $\leftarrow \mathbb{I}\Big[u_k \le \min\Big(1, \frac{\mathcal{N}(x_k^q; \mu_k^p, \sigma_k^2 I)}{\mathcal{N}(x_k^q; \mu_k^q, \sigma_k^2 I)}\Big)\Big]$.
10 **if not** accepted **then**
11 **Set** $e_k \leftarrow (\mu_k^q - \mu_k^p)/\|\mu_k^q - \mu_k^p\|_2$.
12 **Set** $x_k^{\mathrm{corr}} \leftarrow \mu_k^p + (I - 2e_k e_k^\top)(x_k^q - \mu_k^q)$.
13 **return** $(x_k^{\mathrm{corr}}, k-1)$.
14 **end if**
15 **end for**
16 **return** (x_L^q, L).

Algorithm 1 retains each candidate, generating mean, and variance. For $\sigma_k > 0$, the residual $x_k^q - \mu_k^q = \sigma_k \xi_k$ in exact arithmetic, so line 12 is the same reflection rule without an explicit noise-tensor argument. The verifier neither reconstructs proposal means from coarse sources nor draws fresh noise for reflection. Its target evaluation reads each rebuilt source, timestep, and c, not the drafter's feature cache or adapter parameters.

A.8 SCOPE AND VERIFIER COMPATIBILITY

Other transition kernels. The argument covers timestep-varying positive schedules σ_k when proposal and target share the same prescribed covariance at each transition; the experiments instantiate this condition with fixed-posterior DDPM. Learned state-dependent or mismatched covariances require a separately verified density and residual construction. Deterministic ODE steps have point-mass kernels, with zero overlap when proposal and target updates differ, and therefore require a different acceleration interface.

Block and relaxed verification. Block verification uses joint prefix acceptance and a weighted rejection residual sampled by Decomposition (Soen et al., 2026). A candidate RELOAD interface would supply rebuilt source states, timesteps, and the recorded generating means and variances. It would retain the coarse/rebuilt internal history when defining proposal conditionals, then replace both the prefix decisions and reflection correction in Algorithm 2. Establishing exactness requires checking the block acceptance and residual identities under this augmented history; the present reflection proof does not establish that composition. The present experiments isolate drafter improvements under reflection verification. Whether these gains combine with block verification remains untested; a drafter–verifier factorial comparison would measure the interaction. Uncertainty-relaxed FREE (Wen et al., 2025) instead changes the quality–latency trade-off by relaxing verification. The reported experiments use strict sample-level reflection.

Other accelerators. SRDS (Selvam et al., 2024) and EPD-Solver (Zhu et al., 2025) refine trajectories in parallel. ParaDiGMS uses Picard iterations to refine several diffusion steps in parallel (Shih et al., 2023); AsyncDiff pipelines denoising components across devices using hidden states from preceding steps (Chen et al., 2024). Their convergence tolerance or stale-state approximation controls differ from reflection's preservation of a specified chain law. A target-side approximation or solver change defines a new designated kernel whose preservation must be assessed separately. Drafter-side changes, including refinements inspired by xPress (Wang et al., 2026), can use the present proposition if they preserve causal Gaussian proposals, matched covariances, and recorded generating densities. A recurrent rerun head would introduce a neural dependency across positions, so its effect on critical-path latency would need measurement even if the head were small.

B ROLE-LORA PARAMETER ACCOUNTING

The rerun drafter has one transformer block, hidden width $D = 1152$, and MLP width $4D$. A rank-r update UV on a linear map $d_{\text{in}} \to d_{\text{out}}$ contributes $r(d_{\text{in}} + d_{\text{out}})$ parameters; biases remain in the frozen base map.

Table 5: Adapter placement and parameter count for uniform rank eight.

Rerun module	Shape	Symbolic count	$r = 8$
Context projection	$5D \to D$	$6Dr$	55,296
Attention QKV	$D \to 3D$	$4Dr$	36,864
Attention output	$D \to D$	$2Dr$	18,432
MLP fc1	$D \to 4D$	$5Dr$	46,080
MLP fc2	$4D \to D$	$5Dr$	46,080
AdaLN modulation	$D \to 6D$	$7Dr$	64,512
Total	six maps	$29Dr$	267,264

Ranks 4, 8, and 16 therefore use 133,632, 267,264, and 534,528 trainable parameters. We use $\alpha = r = 8$, so the multiplier α/r is one. The compact drafter contains 30,832,256 parameters; the rank-eight update adds 0.867% to this count. The selected checkpoint and runtime use the placement in Table 5.

At BF16, the unmerged adapter factors require $2 \times 267{,}264 = 534{,}528$ bytes (0.510 MiB) of parameter storage. Merging replaces every wrapped layer by

$$W_{\text{merged}} = W + \frac{\alpha}{r} UV. \tag{24}$$

The timed hot path uses merged dense weights, eliminating separate adapter kernels. Deployment stores two dense drafters and the activations for an LB-item rerun. The raw target cache contains $5BND$ scalars, and each role's K/V contains $2BND$ before horizon expansion. Table 28 measures single-method PyTorch allocated and reserved memory; Appendix C.5 separately records the primary benchmark's co-resident-process peaks.

B.1 PACKED-RERUN COMPUTE AND MEMORY SCALING

For one dense transformer block, N image tokens, width D, and a fixed MLP expansion ratio, linear projections and attention require $O(ND^2)$ and $O(N^2D)$ work per item, respectively. Building context once at batch B and rerunning LB items therefore costs

$$C_{\text{rerun}} = O\big(BND^2 + LB(ND^2 + N^2D)\big)\,. \tag{25}$$

This excludes the coarse pass, target scoring, scheduler, and communication. With n_h attention heads, conventional materialized attention uses $O(LBND + LBn_hN^2)$ activation storage, in addition to weights and the $O(BND)$ raw/projected cache. Expanded K/V can add $O(LBND)$ storage if materialized. Memory-efficient attention can avoid storing the full attention matrix, but does not remove its quadratic arithmetic. Kernel workspaces and tensor lifetimes also affect peak memory; Tables 24 and 28 report measured peaks.

With the same VAE downsampling and patch size, the analytical scaling from 256^2 to 512^2 quadruples N: projection work grows fourfold and dense-attention work sixteenfold at fixed B, L, D. Packing batches the itemwise neural evaluations into one invocation while retaining their arithmetic. Horizon selection consequently balances accepted-prefix length against packed execution cost, as measured at ImageNet-256 in Table 22.

C SHARED EXPERIMENTAL PROTOCOLS

C.1 EXPERIMENT AND CHECKPOINT MAP

Primary Role7000 is the rank-eight step-7000 EMA used in the primary comparison. **Follow-up Rank8** is a different rank-eight step-7000 EMA from a separate adaptation run. The frozen Stage-I base and independently trained strict FREE baseline are distinct step-180120 EMA checkpoints. The named checkpoints distinguish training runs, not just ranks or training steps.

Table 6 groups comparisons by checkpoint and case bank. Main and core-component results come from separate executions on the same 256 cases. The seven-method, placement, CFG-channel, and noise studies similarly share Final128. Comparisons within each group therefore reuse test cases.

Table 6: Experiment-to-evidence map. Timing is compared within each run: *formal* denotes the reported matched latency comparison; *diagnostic* denotes supporting timers; *excluded* denotes timers not used in the paper. Study-specific protocols accompany the results in Appendix E.

Study and result entry	Predictor/checkpoint	Evaluation cases	Timing use
Primary latency (Table 1)	Primary Role7000; strict FREE	Locked 256 test cases	Formal, eight ranks
Training-free control (Table 11)	Primary Role7000; strict FREE; carried score	Separate locked 128-case bank	Formal, eight ranks
Source-input intervention (Figure 2; Table 12)	Frozen base and Primary Role7000, each held fixed	Same 1,024 training-disjoint anchor pairs	Proxy only
Core components (Tables 2, 14)	Frozen base; Primary Role7000; separate raw-step-250 pilot	Same locked 256 cases	One-GPU component timing
Adapter holdout (Table 15)	Frozen base vs. Primary Role7000	Disjoint 64-case test bank	AL/calls only
Matched adaptation (Table 16)	Follow-up Rank8; selected full and conditioning-only reruns	Separate locked 128; 64 selection cases	One-GPU sampling timing
Rank/weight/refiner comparison (Table 18)	Follow-up Rank8 reference; variant-selected checkpoints	Final128; separate 64-case selection	Excluded: GPU interference
Placement (Table 20)	Named conditions; per-row checkpoint identities unavailable	Same Final128; separate selection cases	Diagnostic only
Training-step sweep (Table 19(c))	Primary training trajectory, including Primary Role7000	64 selection cases, not Final128	Excluded
CFG channels and shared noise (Tables 26, 21)	Same Follow-up Rank8 file	Same Final128, separate runs	Diagnostic only
Horizon, CFG scales, and hosts (Appendix E.8)	Primary Role7000; strict FREE	64 selection/192 test; CFG/host subsets overlap	Formal within each run

The matched-adaptation comparison reuses the Follow-up Rank8 file on a separate 128-case test bank, disjoint from both its 64-case selection bank and the earlier Final128 bank; Appendix E.5

gives its training and timing protocol. Its training-seed extension adds two seeds for Role-LoRA and conditioning-only on these same selection and test banks, reporting AL and calls (Table 17). The placement study uses its own all-six reference (AL 4.7604), rather than the Final128 seven-method reference (AL 4.7851). Its aggregate record does not identify the loaded checkpoint for each row, so we keep these references separate. The training-step sweep, in contrast, records the same step-7000 file as Primary Role7000; its 64-case selection bank is not an independent test bank.

The reproducibility manifest identifies the executed source (including local modifications), target and drafter checkpoints and branches, configuration and case bank, hardware/software and numerical mode, method order, and timing context. Supplementary campaigns retain different subsets, as stated in Appendix E.8; the full field schema accompanies the reproduction files.

Reported statistics are computed from the retained machine-readable measurements.

C.2 Training configuration and checkpoint selection

Training manifests retain data/subset and preprocessing identities, effective optimization and horizon-supervision settings, selected checkpoint/EMA branches, numerical mode, and the trainable/detached parameter contract. Table 7 gives the recipe needed to reproduce the two stages.

Table 7: Two-stage training recipe. "Base" denotes the horizon-parallel DFlash drafter; "Role" denotes rerun-only specialization after the base EMA weights are copied into both inference roles.

Field	Stage I: base drafter	Stage II: Role-LoRA
Data subset	30% of ImageNet train; permutation seed 0	Separately seeded 30%; seed 20260818
Preprocessing	Resize 256, center crop, random horizontal flip, normalize to $[-1, 1]$, frozen `sd-vae-ft-mse` VAE with latent scale 0.18215	Identical image-to-latent pipeline
Training anchors	$\tau \sim \mathrm{Unif}\{8, \ldots, 999\}$; forward-noised latent followed by one frozen-target transition	Same anchor construction; frozen base drafter constructs detached coarse sources
Initialization	Frozen target image/time/class embeddings and target-initialized output projection; newly instantiated draft block and context path; timestep residual up/bias initialized to zero	Base EMA copied into coarse and rerun roles; V Kaiming-uniform and $U = 0$ in every rank-8 adapter
Trainable state	Draft block, context projection/norm, and rank-16 absolute-timestep heads; 30,832,256 parameters	Six adapter pairs in the rerun copy only; 267,264 parameters
Supervision	All eight target-noise references on detached on-policy base rollouts; Equation 7	All eight target-noise references at detached coarse sources; Equation 7
Optimization	AdamW, constant 10^{-4}, weight decay 0, default $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$; no warm-up or finite-norm clipping	Same optimizer contract
Batch and budget	Global batch 256 on four A100-80GB GPUs; 120 epochs, 180,120 updates	Global batch 16 on one A100-80GB GPU; maximum 8,000 updates
Numerics and EMA	BF16 autocast, TF32 matmul, EMA 0.9999; final step-180120 EMA	BF16 autocast, TF32 matmul, EMA 0.99; selected step-7000 EMA

Both stages use label dropout 0.1 and normalized horizon weights $(0.24, 0.19, 0.15, 0.12, 0.10, 0.08, 0.07, 0.05)$; feature and smoothness loss weights are zero. The Stage-I base and the independently trained strict FREE180 baseline are distinct EMA checkpoints, both taken at optimizer step 180120.

For Stage II, 26 registered candidate steps span 200–8000. The locked 64-case selection bank chooses by paired mean strict target forwards (lower is better), with paired mean strict AL as the

tie-break (higher is better). Step 7000 EMA was locked before opening the disjoint final bank. The final evaluation bank is prohibited from reselection.

The primary 256-case evaluation bank has namespace `reload-u2-h8-final-v1-20260826`, sets `selection_permitted=false`, and has zero measured overlap with the Role-LoRA selection and confirmatory banks. Each raw row stores the realized seed, class label, and rotated method order.

The later 16-case pilot in Table 29 tracks five Role-LoRA checkpoints from one training run. The primary study retains its preselected checkpoint.

C.3 Training cost and amortization

Training ReLoaD from scratch requires both stages. The component comparisons share Stage I, while strict FREE has its own training cost. For an existing Stage-I drafter, the incremental training cost is Stage II: fitting 267,264 adapter parameters while also executing the frozen target, coarse drafter, and rerun backbone. We account for the full run through step 8000, including updates after the selected step-7000 EMA.

The original Stage-II log records 8,000 measured updates in 3,603.791 seconds on one A100-80GB GPU, or 1.001 GPU-hours. The timed training loop includes intermediate checkpoint saves and excludes initialization, the final save, and checkpoint-selection evaluations. Stage I spans interrupted and resumed runs; the recovered records do not establish its total GPU-hours. We therefore keep total upfront cost as an input to the amortization analysis; Table 7 gives both stages' update counts and allocations.

For a reference F and ReLoaD R, let t_F, t_R denote per-image sampler seconds and g_F, g_R their allocated GPU counts on the same hardware. For a positive additional upfront cost C_{train} in GPU-hours under an explicitly stated accounting scope, the corresponding amortization threshold is

$$N_{\text{break}} = \frac{3600\, C_{\text{train}}}{g_F t_F - g_R t_R}, \qquad g_F t_F - g_R t_R > 0. \tag{26}$$

Here, switching from an existing FREE checkpoint charges both ReLoaD training stages and its selection cost; adapting an existing frozen rerun charges Stage II and selection against the frozen-rerun timing. Comparing both pipelines from scratch instead uses their measured upfront cost difference, including the independently trained FREE baseline. A positive upfront cost has no finite break-even count when the denominator is nonpositive. GPU-hours from different accelerator types cannot be combined without an explicit conversion model.

For the primary eight-A100 workload, $t_F = 6.092090\,\text{s}$ and $t_R = 5.482338\,\text{s}$ give 4.878013 allocated GPU-seconds saved per image. Thus $N_{\text{break}} \approx 738.0\, C_{\text{train}}$, with the upfront GPU-hour cost supplied by the deployment scenario. This estimates compute amortization for the measured sampling workload; monetary savings additionally depend on resource pricing.

C.4 Runtime protocol fields

Runtime readbacks and execution hooks jointly check verifier mode, horizon and shared-noise policy, terminal-coarse omission, observed head count and rerun batch size, merged adapters, context caching, and rankwise verification batches.

Runs are rejected when requested protocol fields and runtime readback disagree. The deployed $L = 8$ path must show one coarse invocation with seven useful outputs and one packed rerun over eight causal sources.

C.5 Timing and statistics

Within the primary and allocation campaigns, Target, FREE, and ReLoaD run on the same host in a balanced rotating order. Paired timed rows share the initial-noise seed and class label, synchronize CUDA before and after sampling, and store method order. At least one warm-up cycle precedes measured rows. Sample counts and exclusion rules are fixed before outcomes are inspected. Study-specific runners are described alongside their results in Appendix E; their raw row sequence preserves execution order without a dedicated per-row order field.

Table 8: Recorded systems for the two timing campaigns. System labels anonymize distinct hosts; GPU and software fields are runtime readbacks.

System	Campaign (paired cases)	Allocated ranks	GPU	PyTorch	CUDA
A	Main (256)	8	A100-SXM4-80GB	2.11.0+cu128	12.8
B	Allocation (32)	1/2/4/8	A100-SXM4-80GB	2.11.0+cu128	12.8

Both campaigns enable TF32 with `high` float32 matmul precision and use the same target, base-drafter, strict-FREE, and Role-LoRA checkpoints, but different evaluation banks. Clock and power settings were not recorded; host and bank differences therefore remain confounded in cross-campaign absolute timings.

For paired wall times a_i and b_i, speedup is $\bar{a}/\bar{b}$. The paired bootstrap resamples case indices, not methods independently; the primary 256-case campaign uses 50,000 resamples with seed 20260831. AL and target-call changes use paired per-case differences; pooled acceptance is recomputed within each resample.

For case i, strict AL is the number of accepted draft transitions divided by the number of speculative rounds. Corrected transitions and target-only anchor transitions are excluded from the numerator; terminal-only rounds are excluded from the denominator. Reported AL averages these per-case ratios. Acceptance divides accepted draft transitions by all proposed draft positions, including discarded suffix proposals. The primary and holdout tables pool these counts over cases; Table 14 instead averages per-case acceptance ratios.

For distributed target verification, Calls (`target_forwards`) counts logical target invocations, including anchor evaluations, packed verification calls, and any terminal target-only call. A packed call counts once regardless of its batch size or rank allocation. Rows separately record per-rank calls, packed batch sizes, sample-forwards, GPU-seconds, communication time, and peak memory after warm-up.

For a row with wall time T and G allocated ranks, allocated GPU-seconds are GT. Active-rank GPU-seconds are $G_{\mathrm{act}}T$, where G_{act} counts ranks with at least one measured target-model invocation; it is distinct from the accepted-prefix count A. These wall-time products include waiting; neither measures FLOPs or device busy time. In the primary campaign all methods occupy eight ranks, including the seven waiting ranks during TARGET execution, so TARGET has $G_{\mathrm{act}} = 1$, $G = 8$ and both speculative methods have $G_{\mathrm{act}} = G = 8$. Resource values in Table 9 are means over the same 256 measured rows per method; their aggregation checks per-rank counters and reproduces the primary mean latencies before generating the table values.

The retained communication telemetry measures host-observed blocking NCCL broadcast/all-gather duration without extra CUDA synchronization; all-gather can include rank waiting. Internal timing profiling was disabled, so these records do not provide an exclusive kernel or communication breakdown. All drafters remain resident on rank zero across the three-way comparisons. The largest per-rank allocated-memory peaks across the primary rows are 2.952 GiB for strict FREE and 3.050 GiB for RELOAD; these cover the resident process rather than an isolated deployment footprint for each method.

D IMPLEMENTATION AND VALIDATION

D.1 $L = 8$ EXECUTION

The source-level call graph for the optimized aligned path is:

1. produce seven coarse noise predictions in one batch-B backbone invocation;
2. apply seven affine scheduler prefixes with `include_endpoint=true`, producing eight states;
3. flatten $[B, 8, C, H, W]$ to $[8B, C, H, W]$;

4. build rerun-specific K/V from the unchanged parent target features once at B and expand it across the horizon;
5. invoke the rerun backbone once with local horizon one for every packed item;
6. reshape eight re-encoded noise predictions and serially record the rebuilt proposal chain;
7. stack proposal sources for one logical packed target-verification call.

The packed rerun evaluates all $8B$ aligned samples in one backbone invocation; each item retains its own computation.

The output-head bank pairs a shared rank-16 down-projection with up-projections and biases indexed by absolute diffusion timestep. Each $L = 8$ training example selects eight outputs from this bank. These Stage-I residual heads are distinct from the rank-eight Role-LoRA updates, and their parameters remain frozen during Stage II.

D.2 Itemwise execution contract

The packed-image axis remains separate from the token axis. Attention joins each item's local keys and cached context keys along the token axis, never across horizon items. LayerNorm and context RMSNorm reduce feature channels; AdaLN applies shifts and scales from the item's timestep and class. Target and drafter run in evaluation mode, with attention/projection dropout disabled and no inference label dropout. Target classifier-free guidance combines only conditional and unconditional predictions of the same source, not different horizon positions; the drafters use conditional predictions directly.

Each horizon item forms a separate batch element, so attention requires no horizon-causal mask. Expanded K/V repeats fixed per-image context. Normalization must remain itemwise: current packed-batch statistics would couple items, whereas fixed evaluation-time statistics can preserve independence. Fused kernels must preserve the same axes. We test causal dependencies through future-noise interventions and numerical sensitivity through packed/serial and TF32/FP32 replays. Floating-point rounding can change a decision even when execution remains itemwise.

Cache provenance. The optimized implementation builds role-specific K/V before invoking the rerun. It passes these tensors through `target_context_kv`, so the raw hidden-feature argument can be `None`. Both roles build their K/V from the same parent target hidden features. Their projected K/V tensors can differ because the rerun uses a role-adapted context projection.

An observer followed one complete trajectory at $L = 8$, CFG 1.5, and the step-7000 Role-LoRA checkpoint (seed 20260915031, class 317). It captured 151 parent caches and checked every call, retaining three example traces. All 151 coarse and all 151 rerun cache constructions consumed the corresponding parent target hidden features. All 151 packed rerun calls received the role-specific K/V built for that round, with zero mismatches. The eight inspected drafter dropout modules had zero dropout probability and ran in evaluation mode. The observer left the final sample, all 151 accepted-prefix lengths, and the 302 target forwards unchanged. Appendix E.9.2 reports the complementary layout and precision replays.

D.3 Guidance and recorded transition densities

The three-channel rule follows the released DiT `forward_with_cfg` implementation (Peebles & Xie, 2023). For $w \neq 1$, the target wrapper duplicates each source and timestep, evaluates its class and null labels together, and applies Γ_w from Section 3. On the four-channel latent, this gives $\epsilon^p_{w,j} = \epsilon^p_{\mathrm{uncond},j} + w(\epsilon^p_{\mathrm{cond},j} - \epsilon^p_{\mathrm{uncond},j})$ for $j = 1, 2, 3$, while $\epsilon^p_{w,4} = \epsilon^p_{\mathrm{cond},4}$. At $w = 1$ it evaluates only the supplied label. Cached hidden states and the carried feature come from the conditional branch. Target-only sampling, anchor transitions, and packed verification use this same target rule. The coarse and rerun calls receive the class label and that cache, without a guidance scale or an unconditional prediction; their returned noise fields directly define the coarse and proposal means.

The scheduler uses fixed posterior variance, ignoring the target's learned-variance outputs. For each proposal it samples from the computed transition statistics and stores those same mean and variance tensors with the candidate; reflection reads these stored values alongside the target mean computed

with Γ_w. The exact-arithmetic guarantee concerns this designated guided target kernel and the actual conditional proposal kernel. Replacing the stored proposal mean by a subsequently guided or recomputed prediction would invalidate the density used for correction. Guidance differences between proposal and target can affect acceptance, but do not violate the coupling assumptions when both densities are represented correctly.

For the tested configurations, Table 34 checks the channelwise guidance formula against the target's raw output, the conditional carried feature and drafter arguments, and the stored proposal statistics against scheduler samples. Every shifted-mean negative control is detected, while the observer preserves complete-run outputs and acceptance sequences. This check inspects the carried feature; the full parent-cache dataflow is tested separately above.

The channel ablation in Table 26 replaces Π_3 by the four-channel identity in both target anchor sampling and verification. At $w = 1$ the two formulas coincide; the 128 paired final samples and recorded acceptance counts also coincide. For $w > 1$, the fourth-channel change defines a different target kernel, while the proposal is still evaluated through its recorded generating density. Exactness applies separately to each designated kernel under the same covariance and causality assumptions.

D.4 Two-stage training execution

In both stages, each class label is replaced by the null label with probability 0.1. The resulting label is reused for the anchor transition, coarse rollout, rerun when present, and target references. Training target calls use guidance scale one, without classifier-free guidance amplification; the sampling guidance setting applies at evaluation. Thus the regression references are the target's conditional or null-label noise predictions at the detached training sources.

Stage I: base drafter. For weighted-all $L = 8$ supervision, the source code executes the following contract:

1. encode an ImageNet example with the frozen VAE, sample $\tau \in \{8, \dots, 999\}$, and forward-noise the clean latent to z_τ;
2. evaluate the frozen target at z_τ, capture hidden layers $\{1, 7, 13, 19, 25\}$, and sample one target transition to obtain the anchor $x_{\mathrm{anc}} = z_{\tau-1}$;
3. predict all eight future noise fields from that anchor in one drafter invocation;
4. build the eight target-query sources by rolling detached copies of the drafter outputs through the fixed stochastic scheduler;
5. evaluate the frozen target independently at every source, compute equation 7 with $v = \mathrm{base}$ using the original predictions with gradients and detached target references, update only the DFlash-owned parameters, and maintain EMA weights.

Stage II: Role-LoRA. The Stage-I EMA state is loaded before adapter installation and copied into the frozen coarse and rerun backbones. Training then

1. runs the frozen coarse drafter and builds all eight scheduler sources from detached predictions and states;
2. flattens the detached sources, fixed parent-evaluation feature cache, timesteps, and labels into one rerun batch;
3. evaluates the frozen target independently at every source to obtain detached reference noise predictions;
4. computes the normalized weighted per-position MSE in equation 7 with $v = \mathrm{role}$;
5. updates only the six pairs of Role-LoRA factors and maintains a separate EMA state.

Both stages minimize prediction MSE, with verifier acceptance measured at evaluation. The training records specify both configurations, the step-180120 base EMA and step-7000 role EMA, adapter placement and parameter counts, and the target, base drafter, Role-LoRA, and strict FREE checkpoints.

D.5 Regression tests

The regression suite tests affine prefix endpoints, terminal-coarse omission, aligned horizons, role-decoupled packed-versus-flat output, context-cache equivalence, future-noise causality, packed-versus-serial target means, reflection-coupling marginals, and verifier commit semantics. All 31/31 tests pass. For the deployed $L = 8$ path, 100 production and 100 FP32-strict future-noise interventions at anchor timestep 800 have zero causal failures; observed invocation counts and batch sizes match the runtime configuration (Table 32). These 200 cases form the relevant subset of the archive's 600 tests, which also cover Direct-H16 and RHE-H16.

All diffusion-sampling comparisons use the sample-level reflection verifier analyzed in Appendix A.

All 180 trials across two batch sizes, three horizons, and six timesteps preserve the protected causal prefix; Appendix E.9.1 reports packed–serial numerical discrepancies separately.

Appendix E.9.2 checks shared innovations and proposal/reflection reconstruction along eight full chains, with resampled-noise and swapped-index controls. Replaying saved inputs under four layouts and a precision toggle reveals one precision-induced decision difference. These fixed-chain replays measure local numerical sensitivity separately from causality.

D.6 Runtime validation

The primary benchmark records the loaded configuration, observed coarse outputs, rerun batch size, and invocation counts, rejecting mismatches between requested and observed execution. All primary timing runs pass this check. Appendix E.8.5 describes the pilots' more limited runtime records.

E Additional empirical results

The results follow the main experimental narrative. All image-generation studies use ImageNet-256 and DiT-XL/2; the known-law Gaussian chain is a separate synthetic implementation check. Unless stated otherwise, sampling uses $L = 8$ and sample-level reflection verification. Table 6 identifies checkpoints and case banks, and Appendix C.5 defines the common metrics and statistics. Study-specific protocols below state the available runtime and measurement records.

E.1 Primary comparison and baseline validation

Table 9 supplements the primary latency comparison with acceptance, calls, and GPU-second accounting.

Table 9: Supplementary statistics for the same 256 paired cases as Table 1. GPU-second entries are mean per-image wall-time products. All methods allocate eight ranks; an active rank makes at least one target-model invocation. Accept. pools accepted and proposed counts, and Calls counts logical target invocations.

Method	Active-rank GPU-s	Allocated GPU-s	Accept.↑	Calls↓
Target	12.535	100.280	–	1000.00
Strict FREE	48.737	48.737	57.05%	324.51
ReLoaD	43.859	43.859	**59.97%**	**315.29**

Baseline reproduction validation. The published results provide a scale reference across different class banks and systems. The independent local rerun agrees within 0.3% on the reported algorithmic metrics, supporting baseline repeatability. Headline speedups pair methods within the same local harness and case bank, with active-rank allocations reported explicitly.

E.2 Training-free carried-score comparison

The training-free control uses the target's guided noise prediction from the verified source whose transition produced the current committed state. An initial exact target transition bootstraps this

Table 10: Contract, scale, and repeatability checks for the locally reimplemented strict FREE. External values are from Wen et al. (2025) and are not measurements from our harness.

Check	Published FREE	Local strict FREE	Assessment
Target and data	DiT-XL/2, ImageNet-256	DiT-XL/2, ImageNet-256	corresponding setting
Transition	1,000-step fixed-posterior DDPM, CFG 1.5	same	matched
Training fields	30% subset, 120 epochs, AdamW 10^{-4}, EMA 0.9999	matches reported fields; local batch 256 disclosed	contract aligned
Speculation/verifier	$L = 8$, strict reflection coupling	same sample-level rule	matched
Parallel efficiency	2.680× (8 fixed classes)	3.082× (256 locked pairs)	scale check
Latency speedup	1.820× (20 runs)	2.058× (256 locked pairs)	cross-machine check
Independent local rerun	not applicable	PE 3.078×; AL 4.5245; accept. 56.88%; calls 324.86	within 0.3%
Run integrity	paper protocol	768/768 three-way rows; no foreign GPU PID	passed
Checkpoint identity	reference model	strict FREE step-180120 EMA	fixed local weights

Table 11: Training-free control: 128 paired ImageNet-256 cases, DiT-XL/2, 1,000 DDPM steps, $L = 8$, CFG 1.5. Eight ranks are allocated; TARGET activates one, speculative methods eight. Ratios use this campaign's mean times.

Method	Time (s)↓	AL↑	Calls↓	Speedup vs. Target↑	Speedup vs. FREE↑
TARGET	15.100	–	1000.00	1.000×	–
Strict FREE	7.706	4.584	322.32	1.960×	1.000×
RELOAD	**7.004**	**4.786**	**314.49**	**2.156×**	**1.100×**
Score reuse (training-free)	9.262	1.318	432.91	1.630×	0.832×

cache. The cached noise field is then held constant while the stochastic scheduler sequentially builds up to eight proposals. After strict verification, the score at the parent of the final committed state is carried into the next round, including when the final commit is a reflected correction. Thus the cache is one step behind the current state, and subsequent rounds require no fresh target anchor evaluation. Packed verification returns the noise fields needed for this update. This control has no learned drafter and uses the existing sample-level reflection verifier rather than block-level acceptance or correction.

ASD (Hu et al., 2025) evaluates the current state once per speculative round and reuses its denoised-data estimate in the DDPM proposal means. Our control instead carries the previous verified source's noise prediction. It tests score reuse under our sampler interface, not a reproduction of ASD.

The four-method campaign uses eight A100-SXM4-80GB GPUs, $B = 1$, $L = 8$, CFG 1.5, and the original 1,000-step fixed-posterior DDPM schedule. The learned methods retain the primary target, FREE180120 EMA, coarse180120 EMA, and Primary Role7000 EMA checkpoints. Inference uses FP32 with TF32 enabled and internal stage profiling disabled. The campaign evaluates 128 locked seed–class pairs, with eight excluded warm-up cases and four-way cyclic method order. All drafters remain resident on rank zero throughout the four-way comparison. We report timings within this campaign separately from the primary 256-case study. The retained monitor recorded 515 occupancy snapshots with no observed external GPU process.

All 128 score-reuse rows record zero drafter forwards, backbone samples, and head predictions; scheduler work, target evaluation, and communication remain included in latency. The 512 mea-

sured rows have complete four-way pairing. Per-case mean acceptance is 57.77% for strict FREE, 60.41% for RELOAD, and 16.57% for score reuse. Paired percentile intervals use 20,000 case resamples with seed 20260918. Relative to strict FREE, RELOAD increases AL by 0.2019 (95% CI $[0.1493, 0.2534]$) and reduces logical target calls by 7.83 ($[5.48, 10.16]$). Score reuse has a FREE-relative speedup of $0.832\times$ ($[0.825, 0.839]$), despite avoiding learned-drafter computation.

E.3 SOURCE-INPUT DIAGNOSTICS

Table 12 reports two source-only interventions. The first fixes the unadapted step-180120 DFlash EMA; the second fixes the Primary Role7000 predictor. Both use the same frozen base for coarse-source construction and the same ordered 1,024 image-index–anchor-timestep pairs. The ImageNet training corpus contains 1,281,167 images; both 30% training subsets are excluded before drawing this bank, and the recorded intersection with their union is zero. These two interventions reuse the same bank and are not 2,048 independent trajectories.

All B0/B1/B2 comparisons share target-trajectory scoring states, target context, diffusion timesteps, and transition innovations; only the state supplied to the local predictor changes. Within each predictor condition, position-one source states are checked to be bitwise identical. Both diagnostics run the target and drafter in FP32 with TF32 disabled, `highest` float32 matmul precision, and CFG 1.5. Paired prefix intervals use 50,000 bootstrap resamples with seed 20260826. The retained bundle includes all eight shard JSONs, per-case identities and metrics, the original summary, and a deterministic table-generation check. These measurements evaluate the analytic prefix proxy in equation 11 on fixed target trajectories.

Table 12: Source-only prefix diagnostics on the same 1,024 target trajectories, without and with role adaptation. Weights are fixed within each block; only representation inputs differ. All means use common target-trajectory scoring states. B2 is non-deployable; this proxy is not strict sampling AL.

Variant	Representation input	Prefix proxy, mean [95% CI]↑
Frozen DFlash180 EMA		
B0 anchor input	anchor at every horizon	$2.0946\ [2.0653, 2.1234]$
B1 coarse rerun	causal coarse source x^0_{k-1}	$5.0220\ [4.9189, 5.1244]$
B2 target-source reference	target source x^p_{k-1}	$5.0119\ [4.9082, 5.1152]$
Fixed Role7000		
B0 anchor input	anchor at every horizon	$2.1239\ [2.0946, 2.1526]$
B1 coarse rerun	causal coarse source x^0_{k-1}	$5.3674\ [5.2499, 5.4850]$
B2 target-source reference	target source x^p_{k-1}	$5.4246\ [5.3026, 5.5467]$

B1 improves the paired prefix proxy over B0 by 3.2435 (95% CI $[3.1365, 3.3513]$) and recovers 98.27% of the B0-to-B2 gain with Primary Role7000 weights fixed. This intervention isolates input-state sensitivity on target trajectories: B0 repeats the anchor as the rerun input, B1 supplies coarse sources, and B2 supplies target sources as a diagnostic reference. The repeated-anchor rerun is distinct from the deployment anchor one-pass baseline.

The frozen-base intervention confirms that the source effect precedes role adaptation: B1–B0 is 2.9273 (95% CI $[2.8360, 3.0192]$), and B1–B2 is 0.0100 (95% CI $[0.0081, 0.0119]$). The B1/B2 ratio is 1.0020 (99.5% CI $[1.0014, 1.0026]$). B0 in this block uses repeated-anchor local predictions from the frozen base; the separately measured deployment one-pass proxy is 2.0946 and is excluded from the source-only contrast.

E.3.1 CACHE-INPUT SENSITIVITY

The cache experiment holds weights and architecture fixed while perturbing cached inputs. Sixteen seed–class pairs at timesteps $\{900, 700, 500, 300, 100\}$ yield 80 anchor–timestep pairs for each of two rerun roles and ten interventions. Both roles use the same coarse sources from the unadapted base drafter and are scored against target predictions at those sources. The Gaussian-overlap formula in equation 11 is evaluated on these coarse sources, unlike the target-trajectory diagnostic in Section 4.3. Gaussian perturbations have standard deviation 0.01 or 0.05 times each feature tensor's RMS; layer removal replaces that layer's features with zeros.

Table 13: Cache-input sensitivity at $L = 8$ for the frozen base rerun and Role-LoRA rerun, with weights fixed. Each row/role averages 80 anchor–timestep pairs (1,600 records total) sharing coarse sources. MSE compares predicted noise with target predictions and averages positions; the overlap prefix proxy is not strict AL. Noise 0.01/0.05 denotes Gaussian standard deviation relative to feature RMS. No variant is retrained.

Input intervention	Frozen rerun MSE↓	Frozen rerun Proxy↑	Role-LoRA MSE↓	Role-LoRA Proxy↑
Intact	2.66×10^{-3}	5.1644	2.45×10^{-3}	5.6096
Zero all features	8.47×10^{-1}	0.0036	8.87×10^{-1}	0.0031
Shuffle spatial tokens	2.66×10^{-3}	5.1644	2.45×10^{-3}	5.6096
Zero layer 1	1.13×10^{-1}	1.4751	1.27×10^{-1}	1.2615
Zero layer 7	3.35×10^{-2}	1.5557	3.74×10^{-2}	1.3598
Zero layer 13	1.31×10^{-2}	3.0967	1.50×10^{-2}	2.8336
Zero layer 19	1.17×10^{-2}	3.3592	1.23×10^{-2}	3.4777
Zero layer 25	3.92×10^{-1}	0.2132	4.04×10^{-1}	0.2072
Noise 0.01	3.26×10^{-3}	4.9491	3.08×10^{-3}	5.3327
Noise 0.05	1.76×10^{-2}	3.3003	1.80×10^{-2}	3.3982

With intact context, role specialization raises the mean prefix proxy from 5.1644 to 5.6096 and lowers noise-prediction MSE. Zeroing all features collapses the proxy to about 0.003 for both roles; removing layer 25 causes the largest decline among the tested single-layer removals. Joint spatial-token permutation preserves feature content and leaves the aggregate values nearly unchanged. Together, zeroing and layer removal identify the cached inputs used by the fixed predictor.

E.4 Core components and independent adapter holdout

Table 14: Full component measurements underlying Table 2, using Primary Role7000 and the same 256 cases as the main comparison, in a separate one-GPU run with the strict verifier. Accept. averages per-case acceptance; the primary and holdout tables instead pool accepted and proposed counts.

Variant	Mean time (s)↓	AL↑	Accept.↑	Calls↓
Anchor one-pass	20.3256	2.0155	25.34%	499.35
Serial frozen rerun	15.5544	4.5284	57.11%	325.54
Packed frozen rerun	13.8242	4.5240	57.06%	325.76
Packed refiner (250-step pilot)	13.6737	4.5373	57.21%	324.66
ReLoaD	**13.3846**	**4.7703**	**60.18%**	**315.10**

The packed-refiner pilot uses a frozen parallel coarse drafter and a separate FREE one-step refiner. Each detached coarse source forms an independent batch item with the repeated anchor feature and its local timestep; attention remains itemwise. This exploratory control differs from Role-LoRA in architecture and training budget: it uses the raw `model` branch at step 250, versus Role-LoRA's EMA at step 7000. The same-backbone, matched-selection comparison of adaptation methods appears in Table 16.

E.4.1 Rerun-role specialization

Table 15: Rank-eight rerun-role specialization on a disjoint 64-case holdout. Only adapter factors differ; coarse-source construction, base weights, verifier, and cases are fixed. Resulting sampling trajectories may differ.

Rerun role	AL↑	Acceptance↑	Calls↓
Frozen packed rerun	4.5296	56.94%	325.42
Role-LoRA (step-7000 EMA)	**4.7627**	**59.88%**	**315.63**

The paired AL gain is 0.2331 (95% CI $[0.1547, 0.3085]$), and target calls decrease by 9.80 (95% CI $[6.16, 13.27]$), with positive AL differences on 48/64 cases. Candidate steps are evaluated only on the separate selection bank, and step 7000 EMA is fixed before this holdout is opened.

E.5 MATCHED RERUN-ADAPTATION COMPARISON

We compare frozen re-encoding, rank-eight Role-LoRA, full rerun fine-tuning, and conditioning-only tuning on the same Stage-I base. The coarse drafter remains frozen in all four conditions. Conditioning-only tuning updates the existing context-projection and block AdaLN parameters (14,605,056 parameters), rather than inserting adapters; it is distinct from removing conditioning adapters in Table 20.

The three adapted models share the base EMA initialization, 30% ImageNet training subset specification, learning rate 10^{-4}, batch size 16, weighted eight-position loss, EMA decay 0.99, and an 8,000-update maximum. Each is selected on the same independent 64-case bank from eight candidate updates: 200, 250, 500, 1000, 2000, 4000, 7000, and 8000. The selected EMA checkpoints are step 7000 for Role-LoRA (the Follow-up Rank8 file), step 1000 for full fine-tuning, and step 200 for conditioning-only. Thus the candidate budgets match, while the selected update counts differ. This original four-condition comparison uses training seed 20260818 and one learning rate; historical minibatch order is not established as identical across runs.

Table 16: Rerun-adaptation comparison on 128 paired test cases, $B = 1, L = 8$, CFG 1.5, and one GPU. Step denotes the selected EMA update. AL, sampler seconds, and logical target forwards are case means. Speedup is reference time divided by Role-LoRA time. Pointwise paired intervals condition on these checkpoints and quantify test-case variation, not variation across training seeds.

Adaptation	Step	AL↑	Time (s)↓	Calls↓
Frozen rerun	–	4.564	13.849	323.84
Role-LoRA, rank 8	7000	**4.818**	**13.381**	**313.13**
Full fine-tuning	1000	4.755	13.503	315.96
Conditioning-only	200	4.781	13.457	314.83

Reference	Role-LoRA ΔAL [95% CI]	Speedup [95% CI]
Frozen rerun	$+0.254$ $[0.208, 0.300]$	1.035 $[1.028, 1.042]$
Full fine-tuning	$+0.063$ $[0.023, 0.102]$	1.009 $[1.004, 1.015]$
Conditioning-only	$+0.037$ $[-0.006, 0.080]$	1.006 $[1.000, 1.012]$

The test cases are the first 128 measured seed–class pairs of the locked horizon-evaluation bank. They have zero overlap with the selection bank or the earlier Final128 bank used in Table 18. Evaluation keeps the strict sample-level verifier, shared causal innovations, and numerical settings fixed; Role-LoRA is merged for inference. Warmup rows are excluded. Timing uses unprofiled sampler calls, and 3,420 GPU-monitor observations record no external process or monitoring error. Paired bootstrap intervals use 10,000 resamples with seed 20260906. Training occurred under shared-GPU conditions, so this study reports sampling latency rather than comparative training GPU-hours.

Role-LoRA improves AL by 5.55% over the frozen rerun and yields the highest observed AL among the four conditions. Its AL difference from full fine-tuning is positive throughout the paired 95% interval. The conditioning-only contrast includes zero for AL and one for speedup (unrounded speedup interval $[0.99981, 1.01166]$), so this comparison does not resolve an ordering between those two adaptation choices.

Training-seed extension. We repeat Role-LoRA and conditioning-only training with seeds 20260925 and 20260926, retaining the recipe, learning rate, and eight-checkpoint selection budget above. Each run selects the checkpoint with highest AL on the same 64 selection cases, with earlier updates breaking ties. Table 17 combines these runs with seed 20260818 from Table 16. All six checkpoints are evaluated on the same 128 test cases: the extension adds training replicates, not new test cases.

Table 17: Role-LoRA versus conditioning-only across three training seeds. Selected EMA steps are listed as LoRA/conditioning-only. ΔAL is LoRA minus conditioning-only. All per-seed 95% intervals are recomputed with 10,000 paired case resamples (bootstrap seed 20260925), conditional on the selected checkpoints. The last row averages the three seed means.

Training seed	EMA steps	LoRA AL	Cond. AL	ΔAL [95% CI]
20260818	7000/200	4.8175	4.7806	$+0.0369\ [-0.0067, 0.0807]$
20260925	200/4000	4.7982	4.7698	$+0.0284\ [-0.0083, 0.0655]$
20260926	8000/250	4.7916	4.7626	$+0.0290\ [-0.0097, 0.0667]$
Mean	–	4.8025	4.7710	$+0.0315$

Role-LoRA's mean AL is 0.66% higher, with 313.76 versus 315.20 logical target forwards. All three AL differences are positive, but each per-seed interval includes zero. Role-LoRA achieves these AL values with 267,264 trainable parameters, about $1/54.6$ of conditioning-only's 14,605,056. The extension reports AL and calls only: evaluation ran on separate GPUs with interference in one run, and shared-GPU training timings are excluded from cost comparisons.

E.6 Rank, placement, and checkpoint studies

The seven-method comparison retains 896 measured rows: the same 128 unique seed–class pairs for every method, in two disjoint 64-case shards. Each shard uses one GPU, batch one, horizon eight, CFG 1.5, EMA weights, and the unchanged sample-level reflection verifier. Packed re-encoding uses shared causal innovations with recursive drafting disabled and the unused terminal coarse prediction omitted. The runtime gates pass for all seven methods in both shards. The reported AL is the per-case mean accepted count over speculative rounds, averaged over cases; Calls is the per-image logical target-invocation count.

The five new variants select their checkpoint by highest AL on a separate 64-case bank, breaking ties by fewer logical calls. The candidate steps are 250, 1000, 3000, 7000, and 8000; these selection cases do not overlap the Final128 bank. The selected steps are 8000 for rank four, 3000 for rank sixteen, 1000 for uniform and reversed weighting, and 8000 for Packed-FREE. The Follow-up Rank8 reference instead reuses a previously selected step-7000 checkpoint from a separate adaptation run; it is not the primary comparison's historical Role7000 checkpoint. Its earlier candidate grid and tie rule differ from the new variants'.

The adapted configurations share the step-180120 coarse base and a nominal 8,000-update limit. Accumulation, initialization, and trainable architecture vary: the rank-eight checkpoint records accumulation 16, the new variants use accumulation one, and both specify global batch 16. The new logs record microbatch 16 and 8,000 completed updates; the earlier reference's runtime record leaves execution equivalence unresolved. Packed-FREE initializes a separate refiner from a pretrained FREE checkpoint and trains 29,235,472 parameters. These differences in training and pretraining make this a comparison of selected checkpoints under common evaluation, rather than an isolated test of rank, loss weights, or architecture at equal total compute.

Means and AL/call contrasts are computed from the paired rows, with pointwise percentile 95% intervals from 20,000 paired bootstrap resamples and seed 20260914. The intervals quantify case variation for the selected checkpoints. External processes occupied both shard GPUs during evaluation, so we exclude wall times and timing ratios. Pairing and execution checks passed, allowing us to retain AL and logical target calls.

The capacity and objective panels in Table 19(a,b) report algorithmic results from the seven-method Final128 checkpoint comparison (Table 18); its invalid timers are excluded. Table 15 reports the independent 64-case adapter holdout.

E.6.1 Capacity, objective, and checkpoint panels

The step sweep in Table 19(c) evaluates the primary training trajectory at steps 250, 1000, 3000, 7000, and 8000, including Primary Role7000. It uses 64 selection cases per checkpoint, with two excluded warmups and timing excluded. Step 3000 has the highest observed AL on this bank. The

Table 18: Selected checkpoints on 128 paired cases ($B = 1, L = 8$, CFG 1.5). Step is the adaptation checkpoint; ΔAL compares against Follow-up Rank8, not Primary Role7000, with pointwise paired 95% intervals. Training/selection differ, and interference invalidates timing (Appendix E.6).

Variant	Step	AL↑	Calls↓	ΔAL [95% CI]
Frozen rerun	–	4.531	325.40	$-0.2544\ [-0.3031, -0.2057]$
Role-LoRA, rank 4	8000	**4.790**	314.47	$+0.0047\ [-0.0328, +0.0423]$
Role-LoRA, rank 8	7000	4.785	**314.45**	–
Role-LoRA, rank 16	3000	4.768	315.35	$-0.0175\ [-0.0555, +0.0209]$
Uniform, rank 8	1000	4.769	315.28	$-0.0161\ [-0.0515, +0.0199]$
Reversed, rank 8	1000	4.733	316.84	$-0.0520\ [-0.0872, -0.0172]$
Packed-FREE	8000	4.733	316.44	$-0.0526\ [-0.0963, -0.0092]$

sweep describes checkpoint variation on selection data; the primary test comparison retains its preselected checkpoint. The source record also has an inconsistent variant label, which prevents interpreting this sweep as a loss-weighting comparison.

Table 19: Adapter capacity, horizon weighting, and checkpoint selection.

(a) Adapter capacity

Rank	Trainable parameters	AL↑	Calls↓
0	0	4.531	325.40
4	133,632	4.790	314.47
8	267,264	4.785	314.45
16	534,528	4.768	315.35

(b) Horizon weighting

Horizon weighting	AL↑	Calls↓
Early-heavy	4.785	314.45
Uniform	4.769	315.28
Reversed	4.733	316.84

(c) Checkpoint selection

Training step	AL↑	Target calls↓
250	4.7532	315.61
1000	4.7245	317.09
3000	**4.7765**	**314.66**
7000	4.7704	315.05
8000	4.7761	314.89

Table 19 notes. (a,b) Selected-checkpoint comparisons on Final128 (Table 18), not the earlier 64-case holdout; rank zero is the frozen packed rerun and early-heavy is the weighting reference. AL and calls use the same bank, but training and selection vary, so these comparisons do not isolate rank or loss-weight effects. (c) Rank-eight sweep on 64 paired selection cases, separate from the 128-case placement bank. All five checkpoints share the primary training trajectory; step 7000 is Primary Role7000, not Follow-up Rank8, and remains the main benchmark's preselected checkpoint. Step 3000 exceeds step 8000 by 0.0004 AL; this point estimate does not establish checkpoint superiority.

E.6.2 Adapter placement

The placement, channel-guidance, and innovation studies use the locked Final128 bank, with 128 unique seed–class pairs per condition. They are separate executions on the same bank, not independent banks of evidence. The placement summary contains 640 measured rows for five conditions, with unchanged strict verification and a valid diagnostic-timing flag. Its paired contrasts compare reduced placements against the all-six condition within that campaign; the corresponding all-six AL is 4.7604, distinct from the 4.7851 reference in the earlier seven-method comparison. Excluding conditioning disables adapter updates to the context projection and AdaLN while retaining those backbone modules. The reduced-placement selection records use a separate 64-case bank, with no seed–class overlap with Final128. Loaded checkpoint identities are absent from the aggregate placement record; the table therefore reports its named conditions without assigning unverified per-row checkpoint identities.

Table 20: Role-LoRA placement on 128 paired cases with the unchanged strict verifier. Parameter counts follow Table 5; adapter removal preserves the backbone maps. ΔAL uses this campaign's all-six rank-eight reference. Conditioning comprises context projection and AdaLN modulation.

Placement	Parameters	AL↑	Target calls↓	ΔAL
Frozen re-encoder	0	4.5308	325.40	-0.2296
All six maps, rank eight	267,264	4.7604	315.56	–
Without attention adapters	211,968	4.7839	314.55	$+0.0235$
Without MLP adapters	175,104	4.7788	314.69	$+0.0184$
Without conditioning adapters	147,456	**4.8067**	**313.45**	$+0.0463$

E.7 Shared causal innovations

The CFG and noise studies use the same fixed Follow-up Rank8 checkpoint file, rather than Primary Role7000. Their runner uses $B = 1, L = 8$, evaluation mode, merged adapters, no recursive drafting, and omission of the unused terminal coarse output. Two warmups per condition are excluded and condition order rotates by case. Both summaries pass pairing and strict-verifier checks, mark timing valid, and record no foreign GPU processes at the runner's checks. Their elapsed times remain diagnostic observations, separate from the fully instrumented primary timing campaign and from one another. AL averages per-case accepted lengths; acceptance pools all measured accepted/proposed counts, including terminal truncation. In the noise study those counts are $95{,}929/159{,}261$ for shared causal noise and $58{,}340/276{,}922$ for independent auxiliary noise.

The independent-noise control samples separate Gaussian innovations for coarse-source construction and proposal rebuild. The verifier uses the actual rebuilt candidate and its generating mean; it does not substitute a new residual after sampling. The two constructions consume random numbers differently, so pairing by initial seed and class does not imply identical downstream uniforms or target paths.

Table 21: Shared-noise ablation on 128 locked paired cases, with $L = 8$, CFG 1.5, the fixed Follow-up Rank8 checkpoint, and strict reflection verification. AL is averaged over cases; acceptance pools the recorded accepted/proposed counts. Calls denotes logical target invocations per image; times are diagnostic seconds per image.

Coarse/rebuild innovations	AL↑	Acceptance↑	Calls↓	Diagnostic time (s)↓
Shared causal	**4.7851**	**60.23%**	**314.45**	**13.010**
Independent auxiliary	1.6782	21.07%	544.31	22.577

E.8 Efficiency and robustness

The four-rank supplementary studies in Tables 23, 3, and 27 use their own evaluation banks. The horizon study uses 64 selection cases and 192 disjoint test cases drawn from the historical locked 256-case horizon bank. CFG and cross-system studies reuse subsets of the horizon cases. Sample sizes are reported per study because these evaluation banks overlap.

The three-method CFG and cross-system runs use batch size one, four allocated ranks, two warmups per method, rotating method order, synchronized wall-time boundaries, and no internal profiler. Only rank zero samples the target-only chain; speculative target verification uses all four ranks. The target, FREE, and Role-LoRA checkpoints match the primary comparison; these records do not separately identify a standalone base-drafter checkpoint. Both cross-system records report PyTorch 2.11.0+cu128 and CUDA 12.8; they use the same recorded models and implementation. System B is the server used in the earlier allocation study, rerun here alongside system C. The admitted supplementary JSONs do not include GPU-model, interconnect, clock, or power readbacks, so server identity alone does not verify the hardware configuration of these new runs.

For the CFG and cross-system tables, the stored paired intervals use 10,000 bootstrap resamples with seed 20260906. The FREE-to-RELOAD final192 interval and paired AL/call changes are recomputed from the raw measured rows using 50,000 resamples with seed 20260908; the final192

target-ratio intervals are retained from its source JSON. Every resample preserves the seed–class pairing. The horizon and CFG runners do not reproduce identical trajectories on all shared cases, whereas the two matched-system runs agree on per-case AL and calls. Results are therefore compared only within each campaign, never pooled across runners.

Each timing result admitted from these supplementary campaigns has a retained normal-exit record with valid timing, no detected external GPU process, and no monitor error. The earlier interference-flagged adaptation timers remain excluded. The later Final128 comparison is admitted only for algorithmic metrics, as specified in Appendix E.6.

E.8.1 HORIZON SELECTION

Each method selects its horizon by minimum sampling latency. Increasing L from 8 to 16 raises strict AL and reduces logical target calls for both methods, but increases latency in this four-rank setting (Table 22). Both methods independently select $L = 8$ for the final comparison. These runs use direct coarse prediction and the unchanged sample-level reflection verifier.

Table 22: Horizon selection on a separate 64-case bank with four ranks. Lower mean wall time selects $L = 8$ independently for strict FREE and RELOAD. These are selection statistics, not the final test in Table 23.

L	FREE (s)	RELOAD (s)	FREE AL	RELOAD AL	FREE calls	RELOAD calls
4	7.424	7.282	2.9790	3.0342	459.09	456.78
8	7.168	6.243	4.5381	4.7672	324.28	315.19
16	9.090	7.046	5.5110	6.0362	270.97	253.11

Table 23: Horizon-selected comparison on 192 test cases disjoint from the 64 selection cases. Both methods select $L = 8$ and use four-way verification; Target activates one rank.

Method	L	Time (s)↓	vs. Target↑	AL↑	Calls↓
TARGET	–	12.178	1.000×	–	1000.00
Strict FREE	8	7.075	1.721×	4.5385	324.28
RELOAD	8	6.192	1.967×	4.7597	315.61

On the 192 disjoint test cases, RELOAD improves over strict FREE by 1.143× in speed (95% CI $[1.135, 1.150]$), 0.2212 in AL ($[0.1725, 0.2692]$), and 8.66 fewer logical calls ($[6.43, 10.87]$). Its speedup over one-active-rank TARGET is 1.967× ($[1.956, 1.978]$).

E.8.2 LOCAL PACKING COST

The local packing benchmark holds synthetic inputs fixed within each condition. It covers $B \in \{1, 2\}$ and $L \in \{4, 8, 16\}$ at timestep 700, with 30 alternating paired CUDA-event repeats per condition. Both serial and packed paths build the context cache once; observed backbone calls are L batch-B invocations versus one batch-LB invocation. Separate short runs measure incremental allocated-memory peaks. Across all six fixed-input conditions, packing trades lower re-encoder latency for increased activation memory (Table 24).

E.8.3 GUIDANCE CHANNELS

The channel study uses the same fixed Follow-up Rank8 checkpoint and Final128 evaluation protocol as the noise study in Appendix E.7.

The CFG study changes the target wrapper consistently for anchor transitions and verification. At CFG 1, all 128 paired final samples, accepted/proposed counts, round counts, and target-call counts agree; intermediate-state bitwise identity was not separately recorded.

Table 24: Synthetic-input re-encoder microbenchmark on one GPU at timestep 700. Times average 30 paired CUDA-event measurements. Serial uses L batch-B invocations; packed uses one batch-LB invocation. Additional peak allocation is measured separately in MiB (2^{20} bytes); these are not end-to-end sampler speedups.

B	L	Serial (ms)	Packed (ms)	Speedup	Serial MiB	Packed MiB
1	4	5.036	1.767	2.850×	19.5	63.2
1	8	9.778	2.210	4.424×	19.5	120.6
1	16	19.043	3.510	5.426×	19.6	237.9
2	4	5.114	2.222	2.302×	32.0	139.8
2	8	9.910	3.513	2.821×	32.1	275.1
2	16	19.565	6.070	3.223×	32.4	545.8

Table 25: Fixed-workload verification allocation at $L = 8$, with 32 paired cases per allocation. Panel (a) reports mean seconds per image and ratios of mean times. Panel (b) reports RELOAD accepted draft transitions per round (AL), logical target calls per image, and allocated GPU-seconds per image. Panel (c) gives paired-bootstrap 95% intervals for the target-relative speedups. TARGET has one active rank; both speculative methods use all G ranks.

(a) Sampling latency and speedup

G	Target (s)	FREE (s)	ReLoaD (s)	Target / FREE	Target / ReLoaD	FREE / ReLoaD
1	16.258	14.547	13.575	1.118×	1.198×	1.072×
2	15.561	10.046	9.197	1.549×	1.692×	1.092×
4	16.147	8.248	7.415	1.958×	2.178×	1.112×
8	15.364	7.794	6.985	1.971×	2.200×	1.116×

(b) Acceptance and resource use

G	Active ranks (Target / spec.)	ReLoaD AL	ReLoaD calls	ReLoaD GPU-s
1	1/1	4.832	312.47	13.58
2	1/2	4.829	312.56	18.39
4	1/4	4.873	310.72	29.66
8	1/8	4.814	313.16	55.88

(c) Target-relative speedup: 95% confidence intervals

G	Target / FREE	Target / ReLoaD
1	$[1.101, 1.135]$	$[1.180, 1.216]$
2	$[1.528, 1.571]$	$[1.665, 1.719]$
4	$[1.929, 1.988]$	$[2.138, 2.218]$
8	$[1.935, 2.007]$	$[2.162, 2.237]$

Table 26: CFG channels: 128 paired cases, fixed Follow-up Rank8 checkpoint, $L = 8$, shared causal innovations. ΔAL is three-channel minus four-channel AL; each convention defines its own consistently verified target kernel.

	Three channels		Four channels		
CFG	AL↑	Calls↓	AL↑	Calls↓	ΔAL
1.0	5.1068	301.25	5.1068	301.25	0.0000
1.5	**4.7851**	**314.45**	4.7050	317.89	+0.0802
3.0	**3.2771**	**398.96**	3.0794	413.84	+0.1977

Table 27: Matched systems: the same 32 cases, four ranks, $L = 8$, CFG 1.5. Speedups and paired 95% intervals are computed within each server. System B also hosts the allocation study.

System	Target (s)	FREE (s)	ReLoaD (s)	Speedup vs. FREE↑	95% CI
C	12.158	7.062	6.164	1.146×	$[1.124, 1.168]$
B	15.309	8.443	7.170	1.178×	$[1.155, 1.200]$

E.8.4 Cross-system and isolated-memory measurements

The matched two-server comparison uses the same checkpoints and 32 seed–class pairs. Per-case AL and logical calls agree between servers despite different wall times; each server's paired comparison favors ReLoaD over strict FREE.

Independent loading separates method memory from the co-resident comparison harness. Table 28 resets allocator peaks before warm-up, so peaks cover both warm-up and measured sampling; wall-time means exclude the two warm-ups. Resident allocation is recorded after model loading. Peak reservation describes the PyTorch allocator, not whole-device memory. The target-only reference samples on rank zero, while all four ranks retain target weights and the speculative methods distribute verification. At $L = 8$, each of the three verification-worker ranks has a 2.601 GiB allocated peak for either speculative method. Rankwise peaks are not added to estimate a simultaneous system peak.

Table 28: Independent method loading on system C with four ranks, 30 measured cases per entry, two warm-ups, and no profiler. Memory columns report coordinator-rank GiB (2^{30} bytes); its allocated peak is the largest rankwise peak. Peaks include warm-up; Δpeak is peak allocated minus resident allocation, not a between-method difference. Times exclude loading, VAE decoding, and file output.

Method	L	Time (s)	Resident (GiB)	Peak alloc. (GiB)	Peak reserv. (GiB)	Δpeak (GiB)
Target	–	12.075	2.538	2.574	2.766	0.036
Strict FREE	4	7.385	2.649	2.685	2.889	0.036
ReLoaD	4	7.121	2.812	2.887	3.064	0.075
Strict FREE	8	7.131	2.649	2.711	2.906	0.062
ReLoaD	8	6.255	2.812	2.939	3.199	0.127
Strict FREE	16	8.837	2.649	2.759	3.008	0.110
ReLoaD	16	6.980	2.812	3.046	3.365	0.234

The isolated 30-case runs measure each method in a separate process without method rotation. Their timings characterize these memory-profiling runs; paired speedups come from the matched final comparison. Sampler timers exclude loading, initial-noise construction, VAE decoding, and file output. Kernel and communication durations overlap, preventing an additive phase breakdown.

E.8.5 Training-step and online-horizon pilots

The checkpoint pilot has 16 cases, with seed $20260914000 + i$ and class $(61i + 7) \bmod 1000$ for $i = 0, \ldots, 15$. Two disjoint eight-case shards contain all five Role-LoRA steps; their union has exactly the same cases at every step. The online pilot uses seed $20260914100 + i$ and class $(67i + 19) \bmod 1000$, with every case evaluated under fixed and adaptive policies for both methods. These two pilot banks are disjoint. Their runners request one GPU, batch one, CFG 1.5, EMA weights, and sample-level reflection; the checkpoint runner fixes $L = 8$. Fixed/adaptive execution order alternates by case, and neither runner permits checkpoint or policy selection from these results.

Tables 29 and 31 report percentile 95% intervals from 50,000 paired case resamples with seed 20260914. They quantify AL and call-count variation across cases with the checkpoints and untuned policy held fixed. Retraining and selection uncertainty are outside this resampling design; elapsed times are diagnostic measurements without inferential intervals.

The CFG audit records completion, target/Role checkpoints, and implementation metadata. Pilot and robustness summaries record case identities or chain modes and seeds, with requested execution settings in the archived runners. Full runtime metadata and live per-position tensors are unavailable for these summaries. Their measurements are therefore analyzed separately from the fully instrumented timing campaigns. The checkpoint summaries contain AL and logical calls; analyzing their relationship to prediction error, overlap, or training loss would require additional measurements.

The training-step scan varies only the Role-LoRA checkpoint from one training run while keeping its base drafter and evaluation cases fixed. The requested branch is EMA at each step; these are Role-LoRA checkpoints, not the separately trained raw step-250 packed-refiner pilot in Table 14. The five means span AL 4.6713–4.7746, with nonmonotonic changes in both AL and logical calls.

Table 29: Role-LoRA checkpoint pilot: the same 16 paired cases at each step, fixed $L = 8$. Means and case-bootstrap 95% intervals are descriptive, not a checkpoint-selection criterion. Calls are logical target invocations per image.

Role step	AL↑	95% CI	Calls↓	95% CI
250	4.6713	$[4.5724, 4.7851]$	319.44	$[314.75, 323.69]$
1,000	4.7746	$[4.6474, 4.8974]$	315.00	$[309.69, 320.69]$
3,000	4.6928	$[4.5471, 4.8431]$	318.62	$[312.00, 325.12]$
7,000	4.7511	$[4.6120, 4.8856]$	316.06	$[310.25, 322.38]$
8,000	4.7624	$[4.6688, 4.8538]$	315.38	$[311.75, 319.19]$

The step-1000 minus step-7000 AL contrast is +0.0234 (95% CI $[-0.0999, 0.1438]$), leaving their ordering unresolved. The scan measures checkpoint variation within one training run; elapsed training cost and variation across retraining seeds were not measured. Primary timing retains step 7000, and Appendix E.6 specifies the separate Final128 checkpoints.

The online policy in Table 30 starts at $L = 8$ and moves one rung within $\{4, 8, 16\}$ using the preceding completed round's accepted/proposed fraction: upward at fractions at least 0.75, downward at fractions at most 0.25, and otherwise unchanged. A truncated terminal round does not update the policy. Both methods use the same untuned thresholds and their existing frozen checkpoints. AL increases on 15/16 FREE pairs and 13/16 RELOAD pairs.

Table 30: Online-horizon pilot on 16 paired cases, $B = 1$, CFG 1.5, one GPU. Adaptive (4/8/16) denotes an online horizon policy that switches among $L \in \{4, 8, 16\}$ using the previous completed, untruncated round's acceptance fraction. AL is the mean accepted count per round, averaged over cases. Calls count logical target invocations, not target-scored samples; this instrumented pilot is not a timing comparison.

Method	Horizon policy	AL↑	Calls↓
Strict FREE	Fixed 8	4.5047	326.69
Strict FREE	Adaptive (4/8/16)	5.0040	296.62
RELOAD	Fixed 8	4.7585	314.94
RELOAD	Adaptive (4/8/16)	5.1107	290.88

Table 31: Paired effects of the online heuristic relative to fixed $L = 8$, on the same 16 cases as Table 30. Positive values indicate higher AL or fewer logical calls. Intervals resample seed–class pairs, not individual rounds.

Method	ΔAL	95% CI	Fewer calls	95% CI
Strict FREE	+0.4992	$[0.3271, 0.6722]$	30.06	$[22.25, 37.75]$
RELOAD	+0.3522	$[0.1575, 0.5337]$	24.06	$[16.19, 31.19]$

Logical calls decrease by 30.06 for FREE and 24.06 for RELOAD, but the summary does not provide horizon histograms, target-scored sample counts, or peak memory. The instrumented mean elapsed times are 14.57 versus 15.28 seconds for fixed versus adaptive FREE, and 12.95 versus 14.05 seconds for RELOAD. The adaptive policy trades fewer logical calls for longer instrumented

elapsed time. We assess this untuned policy through paired AL and call counts, with the paired banks and uncertainty defined above.

E.9 CAUSAL AND NUMERICAL CHECKS

Table 32: Implementation checks for the deployed $L = 8$ path and reflection verifier. Causality and packed–serial checks use the same 100 anchors at timestep 800 in production and FP32 control modes. They test local causality and numerical consistency; Appendix A gives the target-law proof.

Audit	Trials	Failures	Tolerance or statistic
Future-noise causality ($L = 8$)	2×100	0	maximum forbidden influence $= 0$
Execution-graph regression	31	0	31/31 passed
Reflection maximal coupling	5×100K	0	maximum acceptance error < 0.0023
Packed–serial, production TF32	100	–	maximum difference 2.2333×10^{-3}
Packed–serial, FP32 control	100	–	maximum difference 6.6757×10^{-6}

For the deployed $L = 8$ path, hooks observe exactly two drafter-backbone invocations: one anchor-conditioned coarse invocation with seven useful outputs and one packed rerun over eight causal sources. Requested protocol fields, live readback, and observed counts agree in every admitted row. The 200 causality trials comprise 100 production-mode tests and 100 repetitions with TF32 disabled. Packed–serial differences are reported discrepancies, not bitwise-equality passes; these local checks at timestep 800 do not bound accumulated floating-point error over a full sampling chain.

E.9.1 EXTENDED CAUSAL AND NUMERICAL AUDIT

The expanded local audit varies batch size, horizon, and timestep beyond the original $L = 8$, timestep-800 checks. All 180 future-innovation interventions preserve the protected output prefix exactly; all 168 nonempty future suffixes change. At $L = 4$, the five trials perturb zero-based positions $(0, 1, 2, 3, 0)$; at $L = 8$ and $L = 16$ they perturb $(0, 1, 2, 3, 4)$, so longer horizons are not exhaustively tested at every position.

Table 33: Extended synthetic-input audit with TF32 enabled. Each (B, L) covers six timesteps $\{32, 100, 300, 500, 700, 950\}$ and five trials per timestep. Causal passes require zero protected-prefix change and a changed nonempty suffix. Packed–serial differences are measured separately, without an error-threshold pass test.

B	L	Causal passes	Prefix max	Max abs. diff.	Max RMSE
1	4	30/30	0	3.35×10^{-3}	4.01×10^{-4}
1	8	30/30	0	3.22×10^{-3}	3.80×10^{-4}
1	16	30/30	0	3.61×10^{-3}	3.45×10^{-4}
2	4	30/30	0	1.29×10^{-5}	1.32×10^{-6}
2	8	30/30	0	1.35×10^{-3}	1.94×10^{-5}
2	16	30/30	0	3.91×10^{-3}	1.45×10^{-4}

The largest packed–serial absolute difference is 3.9082×10^{-3} and the largest RMSE is 4.0147×10^{-4}, attained in different trials. These production-TF32 measurements quantify local numerical differences, while the causal criterion tests protected-prefix invariance. The FP32 control in Table 32 provides a precision comparison. Full-chain error bounds would require separate analysis; Appendix A establishes the distributional result in exact arithmetic.

E.9.2 LIVE DENSITY, INNOVATION, AND REPLAY AUDITS

CFG and generating densities. The selected Role7000 path is checked for $B \in \{1, 2\}$ and CFG $\{1, 1.5, 3\}$ in FP32 with TF32 enabled. Live target outputs satisfy the first-three-channel CFG rule, retain the conditional fourth channel and carried feature, and the coarse/rerun call arguments contain no extra guidance or null-label branch. Every recorded proposal mean, variance, and timestep matches an actual scheduler sample. Across all six configurations, 8,816 batch-position records

have zero candidate reconstruction residual; all deliberately shifted means ($+0.01$) are detected. Observed and control runs have identical final latents, accepted-length sequences, and logical target-call counts.

Table 34: Live CFG/density audit for the selected Role7000 path. One seed per condition; batch one uses class 11, batch two uses classes 11 and 12. Positions count batch-position records. All shifted-mean controls ($+0.01$) are detected; candidate reconstruction residual is zero.

B	CFG	Positions	Controls detected	Observer unchanged
1	1.0	1,199	1,199	yes
1	1.5	1,221	1,221	yes
1	3.0	1,540	1,540	yes
2	1.0	1,436	1,436	yes
2	1.5	1,456	1,456	yes
2	3.0	1,964	1,964	yes

The counts above refer to proposal positions within six trajectories. The shifted-mean control tests detection of an incorrect generating density. The separate channel ablation in Table 26 evaluates four-channel CFG.

Innovation and reconstruction checks. Four seeds are each run in FP32 with TF32 off and on, giving eight source chains at $B = 1, L = 8$, CFG 1.5. All eight observer comparisons preserve final outputs, accepted sequences, and calls. Across 10,003 proposal positions (5,006/4,997 by mode), coarse construction and rebuild share innovations and timestep identities exactly; the candidate reconstruction residual is zero and the largest committed reflection residual is 4.7684×10^{-7}. Resampled-noise and cyclically swapped-index controls are each detected at all 9,979 eligible positions. These controls exclude timestep zero and variance at most 10^{-12}; index swaps also require more than one proposal. Reconstruction checks use absolute and relative tolerance 2×10^{-5}, not a distributional tolerance.

Fixed-live-input layout and precision replay. The first full-horizon round in each 200-timestep bucket provides five saved contexts per chain, hence 40 contexts. Each is re-evaluated serially, in chunks of two or four, and in reverse order with restoration; negating the fourth innovation leaves the first four predictions unchanged in all 160 layout/intervention comparisons. Each context also has one rerun-matmul precision toggle with spatial inputs, innovations, and cached anchor K/V fixed. The source chains are not rerolled under these alternatives.

Table 35: Fixed-live-input replays in each source numerical mode: 20 rounds from four chains, with four layout alternatives and one precision toggle per round. Entries are observed maxima, not pass tolerances. Decision differences use the same verifier uniforms at all eight positions, including unvisited suffixes.

Source mode	Replays	Max $\lvert\Delta\widehat{\epsilon}\rvert$	Max acceptance prob. change	Decision differences
FP32, TF32 off	Layout (80)	6.676×10^{-6}	4.898×10^{-4}	0
FP32, TF32 off	Precision toggle (20)	3.157×10^{-3}	1.192×10^{-2}	1
FP32, TF32 on	Layout (80)	3.239×10^{-3}	1.020×10^{-2}	0
FP32, TF32 on	Precision toggle (20)	3.074×10^{-3}	1.141×10^{-2}	0

All layout replays preserve the coupled accept decisions at the examined positions; precision toggles change one of 320 offline decisions. The replay set includes unvisited suffix positions, so its decision-change count measures local sensitivity rather than live acceptance. Dependency, identity, and reconstruction checks are evaluated separately from numerical differences. Recomputing scheduler statistics and candidates at saved rebuilt sources can change acceptance probabilities even with identical noise outputs, as observed in reverse-and-restore execution.

Known-law synthetic chain. A separate scalar Gaussian test runs the production reflection coupling for 16 transitions, with target $x_{j+1} = 0.93x_j + 0.02 + 0.2z_j$, $x_0 \sim \mathcal{N}(0, 1)$, and proposal

mean shifted by $+0.3$. The 20,000 independent trajectories use fresh innovations and uniforms at each transition. Packed, chunk-257, and reverse-and-restore execution produce identical final samples under the same random inputs. The empirical final mean/variance are 0.20478/0.37016, versus analytic values 0.19625/0.36510; the empirical-CDF maximum discrepancy is 0.01014, below the runner's DKW threshold 0.01573 at $\alpha = 10^{-4}$. The three layouts reuse one random bank. This synthetic-chain diagnostic checks the production coupling against an analytically known distribution across three execution layouts.

E.10 TARGET-RELATIVE SAMPLE QUALITY

Class-balanced 50K evaluation. The locked bank contains 50,000 distinct IDs and exactly 50 examples for each of 1,000 ImageNet classes. The completed metric record reports eight 6,250-image shards for each of RELOAD and TARGET. The banks share labels and initial Gaussian noise; subsequent sampler randomness is method-dependent. FID and KID use `torch-fidelity` 0.3.0 with PyTorch 2.11.0+cu128. We report KID with the stored standard deviation across subset estimates. Subset count, subset size, and subset RNG seed are unavailable in the summary, so this spread is reported as SD rather than an inferential interval. Small negative unbiased KID estimates are retained.

Precision and recall use Inception-2048 features and third-nearest-other-neighbor radii, excluding each reference point's self-distance. Precision tests RELOAD samples against the generated target manifold ($36{,}147/50{,}000$ hits); recall tests target samples against the RELOAD manifold ($36{,}445/50{,}000$ hits). A separate even/odd canonical-ID split of the target bank gives two disjoint 25K sets. The 25K split and 50K comparison differ in both sample size and pairing, so we interpret each FID at its own evaluation size. All metrics use generated target images as the reference; real-ImageNet quality and distributional equivalence were not tested.

E.10.1 INDEPENDENT-BANK TARGET-RELATIVE DIAGNOSTIC

The earlier 10K diagnostic complements the paired 50K comparison in Section 4.6 with an independent generated target bank. Target-B is disjoint from Target-A and approximately class-matched; the two independent comparisons share Target-B.

For the 10K quality diagnostic, Target-B is greedily matched to Target-A's class counts, with unmatched deficits filled from the remaining pool. The count histograms have ℓ_1 distance 982 (4.91% total variation after normalization); both independent comparisons share Target-B. All banks cover 1,000 classes, and both comparisons use the same fixed sample selection and Inception detector. KID uses 100 subsets of size 1,000 with seed 20260905; its 95% subset Monte Carlo interval is the subset mean $\pm 1.96 s/\sqrt{100}$, where s is the subset-estimate standard deviation. This interval conditions on the fixed banks and excludes sampling variation from regenerating images. The resulting FID/KID are generated-to-generated agreement measures.

Table 36: Target-relative diagnostic with 10,000 generated samples per set. Target-A shares IDs with RELOAD-A; Target-B is disjoint and approximately class-matched. Each non-class-balanced bank covers all 1,000 classes. Both KID columns are scaled by 10^3; the 95% subset Monte Carlo (MC) interval is conditional on the fixed banks. These are generated-to-generated metrics; bold marks the primary comparison.

Generated sets	Inter-set FID↓	KID $\times 10^3$↓	KID 95% MC interval ($\times 10^3$)
RELOAD-A vs. Target-A	5.360	-0.0324	$[-0.0415, -0.0233]$
RELOAD-A vs. Target-B	**5.860**	**-0.0316**	$\mathbf{[-0.0412, -0.0220]}$
Target-A vs. Target-B	5.831	-0.0278	$[-0.0379, -0.0176]$

Inter-set FID is 5.860 for RELOAD–Target-B, versus 5.831 for Target-A–Target-B, a descriptive gap of 0.029 (0.50%). Their KID estimates are -0.0316 and -0.0278 in units of 10^{-3}. The unbiased estimates can be negative; the subset intervals condition on the fixed banks. The matched-ID FID (5.360) is coupling-sensitive, so the independent-bank comparison is primary within this diagnostic. Different sample sizes and bank construction preclude treating its absolute FID values as a direct comparator for the later 50K result.