EXPLORE, REVISIT, OR COMMIT: PREDICTION DYNAMICS IN CONTINUOUS DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Continuous diffusion language models (CDLMs) refine continuous states without committing to tokens, so although every sampling step costs a full model evaluation, what each step contributes remains unclear. We address this question by studying prediction dynamics, how CDLM updates form and revise token predictions, through a framework that assigns each position at each step to commitment, revisitation, or exploration. Applying it to four representative CDLMs (LangFlow, Plaid, ELF, and Cola-DLM), we find that revisitation, in which the predicted distribution moves away from an earlier state and then back toward it, is widespread: it persists after the top-1 token stabilizes, and additional sampling steps advance commitment without removing it. We hypothesize that revisitation stems from candidate multiplicity and cross-position dependence, and our experiments support both sources: revisitation often involves competition among a few candidates, with the same rival returning at the same position, and in LangFlow and ELF, freezing the prediction memory of other positions changes how a revisiting position is revised. Building on these findings, we propose velocity blending, a training-free sampling control that blends consecutive velocities in two ways: late interval merging saves about a quarter of sampling time while changing far fewer native tokens than early merging, whereas velocity damping has setting-dependent effects on revisitation and quality. Taken together, our results separate observing, altering, and exploiting prediction dynamics.

1 INTRODUCTION

Language models represent text as sequences of tokens (Vaswani et al., 2017), but they differ in how they organize token decisions across successive generation steps. Autoregressive (AR) models decode left to right, conditioning on preceding tokens (Brown et al., 2020; Touvron et al., 2023). Similarly, discrete diffusion language models (DDLMs) decode several tokens at each step (Austin et al., 2021; Sahoo et al., 2024; Nie et al., 2025), deciding which positions to fill in parallel and in what order (Kim et al., 2026; Ringel et al., 2026; Wang et al., 2025; Huang et al., 2026). Under progressive decoding, sampling steps have a clear role: more steps fix fewer tokens at once, letting earlier tokens inform later ones (Kim et al., 2025; Ringel et al., 2026; Li & Cai, 2025).

By contrast, the continuous diffusion language model (CDLM) samplers studied here update continuous states within a sequence or block without explicitly locking individual tokens; these states serve both as token readouts (Dieleman et al., 2022; Hu et al., 2026) and as context for other positions. CDLMs have advanced rapidly, from embedding diffusion (Li et al., 2022; Gulrajani & Hashimoto, 2023) to continuous flows (Hu et al., 2024; Chen et al., 2026; Hu et al., 2026) and latent priors with learned decoders (Guo et al., 2026), yet this progress has centered on what the model predicts, while samplers typically fix the number and placement of steps before generation. Since sampling repeatedly evaluates the model, not knowing what each update contributes leaves open where computation is best spent. Existing analyses characterize diffusion regimes (Biroli et al., 2024) or when tokens stop changing in discrete models (Lu, 2026); we instead study *prediction dynamics* in CDLMs by measuring stabilization and the withdrawal and recovery of candidate support at each position.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

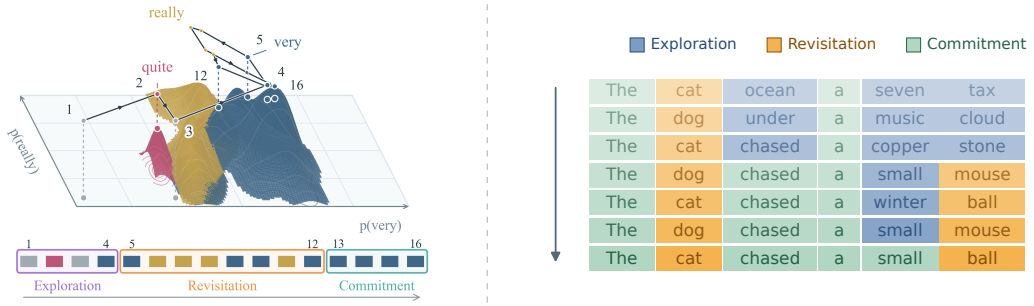


Figure 1: The two views illustrate exploration, revisitation, and commitment during generation. *Left:* a token trajectory; peak colors distinguish candidates, and surface height depicts readout density. *Right:* top-1 tokens across a sentence over seven steps.

To fill this gap, we study the prediction dynamics of CDLMs and whether they can guide more efficient sampling. We track the token distribution at each position and step and assign each observation to one of three mutually exclusive modes: **commitment**, where the top-1 token already equals the final token and does not change again (Lu, 2026); **revisitation**, where the predicted distribution moves away from an earlier state and then back toward it; and **exploration**, where the top-1 prediction has not yet stabilized (Section 3.2). In Figure 1, one position tries several candidates, alternates between *really* and *very*, and then commits to *very*, while positions across a sentence pass through these modes at different times. Unlike AR and progressive masked-diffusion samplers without remasking, which retain decoded tokens, the CDLM samplers studied here can revisit, redistributing candidate support even beneath stable top-1 predictions. Intuitively, revisitation can be costly: updates that move support back and forth spend computation without changing the output, and competition still unresolved at the end of sampling may leave tokens that do not fit together. This motivates studying how revisitation arises and how prediction context shape it.

We hypothesize that revisitation has two sources: *candidate multiplicity*, as language often admits several plausible tokens at a position (Gu et al., 2018; Huang et al., 2022), and *cross-position dependence*, as the suitability of each token depends on the tokens chosen at other positions. Progressive decoding handles this dependence by conditioning later predictions on already decoded tokens, while predictions within a step do not directly condition on one another's new outputs (Gu et al., 2018; Liu et al., 2025; Li et al., 2026). CDLMs instead update positions jointly without fixing tokens: each evaluation's output predictions enter subsequent evaluations through the updated continuous state and, in self-conditioned models, explicit prediction memory (Chen et al., 2023; Karimi Mahabadi et al., 2024). Each position is thus refined in a changing context, where context and prediction feedback can move support away from a candidate and back before dependent choices settle, producing revisitation.

We examine these sources in four representative CDLMs spanning the families above: embedding diffusion (Plaid; Gulrajani & Hashimoto, 2023), continuous flows (LangFlow and ELF; Chen et al., 2026; Hu et al., 2026), and latent priors with learned decoders (Cola-DLM; Guo et al., 2026). Our empirical study yields three findings. First, late updates increasingly act on committed positions, yet revisitation reaches many positions and distributions keep changing after top-1 stabilization (Section 4.1). Second, revisitation often involves competition among a few candidates, with the same rival returning at the same position, consistent with candidate multiplicity (Section 4.2). Third, additional sampling steps advance commitment without removing revisitation, and in LangFlow and ELF, freezing the prediction memory of other positions changes how a revisiting position is revised, consistent with cross-position dependence (Section 4.3). Together, these findings describe where sampling computation goes: late updates increasingly refine committed positions, whereas revisitation persists across the step counts we test. They thus point to two targets for adjusting sampling, omitting late updates and moderating back-and-forth updates, which we test next.

Guided by these findings, we propose *velocity blending*, a training-free sampling control that blends the current and previous velocities advancing the continuous state (Section 5). *Interval merging* fully reuses the previous velocity over two adjacent solver intervals and skips the model call between them, omitting late updates; *velocity damping* partially blends consecutive velocities at selected positions, moderating back-and-forth updates (Baqué et al., 2016). Across six settings, fixed late merging saves about a quarter of sampling time while changing far fewer native tokens than early merging, with ROUGE-L and BLEU near native on summarization and translation. Our analysis

also identifies what adaptive control requires: an online estimate of commitment, which predicts whether a merge changes the output better than the tested signals, and selectors that reach positions before their revisitation ends.

Our contributions are threefold. (1) We introduce a quantitative framework for CDLM prediction dynamics, distinguishing three modes: commitment, revisitation, and exploration (Section 3). (2) Across four representative CDLMs, we show that revisitation is widespread and persists under additional sampling steps, with evidence for both hypothesized sources: competition among a few candidates and, in LangFlow and ELF, the influence of context updates at other positions (Section 4). (3) We propose velocity blending, a training-free sampling control with two operations, interval merging and velocity damping, and evaluate them across six settings (Section 5).

2 Related Work

Diffusion Language Models. Diffusion language models cast text generation as denoising in discrete or continuous state spaces. Discrete diffusion models denoise categorical token states (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024; Nie et al., 2025; Ye et al., 2025). Their samplers make token decisions and revisions explicit: confidence- and planning-based schedules select tokens to unmask (Wu et al., 2026; Lee et al., 2025; Peng et al., 2025), remasking and deferred commitment reopen or postpone decisions (Wang et al., 2025; Huang et al., 2026; Zhai et al., 2026; Shu et al., 2026), and early-convergence detection retains them (Zhang et al., 2026a). Dependency-aware schedules coordinate parallel choices (Kim et al., 2026; Ringel et al., 2026), while joint-output methods model their dependence (Liu et al., 2025; Li et al., 2026; Ji et al., 2026). Continuous models evolve representations in embedding, simplex, or learned latent spaces (Li et al., 2022; Gong et al., 2023; Karimi Mahabadi et al., 2024; Dieleman et al., 2022; Gulrajani & Hashimoto, 2023; Chen et al., 2026; Hu et al., 2026; Guo et al., 2026). Within each generated sequence or block, the samplers studied here keep updating continuous representations rather than explicitly locking individual tokens. We characterize their position–observation pairs through retrospective commitment (Lu, 2026), revisitation, and exploration.

Prediction Dynamics and Sampling Control. Prior analyses characterize diffusion regimes (Biroli et al., 2024), when linguistic properties emerge during sampling (Lu, 2026), and overwritten intermediate answers (Wang et al., 2026); we instead track whether a position's distribution returns toward an earlier state. Revision policies decide which tokens to change: STaRR adjusts remasking using temporal and spatial confidence differences (Sun et al., 2026), and Posterior Refinement uses full-draft confidence to retain or regenerate tokens (Agarwal et al., 2026), whereas revisitation measures departure from and renewed proximity to an earlier distribution. To reduce evaluations of fixed generators, fast solvers use higher-order approximations to reduce integration error (Lu et al., 2025) and Few-shot Temporal Pruning searches time grids with validation data (Li et al., 2024); training and distillation enable few-step generation (Hu et al., 2024; Lee et al., 2026; Yoo et al., 2026); and Token Time Continuous Diffusion assigns position-specific times (Bansal & Sanghavi, 2026). This work focuses on candidate competition and its response to context and feedback; following damping in parallel mean-field inference (Baqué et al., 2016), we use these dynamics to motivate velocity damping and interval merging without retraining. And we find that merging late solver intervals, where most positions have committed, can preserve native outputs better than merging early ones. We provide a detailed survey on these two topics in Appendix A.

3 Preliminaries

3.1 Unified Overview of CDLMs

Let $Z_k \in \mathbb{R}^{L\times d}$ denote the continuous state at solver boundary k, with L positions and representation dimension d. For embedding-prediction interfaces, $\widehat{X}_k$ is the predicted-clean representation, an estimate of the uncorrupted input rather than a categorical distribution, and S_k is the incoming prediction memory used for self-conditioning. Across models, observation $n \in \mathcal{N}$ provides a distribution $p_{n,i} \in \Delta^{V-1}$ at position i over V vocabulary entries w, with token $y_{n,i} = \arg\max_w p_{n,i}(w)$ and confidence $\max_w p_{n,i}(w)$. Observation index n records diagnostic readouts, and $N = \max\mathcal{N}$ denotes the terminal observation.

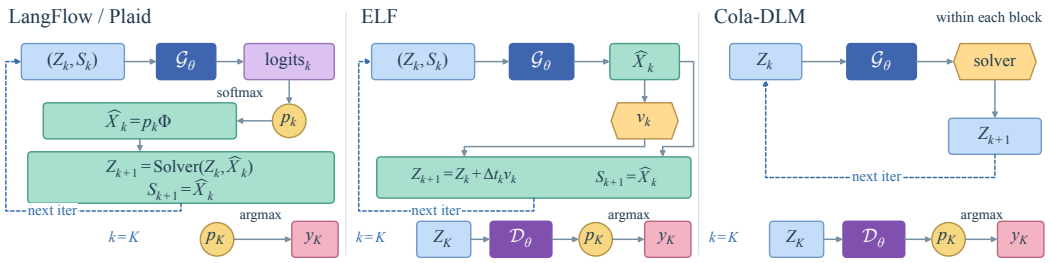


Figure 2: Simplified iterative interfaces and terminal top-1 readouts. $\mathcal{G}_\theta$ and $\mathcal{D}_\theta$ denote prediction and decoding; k indexes solver updates, and K is the final update (per block for Cola-DLM).

For ELF with conditioning q, the guided prediction mapping returns velocity v_k and predicted-clean representation $\widehat{X}_k$:

$$(v_k, \widehat{X}_k) = \mathcal{G}_\theta(Z_k, S_k, t_k; q), \qquad Z_{k+1} = Z_k + \Delta t_k v_k, \qquad S_{k+1} = \widehat{X}_k, \tag{1}$$

where t_k is diffusion time, increasing from 0 (noise) to 1 (clean), and $\Delta t_k = t_{k+1} - t_k$ its signed increment. Here $\mathcal{G}_\theta$ includes the deterministic conversion of $\widehat{X}_k$ to v_k using Z_k and t_k. Figure 2 contrasts the iterative interfaces and terminal top-1 readouts. Z is an embedding state for LangFlow, Plaid, and ELF, and a VAE latent state for Cola-DLM; $\Phi \in \mathbb{R}^{V\times d}$ denotes effective token embeddings, and $\mathcal{D}_\theta$ includes softmax. Our intermediate ELF diagnostics instead decode memory through $\mathcal{D}_\theta(S)$ after each completed solver interval.

3.2 Measuring Prediction Dynamics

Each classifiable position–observation pair receives exactly one of three modes: commitment, revisitation, or exploration, with revisitation taking priority over commitment.

Commitment. Following temporal analysis of diffusion language models (Lu, 2026), commitment records agreement with the terminal token sustained from observation

$$\tau_i^{\text{stable}} = \min\{n \in \mathcal{N} : \forall m \in \mathcal{N},\ m \geq n \Rightarrow y_{m,i} = y_{N,i}\}. \tag{2}$$

A stable top-1 token leaves other probabilities and continuous coordinates free to change.

Revisitation. We measure revisitation over consecutive observations $p_a, \ldots, p_b$ with window size $W = b - a + 1$, suppressing the position index i. Using the full-vocabulary Hellinger distance $h(p, p') = \|\sqrt{p} - \sqrt{p'}\|_2/\sqrt{2}$, the excursion of reference j is $E_{j:b} = \max_{j<n\leq b} h(p_j, p_n)$. Relative revisitation measures the recovered fraction of an excursion over eligible references; on valid windows we detect it using $W = 4$ and $\epsilon = 10^{-3}$:

$$O^\epsilon_{a:b} = \max_{a\leq j\leq b-2,\ E_{j:b}>\epsilon} \left[1 - \frac{h(p_j, p_b)}{E_{j:b}}\right]_+, \tag{3}$$

$$\alpha_{n,i} = \mathbf{1}\{O^\epsilon_{n-3:n,i} > 1/3\}. \tag{4}$$

Here $[x]_+ = \max(x, 0)$ and $\max \emptyset = 0$. The score measures relative recovery, whereas α detects whether a window contains revisitation.

Exploration. Exploration comprises observations before stable commitment and outside revisitation coverage, including temporary matches to the terminal token.

The indicators Q^{R}, Q^{C}, and Q^{E} encode the mutually exclusive modes:

$$Q^{\mathsf{R}}_{n,i} = \mathbf{1}\{\sum_{r=0}^{2} \alpha_{n+r,i} > 0\}, \quad Q^{\mathsf{C}}_{n,i} = (1 - Q^{\mathsf{R}}_{n,i})\mathbf{1}\{n \geq \tau_i^{\text{stable}}\}, \quad Q^{\mathsf{E}}_{n,i} = 1 - Q^{\mathsf{R}}_{n,i} - Q^{\mathsf{C}}_{n,i}, \tag{5}$$

with $\alpha_{N+1,i} = \alpha_{N+2,i} = 0$. A detection at n assigns observations $n-2$, $n-1$, and n to revisitation; overlapping intervals are counted once. This coverage is retrospective; an online intervention can act only after detection. Consequently, after stable commitment, a position can alternate between commitment and revisitation but cannot return to exploration. Appendices B and C detail the measurement protocols and supplementary definitions, deriving corresponding mathematical properties.

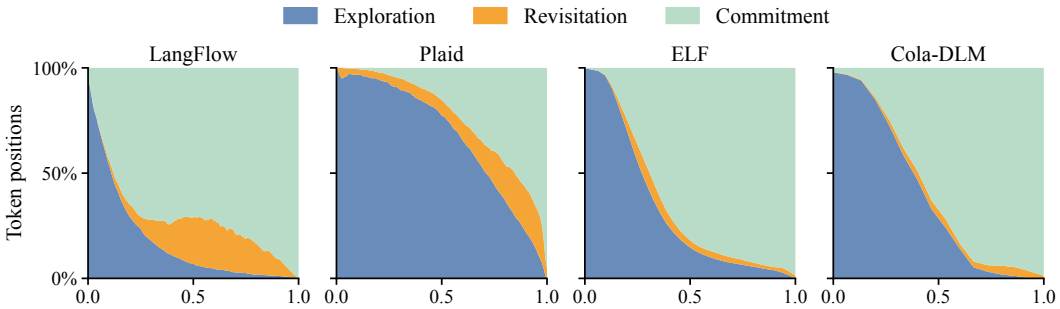


Figure 3: Prediction dynamics across four models ($\epsilon = 10^{-3}$, $n = 64$), with revisitation shown in orange. The horizontal axis shows normalized observation indices.

4 Empirical Study of Prediction Dynamics

We examine how prediction dynamics evolve during sampling, what candidate structure revisitation exhibits, and how it responds to context, feedback, and integration step size. Figure 3 compares native updates on 64 runs per model: unconditional generation for LangFlow, Plaid, and ELF, and LAMBADA continuations (Paperno et al., 2016) for Cola-DLM. The LangFlow length-128 panel produces highly repetitive outputs, so its findings concern this configuration. Statistics exclude source positions and average within then across documents (one prompt or unconditional run), with 95% document-bootstrap intervals. Appendix B.1 details the configurations; Appendix D provides supplementary analyses, controls, and interventions.

4.1 Mode Composition over Sampling

The combined share of exploration and revisitation declines over sampling at model-specific rates (Figure 3): at normalized progress .9, it covers 43.0% of Plaid positions, compared with 9.2%, 5.3%, and 5.3% in LangFlow, ELF, and Cola-DLM. In LangFlow, Plaid, and ELF, revisitation is widespread: it is detected in 1.55–6.13% of windows, reaches 24.8–91.1% of positions, and appears in every run. Across all four models, probability distributions continue to change after top-1 stabilization, accounting for 4–31% of total probability movement (Appendix D.1). Late updates resolve remaining choices and redistribute support beneath stable top-1 predictions; commitment therefore marks lexical stability, not the end of refinement.

4.2 Revisitation as Candidate Competition

Revisitation often involves identifiable competitors whose probabilities change with or without a top-1 switch. Figure 4 illustrates this structure at four Plaid positions, where the competing candidates are related words that fit the same context.

To measure this structure, we group nearby detections into episodes and follow the token that led before the episode down to its lowest probability. The rival is the token with the largest gain over that interval; episodes whose rival can be verified from cached top-10 probabilities form certified contests. In most certified contests in LangFlow, Plaid, and ELF, the leading token reaches its lowest probability at or after the commitment. It indicates that candidate competition can occur beneath a stable top-1 prediction.

Figure 5 compares two aspects of candidate competition across models: whether one rival gains more than half the probability lost by the leading token, and whether the same rival recurs at the same position. Across LangFlow, Plaid, ELF, and Cola-DLM, withdrawn support often concentrates on a single rival within a contest, consistent with candidate multiplicity. Among large contests, concentration remains similar in Plaid and ELF but weakens in LangFlow, where stronger withdrawals distribute support more broadly (Table 7). Across consecutive eligible contests, the same rival frequently recurs in LangFlow, Plaid, and ELF. Cola-DLM yields too few repeated contests to estimate recurrence. This candidate structure identifies where withdrawn support goes; we next test whether continued updates at other positions affect how a revisiting position revises its prediction.

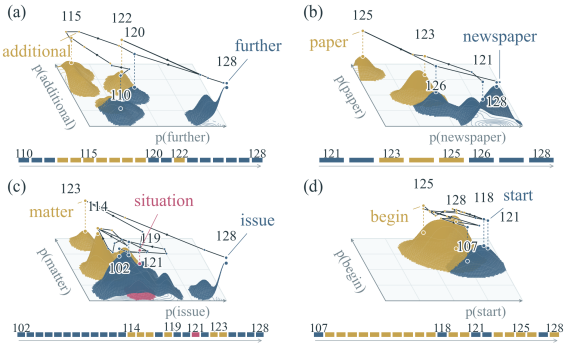


Figure 4: Revisitation in Plaid: surfaces show relative readout density over candidate probabilities, paths show readout order, and colors mark top-1 tokens.

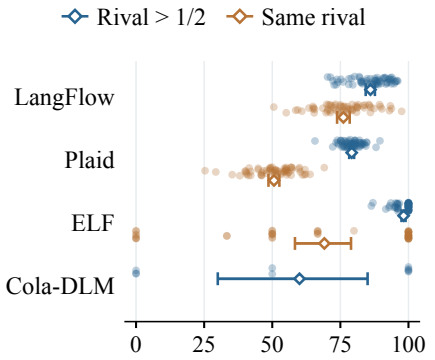


Figure 5: Concentration and recurrence: document proportions (dots), means and 95% bootstrap intervals.

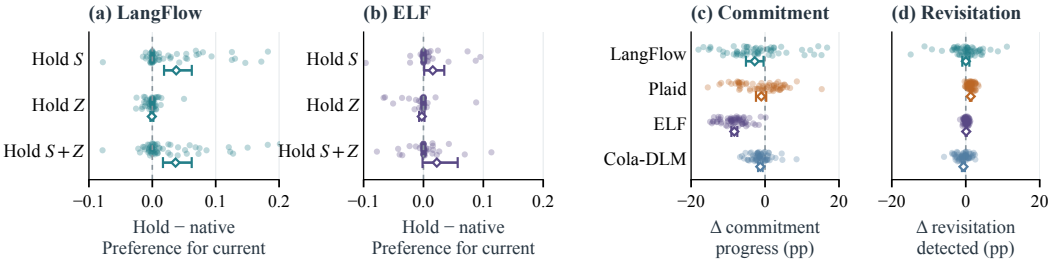


Figure 6: Context freezing (a–b) and matched-cadence budget comparisons (c–d): paired differences (dots), means (diamonds), and 95% bootstrap intervals; dots are clipped to the displayed ranges, but all runs enter the summaries.

4.3 Cross-Position Feedback and Sampling Budget

We test whether ongoing context updates redirect revisiting predictions, then ask what additional sampling steps accomplish (Figure 6). Each comparison uses 64 runs per model.

We isolate the effect of surrounding updates by keeping a revisiting position's own inputs unchanged while withholding one update of the other positions' prediction memory S, continuous state Z, or both. At the first detected revisitation n in LangFlow and ELF, we select one position, save the surrounding states entering call $n+1$, and run that call identically in all branches. Before probing call $n+2$, the Hold branches restore the saved surrounding S, Z, or both, while the native branch keeps their updated values.

We measure whether this intervention shifts its prediction toward the current distribution p_n or the previous distribution p_{n-1}, using $h(p_{\text{probe}}, p_{n-1}) - h(p_{\text{probe}}, p_n)$. Figure 6a–b plots Hold-minus-native differences: positive values indicate greater preference for p_n over p_{n-1} than under native continuation. Holding S produces this shift in both models, whereas holding Z alone has little effect; holding both has positive mean effects, with the ELF interval crossing zero. Thus, updates to surrounding prediction memory influence the selected position's revision direction, although subsequent revisitation scores do not clearly decrease. State-replacement controls provide complementary evidence for feedback through other positions (Appendix D.4).

Having established that surrounding updates affect revision direction, we next ask whether more sampling steps bring earlier commitment and less revisitation. We increase LangFlow/Plaid from 128 to 512 calls, ELF from 32 to 128 intervals, and Cola-DLM from 16 to 32; Figure 6c–d reports larger-minus-smaller budget differences, so negative commitment values indicate earlier stabilization. At matched observation cadence, LangFlow and ELF reach commitment earlier with little change in revisitation detection rate, whereas Plaid exhibits more revisitation and an uncertain change in commitment timing (Figure 6c–d). On LAMBADA and Social IQa (SIQA; Sap et al., 2019), Cola-DLM likewise advances commitment and modestly reduces detection, with revisitation

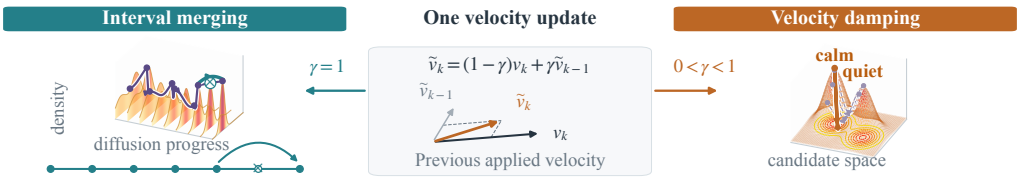


Figure 7: Both sampling controls blend consecutive velocities: interval merging keeps the previous velocity and skips a call, while velocity damping mixes the previous and new velocities at selected positions and calls; landscapes and paths are schematic.

remaining present at the larger budget. Additional updates can therefore stabilize token choices while candidate support continues to change, motivating the sampling controls in Section 5.

5 Controlling Sampling with Prediction Dynamics

Section 4 motivates two questions for sampling control: whether late updates can be omitted while preserving settled predictions, and whether moderating changes in the applied velocity can alter revisitation. Neither property guarantees these effects: a committed position can still inform unsettled ones, and revisitation is measured on predictions rather than on the velocity that damping modifies. We therefore test both through a common velocity update and ask which signal each needs to decide when and where to act (Figure 7).

5.1 Method

Velocity blending. In deterministic Euler sampling, each call advances the continuous state with the velocity it computes, $Z_{k+1} = Z_k + \Delta t_k v_k$ (Equation 1). Both operations replace this velocity by a blend of the new velocity and the velocity applied at the previous call:

$$\widetilde{v}_{k,i} = (1 - \gamma_{k,i})\, v_{k,i} + \gamma_{k,i}\, \widetilde{v}_{k-1,i}, \qquad Z_{k+1} = Z_k + \Delta t_k \widetilde{v}_k, \qquad 0 \le \gamma_{k,i} \le 1, \tag{6}$$

with $\widetilde{v}_0 = v_0$ and $\gamma_{k,i} = 0$ at source positions. *Interval merging* sets $\gamma_{k+1,i} = 1$ at every generated position: the state keeps moving with $\widetilde{v}_k$, v_{k+1} receives zero weight, and call $k+1$ is skipped, so two Euler steps become one,

$$Z^{\text{merge}}_{k+2} = Z_k + (\Delta t_k + \Delta t_{k+1})\widetilde{v}_k, \qquad S^{\text{merge}}_{k+2} = \widehat{X}_k, \tag{7}$$

where $\widetilde{v}_k = v_k$ without preceding damping and the next call returns to $\gamma = 0$; Plaid, which has no explicit velocity, reuses $\widehat{X}_k$ instead (Appendix B.5). Velocity blending thus unifies the continuous-state update; merging additionally skips the model evaluation and its prediction-memory refresh. If late calls change few outputs, merging trades them for time; because all generated positions share the diffusion time, it acts on intervals rather than tokens. *Velocity damping* uses $0 < \gamma_{k,i} < 1$ at selected positions: it keeps every call but shrinks the change from the previously applied velocity by a factor $1 - \gamma_{k,i}$, analogous to the damping that stabilizes parallel updates in mean-field inference (Baqué et al., 2016). It targets revisitation; a merging controller thus decides only when to act, and a damping controller where and when (Appendix B.5).

Control signals. Controllers use one of three signals of increasing information and cost, each with a merging rule, a damping rule, and matched controls. *Fixed schedules* use no signal: merging omits late (Tail) or early (Head) intervals, against a uniform grid of the same budget, and damping acts on all generated positions during the first half of sampling (uniform) or releases contiguous position groups from left to right (staggered), against reversed or random release order. *Continuous history* uses recent changes in velocity and prediction memory, requiring neither the omitted call nor decoded readouts. A gate merges once when the merge is predicted to stay close to the fine path: since $Z^{\text{merge}}_{k+2} - Z^{\text{fine}}_{k+2} = \Delta t_{k+1}(v_k - v^{\text{fine}}_{k+1})$, it regresses this discrepancy on diffusion time and positional summaries of

$$\eta_{k,i} = \frac{\|v_{k,i} - v_{k-1,i}\|_2}{\|v_{k,i}\|_2 + 10^{-8}}, \qquad \delta S_{k,i} = \|\widehat{X}_{k,i} - S_{k,i}\|_2, \tag{8}$$

against a preset single merge (Appendix E.1); a selector damps, for one call, positions whose recent prediction-memory path largely cancels out or whose latest memory motion is large. *Categorical readouts* decode the current predictions: a readout gate switches to Tail merging once an online proxy for commitment, recent top-1 stability or the share of low-confidence positions, crosses a threshold, and damping acts, for one call, on positions admitted by the revisitation detector (Equation 9). Both damping selectors are compared with random positions matched in timing, count, and strength.

Table 1: Effects of interval merging and velocity damping on sampling time, prediction dynamics, and quality in unconditional (top) and conditional (bottom) generation.

			LangFlow				Plaid				ELF			
Op.	Signal	Controller	Time	ΔTok%	ΔO	ΔNLL	Time	ΔTok%	ΔO	ΔNLL	Time	ΔTok%	ΔO	ΔNLL
Merge	Fixed	Tail	0.75×	2.40	+2.71	−.016	0.75×	61.51	−1.61	+.093	0.76×	5.08	−0.03	+.020
		Head	0.75×	74.13	−4.46	+.124	0.75×	90.14	−1.99	+.073	0.76×	57.97	+0.63	+.823
		Uniform	0.75×	63.53	—	+0.040	0.75×	81.90	—	+0.037	0.76×	40.60	—	+0.337
		Single	0.99×	2.59	+0.08	−.012	0.99×	13.53	+0.21	+.010	0.97×	0.46	+0.02	−.001
	History	Gate	1.01×	0.00	+0.24	.000	0.98×	15.74	−1.34	.012	0.98×	0.46	−0.15	+.005
	Readout	Gate	0.79×	5.94	+2.80	+.018	0.75×	61.49	−1.86	+.156	1.08×	0.78	+0.01	−.003
Damp	Fixed	Uniform	1.00×	49.66	+0.03	+.122	1.00×	58.18	+0.01	−.023	1.02×	23.68	+0.39	+.083
		Staggered	1.01×	56.62	−10.64	+.108	1.00×	72.64	−0.37	+.072	1.00×	32.59	0.00	+.168
	History	History score	1.02×	21.70	−3.94	+.012	1.00×	15.53	−0.06	+.009	1.07×	0.04	0.00	−.001
		Random	1.02×	20.04	+3.75	+.017	1.00×	11.76	+0.28	+.028	1.07×	0.04	0.00	.000
	Readout	Revisitation	4.69×	8.19	−10.23	+.018	1.45×	8.73	+0.36	+.020	6.42×	3.08	−1.16	−.020
		Random	4.68×	4.57	+2.67	+.025	1.44×	10.27	−0.37	+.012	6.48×	3.86	+1.67	−.018
Both	Fixed	Staggered + Tail	0.75×	56.19	+0.49	+.093	0.75×	76.48	−6.29	+.170	0.76×	32.79	+0.31	+.172
	History	Score + Gate	1.07×	21.28	+2.04	+.003	0.98×	3.5	−3.21	0	1.05×	0.52	0.00	+.003

			ELF-XSum				ELF de→en				Cola-DLM LAMBADA			
Op.	Signal	Controller	Time	ΔTok%	ΔO	ΔRL	Time	ΔTok%	ΔO	ΔBLEU	Time	ΔTok%	ΔO	ΔAcc
Merge	Fixed	Tail	0.75×	1.17	−0.11	+.139	0.75×	0.28	−0.10	+.298	0.76×	6.64	+4.38	−2
		Head	0.76×	15.72	−0.12	−1.962	0.76×	5.52	+0.56	+.116	0.76×	30.37	+0.16	+2
		Uniform	0.75×	9.50	—	−0.226	0.75×	2.71	—	+.055	0.76×	14.79	—	0
		Single	0.97×	0.10	−0.12	.000	0.98×	0.02	+0.09	.000	0.97×	2.86	+4.21	0
	History	Gate	0.98×	0.12	+0.19	.000	0.99×	0.05	0.00	+.184	0.97×	3.03	+2.27	+1
	Readout	Gate	0.70×	1.05	−0.11	+.054	0.67×	0.44	−0.10	+.379	0.72	6.12	−1.51	0
Damp	Fixed	Uniform	1.00×	1.59	0.00	−.039	0.99×	0.56	+0.05	+.045	1.00×	24.90	+5.14	−1
		Staggered	1.00×	1.81	−0.05	−.005	0.99×	0.74	+0.10	+.133	1.01×	31.90	+0.93	0
	History	History score	1.01×	0.12	−3.20	.000	1.01×	0.03	0.00	+.230	1.00×	2.50	+1.65	0
		Random	1.00×	0.10	0.00	.000	1.03×	0.07	0.00	+.192	0.99×	4.74	−1.34	0
	Readout	Revisitation	1.70×	0.27	−3.97	+.027	2.58×	0.10	+16.01	+.046	1.14×	0.00	+9.76	0
		Random	1.70×	0.22	+0.60	+.027	2.63×	0.04	−0.07	+.046	1.13×	0.11	0.00	0
Both	Fixed	Staggered + Tail	0.76×	2.56	−0.13	+.058	0.76×	0.86	+0.17	+.155	0.77×	32.21	+21.33	0
	History	Score + Gate	0.98×	0.17	0.00	.000	1.03×	0.07	0.00	+.230	0.99×	1.23	1.10	+1

5.2 Experiments

Unconditional generation uses 64 paired runs each for LangFlow, Plaid, and ELF; conditional generation uses ELF on XSum (Narayan et al., 2018) and on WMT14 German-to-English (Bojar et al., 2014), to which the ELF-XSum gate, selectors, and thresholds transfer frozen, and Cola-DLM on LAMBADA. Table 1 reports time relative to native, including decision readouts (Time); the percentage of token slots changed from native (ΔTok%); the change in later revisitation (ΔO, 10^{-3}; negative values indicate less revisitation than native); and paired changes in GPT-2-large NLL, ROUGE-L, BLEU, or correct LAMBADA answers (Appendices B.5 and E.4).

Across the six settings, fixed late merging saves about a quarter of sampling time while changing fewer native tokens than matched-cost early merging; adaptive control and velocity damping show benefits in particular settings, but we have not identified a configuration that improves quality and cost consistently across models.

Interval merging. Late merging preserves native outputs far better than early merging: at the same $0.75\times$ cost, Tail changes 0.3–5.1% of tokens in LangFlow, ELF, ELF-XSum, and German-to-

English, against 5.5–74.1% for Head, because late intervals act mostly on committed positions (Section 4.1). The same account explains the exceptions: Plaid, which retains a larger exploration-plus-revisitation share late in sampling (43% at progress .9), remains sensitive (61.5% against 90.1%), and against a same-budget uniform grid, Tail lowers gPPL in ELF and preserves more native tokens with a higher BLEU in translation, whereas LangFlow and Plaid show no reliable ranking; in Cola-DLM, Tail loses two answers where Head gains two, so agreement and accuracy can diverge. Adaptive merging gives setting-dependent results: the Cola-DLM readout gate retains the native number of correct answers at $0.72\times$ sampling time, while the ELF history gate does not establish an advantage over fixed placement. In an ELF native-prefix ablation, history features reduce the local-error RMSE from .315 to .181, yet the deployed gate changes as many tokens as a preset single merge, since output change depends on commitment, which the tested signals capture mainly in retrospect (AUC .867 against .835; Appendix E.2).

Velocity damping. Velocity damping can reduce revisitation, with quality effects that depend on the model and setting. Early schedules change 24–73% of tokens in unconditional generation, like early merging, and staggered release lowers LangFlow revisitation (-10.64) yet raises NLL by .11. Plaid History + Gate combines lower revisitation with a reported NLL change of zero at $0.98\times$ sampling time. Pulse responses vary across models and selectors; in ELF-XSum, most targeted revisitation has ended before intervention (34 of 37 positions; Appendix E.4). In ELF-XSum, preceding damping does not reliably increase accepted merges (44–50 prompts, against 46) and leaves the incremental effect of Tail similar: relative to the same damping without merging, Tail changes 1.07% of tokens after staggered damping, against 1.17% relative to native without damping (4.76% against 5.02% in ELF). Commitment thus helps locate computation that can be saved; translating changes in revisitation into quality and cost gains across models remains an open direction.

6 Discussion

Limitations. Our measurements cover selected configurations of four CDLMs, and mode statistics depend on observation cadence and detection thresholds. Token readouts do not measure correctness (Wang et al., 2026); quality uses automatic metrics. Context interventions cover only LangFlow and ELF; merging is compared with matched placements, not alternative solvers, optimized time grids (Lu et al., 2025; Li et al., 2024), or few-step trained or distilled generators (Hu et al., 2024; Lee et al., 2026). These associations require further evaluation across tasks, generation lengths, and sampling regimes. Local changes in revisitation do not directly quantify update value: assessing it requires paired quality comparisons and accounting for the computational overhead of observing and acting on sampling history.

Future work. Our results suggest combining online stability estimates, as in early-convergence detection (Zhang et al., 2026a), with continuous motion to select intervals to omit, and intervening during ongoing competition, drawing on token-retention and revision policies (Sun et al., 2026; Agarwal et al., 2026). Different commitment times also motivate per-position time schedules (Bansal & Sanghavi, 2026). Inexpensive continuous-state signals could identify intervention opportunities without frequent categorical readouts. Policies should be compared with fixed schedules at matched model evaluations and measured runtime, freezing selection rules before testing on new tasks or sampling budgets. Commitment and revisitation signals may inform training or distillation.

7 Conclusion

We asked what successive updates contribute in continuous diffusion language models. They both advance token stability and keep redistributing candidate support beneath stable top-1 predictions: competition concentrates on a few recurring rivals, and in LangFlow and ELF, the prediction memory of surrounding positions shapes how a revisiting position is revised. Testing their computational meaning, we find that late, largely committed intervals can be omitted with fewer output changes than matched-cost early merging, and the timing results motivate interventions that act while candidate competition is still ongoing. Prediction stability, distributional revision, and the value of each update must therefore be studied together, as our framework allows.

AI USE STATEMENT

We used OpenAI Codex and Claude Code to assist with the literature survey, including suggesting and summarizing candidate related work for the main text and appendices; to revise the manuscript for grammar, wording, and concision; and to prepare figures, including visualization code and schematic layouts. This assistance also included checking bibliographic metadata, identifying missing citations, refining descriptions of related methods. The authors checked suggested references against the original publications and reviewed and revised the text and figures, including the interpretation of cited work and the correspondence between visualizations and the accompanying descriptions. AI-generated suggestions were treated as drafts requiring verification rather than as authoritative sources. The authors take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

All experiments use publicly released checkpoints of LangFlow, Plaid, ELF, and Cola-DLM; native baselines retain the declared update rules, and interventions modify the specified inputs or solver updates. Appendix B.1 lists the configurations and panels. Measurements are defined in Section 3 and Appendix B.3, and the operators, protocols, and frozen thresholds in Section 5.1 and Appendices B.5 and E. Descriptive definitions were developed using exploratory panels; confirmation policies and thresholds were frozen before their evaluation. Reported intervals use document-level bootstrap resampling with fixed seeds, paired for within-document contrasts. The core measurement definitions and their model-independent reference implementation are available at this anonymous repository.

ETHICS STATEMENT

This work analyzes existing language models and datasets. Applications of the studied sampling controls should preserve safeguards against harmful generation and respect model and data licenses.

REFERENCES

Manan Agarwal, Sheel Shah, Chanhyuk Lee, Jaehoon Yoo, Jerry Huang, Seunghoon Hong, Aditi Raghunathan, Jinwoo Kim, and Nicholas M. Boffi. Posterior refinement: Fast language generation via any-order flow maps. *arXiv preprint arXiv:2606.24773*, 2026. URL https://arxiv.org/abs/2606.24773.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Parikshit Bansal and Sujay Sanghavi. Enabling approximate joint sampling in diffusion LMs. *arXiv preprint arXiv:2509.22738*, 2025. URL https://arxiv.org/abs/2509.22738.

Parikshit Bansal and Sujay Sanghavi. Token time continuous diffusion for language modeling. *arXiv preprint arXiv:2607.14106*, 2026. URL https://arxiv.org/abs/2607.14106.

Pierre Baqué, Timur Bagautdinov, François Fleuret, and Pascal Fua. Principled parallel mean-field inference for discrete random fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5848–5857, 2016. doi: 10.1109/CVPR.2016.630. URL https://arxiv.org/abs/1511.06103.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422. URL https://direct.mit.edu/coli/article/48/1/207/107571/.

Victor Besnier, Mickael Chen, David Hurych, Eduardo Valle, and Matthieu Cord. Halton scheduler for masked generative image transformer. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ce29be756db03b0562204265256db075-Abstract-Conference.html.

Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15:9957, 2024. doi: 10.1038/s41467-024-54281-3. URL https://www.nature.com/articles/s41467-024-54281-3.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 12–58. Association for Computational Linguistics, 2014. doi: 10.3115/v1/W14-3302. URL https://aclanthology.org/W14-3302/.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Chang_MaskGIT_Masked_Generative_Image_Transformer_CVPR_2022_paper.html.

Huiwen Chang, Han Zhang, Jarred Barber, Aaron Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Patrick Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip

Krishnan. Muse: Text-to-image generation via masked generative transformers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 4055–4075. PMLR, 2023. URL https://proceedings.mlr.press/v202/chang23b.html.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog Bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2208.04202.

Yuxin Chen, Chumeng Liang, Hangke Sui, Ruihan Guo, Chaoran Cheng, Jiaxuan You, and Ge Liu. LangFlow: Continuous diffusion rivals discrete in language modeling. *arXiv preprint arXiv:2604.11748*, 2026. URL https://arxiv.org/abs/2604.11748.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H. Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, Curtis Hawthorne, Rémi Leblond, Will Grathwohl, and Jonas Adler. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022. URL https://arxiv.org/abs/2211.15089.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359. URL https://aclanthology.org/2021.tacl-1.10/.

Xun Fang, Yunchen Li, Hang Yuan, and Zhou Yu. Factorization-error-free discrete diffusion language model via speculative decoding. *arXiv preprint arXiv:2605.14305*, 2026. URL https://arxiv.org/abs/2605.14305.

Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. Empowering diffusion models on the embedding space for text generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4664–4683. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.261. URL https://aclanthology.org/2024.naacl-long.261/.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-Predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6112–6121. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1633. URL https://aclanthology.org/D19-1633/.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. DiffuSeq: Sequence to sequence text generation with diffusion models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.08933.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1711.02281.

Ishaan Gulrajani and Tatsunori B. Hashimoto. Likelihood-based diffusion language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 16693–16715, 2023. doi: 10.52202/075280-0730. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/35b5c175e139bff5f22a5361270fce87-Abstract-Conference.html.

Hongcan Guo, Qinyu Zhao, Yian Zhao, Shen Nie, Rui Zhu, Qiushan Guo, Feng Wang, Tao Yang, Hengshuang Zhao, Guoqiang Wei, and Yan Zeng. Continuous latent diffusion language model. *arXiv preprint arXiv:2605.06548*, 2026. URL https://arxiv.org/abs/2605.06548.

Satoshi Hayakawa, Yuhta Takida, Masaaki Imaizumi, Hiromi Wakaki, and Yuki Mitsufuji. Demystifying MaskGIT sampler and beyond: Adaptive order selection in masked diffusion. *Transactions on Machine Learning Research*, 2026. URL https://arxiv.org/abs/2510.04525.

Michael Hersche, Nicolas Menet, Ronan Tanios, and Abbas Rahimi. Locally coherent parallel decoding in diffusion language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.20216.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Chunsan Hong, Sanghyun Lee, Chieh-Hsin Lai, Satoshi Hayakawa, Yuhta Takida, Yuki Mitsufuji, Seungryong Kim, and Jong Chul Ye. Understanding and accelerating the training of masked diffusion language models. *arXiv preprint arXiv:2605.13026*, 2026. URL https://arxiv.org/abs/2605.13026.

Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He. ELF: Embedded language flows. *arXiv preprint arXiv:2605.10938*, 2026. URL https://arxiv.org/abs/2605.10938.

Vincent Tao Hu, Di Wu, Yuki M. Asano, Pascal Mettes, Basura Fernando, Björn Ommer, and Cees G. M. Snoek. Flow matching for conditional text generation in a few sampling steps. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 380–392. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.eacl-short.33. URL https://aclanthology.org/2024.eacl-short.33/.

Fei Huang, Tianhua Tao, Hao Zhou, Lei Li, and Minlie Huang. On the learning of non-autoregressive transformers. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9356–9376. PMLR, 2022. URL https://proceedings.mlr.press/v162/huang22k.html.

Zemin Huang, Yuhang Wang, Zhiyang Chen, and Guo-Jun Qi. Don't settle too early: Self-reflective remasking for diffusion language models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5d8bc24ef15b7a808b6dc403c01583e2-Abstract-Conference.html.

Pengyu Ji, Zichen Zhang, Xiang Hu, and Kewei Tu. DA-DLM: Explicitly modeling token dependencies in diffusion language models. *arXiv preprint arXiv:2609.15070*, 2026. URL https://arxiv.org/abs/2609.15070.

Ziqi Jin, Bin Wang, Xiang Lin, Lidong Bing, and Aixin Sun. On the role of discreteness in diffusion LLMs. *arXiv preprint arXiv:2512.22630*, 2025. URL https://arxiv.org/abs/2512.22630.

Jaehyeong Jo and Sung Ju Hwang. Continuous diffusion model for language modeling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 97394–97430, 2025. doi: 10.52202/085713-3256. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/8c4046a93293442e88f54e19a5849d68-Abstract-Conference.html.

Rabeeh Karimi Mahabadi, Hamish Ivison, Jaesung Tae, James Henderson, Iz Beltagy, Matthew Peters, and Arman Cohan. TESS: Text-to-text self-conditioned simplex diffusion. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2347–2361. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.eacl-long.144. URL https://aclanthology.org/2024.eacl-long.144/.

Bumjun Kim, Dongjae Jeon, Moongyu Jeon, and Albert No. DAPD: Dependency-aware parallel decoding via attention for diffusion LLMs. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64624.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M. Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 30749–30768. PMLR, 2025. URL https://proceedings.mlr.press/v267/kim25ah.html.

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M. Boffi, and Jinwoo Kim. Flow map language models: One-step language modeling via continuous denoising. *arXiv preprint arXiv:2602.16813*, 2026. URL https://arxiv.org/abs/2602.16813.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1173–1182. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1149. URL https://aclanthology.org/D18-1149/.

Sanghyun Lee, Seungryong Kim, Jongho Park, and Dongmin Park. Lookahead unmasking elicits accurate decoding in diffusion language models. *arXiv preprint arXiv:2511.05563*, 2025. URL https://arxiv.org/abs/2511.05563.

José Lezama, Huiwen Chang, Lu Jiang, and Irfan Essa. Improved masked image generation with Token-Critic. In *Computer Vision – ECCV 2022*, volume 13683 of *Lecture Notes in Computer Science*, pp. 70–86, 2022. doi: 10.1007/978-3-031-20050-2_5. URL https://link.springer.com/chapter/10.1007/978-3-031-20050-2_5.

Bocheng Li and Linli Xu. Improving few-step language flows with untied self-conditioning. *arXiv preprint arXiv:2608.22244*, 2026. URL https://arxiv.org/abs/2608.22244.

Bocheng Li, Zhujin Gao, Yongxin Zhu, Kun Yin, Haoyu Cao, Deqiang Jiang, and Linli Xu. Few-shot temporal pruning accelerates diffusion models for text generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 7259–7269. ELRA and ICCL, 2024. URL https://aclanthology.org/2024.lrec-main.637/.

Bocheng Li, Zhujin Gao, and Linli Xu. Unifying continuous and discrete text diffusion with non-simultaneous diffusion processes. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11530–11551. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.565. URL https://aclanthology.org/2025.acl-long.565/.

Gen Li and Changxiao Cai. Breaking AR's sampling bottleneck: Provable acceleration via diffusion language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 11700–11725, 2025. doi: 10.52202/085713-0392. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/111298628bfb153ee1f84b10fec3a8b9-Abstract-Conference.html.

Ian Li, Zilei Shao, Benjie Wang, Rose Yu, Guy Van den Broeck, and Anji Liu. Breaking the factorization barrier in diffusion language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.00045.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4328–4343, 2022. doi: 10.52202/068431-0313. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/1be5bc25d50895ee656b8c2d9eb89d6a-Abstract.html.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81. Association for Computational Linguistics, 2004. URL https://aclanthology.org/W04-1013/.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Anji Liu, Oliver Broadrick, Mathias Niepert, and Guy Van den Broeck. Discrete copula diffusion. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/dd1fef536655685898a6602bfbf16857-Abstract-Conference.html.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848. PMLR, 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025. doi: 10.1007/s11633-025-1562-4. URL https://link.springer.com/article/10.1007/s11633-025-1562-4.

Harry Lu. Measuring temporal linguistic emergence in diffusion language models. *arXiv preprint arXiv:2604.23235*, 2026. URL https://arxiv.org/abs/2604.23235.

Omer Luxembourg, Haim Permuter, and Eliya Nachmani. Plan for speed: Dilated scheduling for masked diffusion language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/65445.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206/.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 50608–50646, 2025. doi: 10.52202/085713-1689. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/48b383b24230e0e6e649d9c98dae4d8c-Abstract-Conference.html.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a365e37c18fb91af547a2f0012a89e98-Abstract-Conference.html.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144/.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318. Association for Computational Linguistics, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Avishek Joey Bose, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling. *arXiv preprint arXiv:2502.03540*, 2025. URL https://arxiv.org/abs/2502.03540.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191. Association for Computational Linguistics, 2018. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319/.

Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. CANDI: Hybrid discrete-continuous diffusion models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.22510.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 3363–3377. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.295. URL https://aclanthology.org/2021.eacl-main.295/.

Liran Ringel, Ameen Ali, and Yaniv Romano. Dependency-guided parallel decoding in discrete diffusion language models. *arXiv preprint arXiv:2604.02560*, 2026. URL https://arxiv.org/abs/2604.02560.

Litu Rout, Constantine Caramanis, and Sanjay Shakkottai. Anchored diffusion language model. In *Advances in Neural Information Processing Systems*, volume 38, pp. 89607–89661, 2025. doi: 10.52202/085713-2997. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/8143b8c73073a9a23b9c18e400066471-Abstract-Conference.html.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184, 2024. doi: 10.52202/079017-4135. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T. Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 52584–52619. PMLR, 2025. URL https://proceedings.mlr.press/v267/sahoo25a.html.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454/.

Johannes Schusterbauer, Ming Gui, Yusong Li, Pingchuan Ma, Felix Krause, and Björn Ommer. Denoising, fast and slow: Difficulty-aware adaptive sampling for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2604.19141.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, pp. 103131–103167, 2024. doi: 10.52202/079017-3277.

URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/bad233b9849f019aead5e5cc60cef70f-Abstract-Conference.html.

Yingte Shu, Yuchuan Tian, Chao Xu, Yunhe Wang, and Hanting Chen. Deferred commitment decoding for diffusion language models. *arXiv preprint arXiv:2601.02076*, 2026. URL https://arxiv.org/abs/2601.02076.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265. PMLR, 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Haocheng Sun, Cynthia Xin Wen, and Edward Hong Wang. Why mask diffusion does not work. *arXiv preprint arXiv:2510.03289*, 2025. URL https://arxiv.org/abs/2510.03289.

Xinhao Sun, Huaijin Zhao, Maoliang Li, Zihao Zheng, Jiayu Chen, Yun Liang, and Xiang Chen. STaRR: Spatial-temporal token-dynamics-aware responsive remasking for diffusion language models. *arXiv preprint arXiv:2601.04205*, 2026. URL https://arxiv.org/abs/2601.04205.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. URL https://arxiv.org/abs/2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 147282–147339, 2025. doi: 10.52202/085713-4927. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/d877ea0dd78bdbe54830670618c1de09-Abstract-Conference.html.

Wen Wang, Bozhen Fang, Chenchen Jing, Yongliang Shen, Yangyi Shen, Qiuyu Wang, Hao Ouyang, Hao Chen, and Chunhua Shen. Time is a feature: Exploiting temporal dynamics in diffusion language models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/76931eaba1fcb55b70cde7d0de0161ef-Abstract-Conference.html.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dLLM: Training-free acceleration of diffusion LLM by enabling KV cache and parallel decoding. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5d8d4e6061c3ba96c240b7fa1ae3471d-Abstract-Conference.html.

Kaisen Yang, Jayden Teoh, Kaicheng Yang, Yitong Zhang, and Alex Lamb. Improving sampling for masked diffusion models via information gain. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.18176.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7B: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025. URL https://arxiv.org/abs/2508.15487.

Jaehoon Yoo, Wonjung Kim, Floor Eijkelboom, Chanhyuk Lee, Nicholas M. Boffi, Seunghoon Hong, and Jinwoo Kim. Self-conditioned flow map language models via fixed-point flows. *arXiv preprint arXiv:2607.00714*, 2026. URL https://arxiv.org/abs/2607.00714.

Kevin Zhai, Sabbir Mollah, Zhenyi Wang, and Mubarak Shah. CORE: Context-robust remasking for diffusion language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.04096.

Haohui Zhang, Zhiye Wang, Xiaoying Gan, Xinbing Wang, and Bo Jiang. LEAP: Unlocking dLLM parallelism via lookahead early-convergence token detection. *arXiv preprint arXiv:2605.10980*, 2026a. URL https://arxiv.org/abs/2605.10980.

Runzhe Zhang, Letian Chen, Wenpeng Zhang, Zhouhan Lin, and Peilin Zhao. FlowLM: Few-step language modeling via diffusion-to-flow adaptation. *arXiv preprint arXiv:2605.20199*, 2026b. URL https://arxiv.org/abs/2605.20199.

Tamim Zoabi, Ameen Ali, Liran Ringel, and Lior Wolf. Mean-field parallel decoding for discrete diffusion language models. *arXiv preprint arXiv:2606.15805*, 2026. URL https://arxiv.org/abs/2606.15805.

A Extended related work

A.1 Diffusion language models

Diffusion language models build on denoising diffusion and continuous flows (Sohl-Dickstein et al., 2015; Ho et al., 2020; Lipman et al., 2023; Liu et al., 2023), but continuous time does not imply continuous states. D3PM uses categorical transitions, SEDD learns discrete distribution ratios, and masked formulations predict clean tokens from corrupted sequences (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2025); LLaDA and Dream extend these approaches to large-scale language modeling (Nie et al., 2025; Ye et al., 2025).

Diffusion-LM, DiffuSeq, and CDCD establish continuous approaches to text and categorical data (Li et al., 2022; Gong et al., 2023; Dieleman et al., 2022), with later work refining representation and training choices (Gao et al., 2024; Jo & Hwang, 2025). Plaid emphasizes likelihood-based modeling, TESS uses self-conditioned simplex representations, and NeoDiff permits nonsimultaneous denoising (Gulrajani & Hashimoto, 2023; Karimi Mahabadi et al., 2024; Li et al., 2025). FlowSeq, LangFlow, and ELF develop continuous language flows; FlowLM adapts pretrained diffusion models, and Cola-DLM combines a continuous latent prior with a conditional decoder (Hu et al., 2024; Chen et al., 2026; Hu et al., 2026; Zhang et al., 2026b; Guo et al., 2026). Self-conditioning passes predictions between evaluations (Chen et al., 2023; Karimi Mahabadi et al., 2024) and constitutes the prediction memory whose feedback Section 4.3 tests. Untied Self-Conditioning corrects its coupling with solver updates in LangFlow and ELF without retraining (Li & Xu, 2026). Hybrid approaches such as CANDI combine discrete and continuous corruption, while diffusion-duality and discreteness analyses examine their relationships (Pynadath et al., 2026; Sahoo et al., 2025; Jin et al., 2025). We study LangFlow, Plaid, ELF, and Cola-DLM, which span embedding-space flows (LangFlow, ELF), likelihood-trained embedding diffusion (Plaid), and a latent prior with a conditional decoder (Cola-DLM), all with public checkpoints and native samplers.

A.2 Temporal analysis of diffusion models

Temporal studies examine commitment in LLaDA, theoretical diffusion regimes, token recoverability across noise levels, and intermediate answers that are later overwritten (Lu, 2026; Biroli et al., 2024; Pynadath et al., 2026; Wang et al., 2026). Our modes instead partition position–observation pairs operationally, rather than identify theoretical phases; revisitation measures candidate-support withdrawal and recovery and depends on observation cadence. Following probing studies that distinguish decodability from behavioral relevance (Hewitt & Liang, 2019; Ravichander et al., 2021; Belinkov, 2022; Elazar et al., 2021), we separate readout measurements, intervention effects, and task outcomes.

A.3 Decoding order and revision policies

Masked decoding selects positions using confidence, information gain, lookahead, or decision readiness (Kim et al., 2025; Wu et al., 2026; Yang et al., 2026; Lee et al., 2025; Peng et al., 2025; Luxembourg et al., 2026; Shu et al., 2026; Zhang et al., 2026a), while refinement and remasking revise provisional text (Lee et al., 2018; Ghazvininejad et al., 2019; Wang et al., 2025; Huang et al., 2026; Zhai et al., 2026). Related policies govern visual tokens (Chang et al., 2022; 2023; Besnier et al., 2025; Hayakawa et al., 2026; Lezama et al., 2022), while NeoDiff and PatchForcing vary position-specific denoising progress (Li et al., 2025; Schusterbauer et al., 2026). Token-selection policies make explicit decisions about which predictions to retain or revise, including STaRR's remasking and Posterior Refinement's retain-or-regenerate decisions (Sun et al., 2026; Agarwal et al., 2026). In the continuous samplers studied here, revisitation measures candidate-support motion without selecting tokens for revision; damping and merging act on the continuous update rather than on discrete token choices.

A.4 Factorization and token dependencies

Parallel choices can be individually plausible but jointly incompatible (Gu et al., 2018; Huang et al., 2022). Factorized outputs do not directly model dependence among newly sampled values (Liu et al., 2025; Li et al., 2026), although the final multi-step sequence distribution need not factorize

(Jin et al., 2025). Discrete Copula Diffusion, CoDD, and DA-DLM model joint outputs through copulas, circuits, or graphs, while ADJUST, CoDiLA, and FeF-DLLM use conditioned sampling (Liu et al., 2025; Li et al., 2026; Ji et al., 2026; Bansal & Sanghavi, 2025; Hersche et al., 2026; Fang et al., 2026); dependency-aware schedules coordinate update sets (Kim et al., 2026; Ringel et al., 2026; Zoabi et al., 2026). Anchoring, information-gain, and context-locality studies examine cross-position influence (Rout et al., 2025; Yang et al., 2026; Sun et al., 2025; Hong et al., 2026). We probe this coordination in continuous models by altering context and prediction feedback at other positions and measuring a focal position's response (Section 4.3).

A.5 Sampling budgets and update control

For fixed generators, higher-order solvers reduce integration error through more accurate approximations, allowing fewer evaluations (Lu et al., 2025), and Few-shot Temporal Pruning searches nonuniform grids with validation data (Li et al., 2024); both keep one time grid for all positions. Training and distillation instead learn few-step generators through FlowSeq, flow maps, and compressed self-conditioning iterations (Hu et al., 2024; Lee et al., 2026; Yoo et al., 2026), while Token Time Continuous Diffusion assigns position-specific times (Bansal & Sanghavi, 2026). Interval merging keeps the shared grid and frozen weights, and asks which intervals can be removed given commitment rather than local error. Damping of simultaneous updates has a precedent in parallel mean-field inference (Baqué et al., 2016) and dependency-aware mean-field decoding (Zoabi et al., 2026); velocity damping applies it per position to the continuous update, without selecting tokens.

B Experimental Setup and Measurement Details

B.1 Models and native sampling

Table 2 summarizes the main panels.

Model / checkpoint	Task / evidence	Output slots	Updates	Probability channel
LangFlow, OWT	Unconditional; 64 draws	128	127	Native prediction logits
Plaid, fixed weights	Unconditional; 64 draws	128	127	Native logits; extra terminal readout
ELF-B-OWT	Unconditional; 64 draws	128	32	Decoder of memory S
ELF-XSum, 39,800 EMA1	XSum; 64 reused rows per panel	64	32	Decoder of memory S
Cola-DLM, DiT + VAE	LAMBADA; 64 prompts (Figure 3)	32 requested	2×16	VAE readout of Z, before penalty
Cola-DLM, DiT + VAE	Unconditional; 64 draws (candidate analysis)	128	8×16	VAE readout of Z, before penalty

Table 2: Main observation panels, sampling budgets, and readout channels; cohort reuse is specified in Appendix B.2.

LangFlow. Length-128 generation uses 128 evaluations and 127 Euler–EDM updates; native logits give probabilities and embedding predictions give self-conditioning. Outputs are highly repetitive in this shorter-than-published configuration (1,024 tokens): the 128 nonempty, EOS-free slots contain only 4–39 distinct tokens, which bounds the scope of the LangFlow statistics rather than evaluating fluency.

Plaid. Length-128 ancestral sampling uses score temperature .9, noise scale 1, FP64 states, FP32 inputs, and BF16 Transformer computation. The 128 loop evaluations make 127 updates; a terminal evaluation refreshes memory and reads tokens without an update. Raw clean predictions supply S, temperature-adjusted predictions supply the solver.

ELF. ELF-B-OWT uses 32 uniform intervals, CFG1/SC1, and noise scale 2; ELF-XSum uses checkpoint step 39,800, EMA1, CFG2/SC1, and 32 native intervals. ELF German-to-English uses

the ELF-B WMT14 checkpoint with CFG2 and 32 guided intervals (65 native evaluations) on 64 examples unused in earlier development; BLEU (Papineni et al., 2002) uses sacreBLEU (Post, 2018) (13a, lowercased), with native sentence and corpus BLEU of 31.65 and 32.52. Diagnostics decode S at decoder time and self-conditioning scale 1; terminal $\mathcal{D}_\theta(Z)$ defines final-body masks but is not appended to the identity history.

Cola-DLM. Unconditional draws start with empty external prefix and caches and generate eight 16-token blocks with 16 Euler intervals each, temperature zero, repetition penalty 1.1, and no early stopping. Guidance is 1 in block 0 and 7 thereafter; blocks share generated history and are not independent documents. The composition panel instead uses 64 LAMBADA continuations with two blocks and CFG7, preserving native overlap and caches. States use FP32 with BF16 autocast; diagnostics decode pre-penalty probabilities and verify terminal logits against native readout. Completed blocks provide history KV, not an ELF-style S buffer.

Quality metrics. Unconditional text is scored by NLL under GPT-2-large (Radford et al., 2019); summaries are scored by ROUGE-L (Lin, 2004). Translation uses BLEU as specified above, and LAMBADA uses answer accuracy.

B.2 Sample splits, reuse, and checkpoint identity

Tables 3 and 4 specify sample reuse. The policy study uses disjoint 32/32/64 development/calibration/confirmation splits. Policies are fitted and calibrated on the first two splits and frozen before confirmation; diagnostics and policies share samples within each split. Descriptive thresholds were set on native development histories.

Panel	Size	Purpose and sample overlap	Location
Native dynamics	64/model; Cola +192	Shared descriptive/candidate predictions: LangFlow/Plaid 16+48, ELF 64. Cola recurrence additionally checks 192 new draws, separately by block; composition uses earlier LAMBADA continuations.	D.1–D.2
Budget and cadence	LangFlow, Plaid, ELF: 64 each	Paired budgets with matched and dense observations.	D.2
Cola integration controls	32/task	Current-detector comparison pools 64 prompts with task-stratified resampling.	D.3
Context hold	LangFlow, ELF: 16+64 each	Separate development/evaluation per model; 64/64 evaluation triggers; no pooling.	D.5
Native history screen	ELF-XSum 64	Selector-development prompts; predicts future native revisitation without intervention.	E.4
Feedback pathway	LangFlow 64; ELF-XSum 32+64	LangFlow: 49 triggers. ELF: 18/47 triggers in disjoint panels.	D.4
ELF-XSum velocity damping	64 prompts; 29–61 events	Pulses, schedules, and merging share prompts, noise, and native outputs.	E.4
Cola velocity controls	32 LAMBADA prompts	History/readout pulses, matched random controls, and schedules; common paired native.	B.5

Table 3: Observation and intervention panels, with sample reuse and operator identity distinguished.

B.3 Prediction dynamics measurements

Generated-position measurements exclude source and prompt-overlap positions. Final-body masks precede the first native terminal EOS; current-body masks use the window's last observation, and whole-window masks exclude positions following any EOS in that window. Without EOS, all generated positions remain. Commitment uses the terminal identity of the declared readout channel; LangFlow, Plaid, and Cola-DLM use raw-logit identities. Probability reductions renormalize full-vocabulary softmax.

Panel	Size	Purpose and sample overlap	Location
Cost panel	ELF-XSum 16+48	Reused development prompts; paired native/grid comparisons.	E.2
Policy development and evaluation	ELF-XSum 32/32/64	Disjoint splits; diagnostics/policies share each split.	E.1, E.3
Fixed-grid placement	ELF-XSum 64+64	Reuses cost/confirmation panels for placement controls.	E.2
Unconditional grids	64/model	Reused LangFlow/Plaid/ELF seeds; timing: first 16, five repeats.	E.2
Fixed-grid confirmation	ELF/Cola: 64 each	New ELF noise/Cola LAMBADA sources; frozen policies; timing: 16, three repeats.	E.2
Cola task grids	32 prompts	Reused LAMBADA sources, four grids; three warm repeats; engineering excluded.	E.2
Unconditional online gates	LangFlow, ELF: 32 each	Development policies; quality: 32, timing: 16.	E.3
Stability-triggered merging	LangFlow, Plaid: 64 each	Independent; fixed/shuffled controls; timing: 16, three repeats.	Table 10
Shared controller panel	ELF-XSum 64 new prompts	Readout/history policies, pulses, schedules, and combinations share prompts/native outputs; timing: 16.	Table 10
Plaid single merge	64 runs	Paired output/quality panel; timing: first 16, three warm repeats.	E.2
Cross-setting velocity controls	64/setting; Cola 32	Shared native/pulse/schedule panels; separate LangFlow history timing on 16 runs, three repeats; Cola controls use their current paired native.	E.4
Combined-policy dynamics	LangFlow: 16 primary; others: 32; 64 extension	LangFlow, Plaid, ELF-OWT, ELF-XSum, translation; primary is a fixed prefix of each reused 64-run policy panel.	E.4

Table 4: Sampling-policy panels and their relationships; quality and timing sample sizes are distinguished.

B.4 Supplementary measurements and aggregation

For amplitude-gated interventions, let j be the earliest reference maximizing $O_{a:b,i}$ with excursion floor 10^{-6}, and $E_{j:b,i}$ its excursion. The selection rule is zero when $O = 0$; otherwise

$$\chi_{a:b,i}(\boldsymbol{\xi}) = \mathbf{1}\{O_{a:b,i} \geq \xi_O, \quad E_{j:b,i} \geq \xi_E, \quad h(p_{b-1,i}, p_{b,i}) \geq \xi_h\}\,. \tag{9}$$

The default is $\boldsymbol{\xi} = (.5, .1, .05)$. Descriptive detection α instead uses threshold $1/3$ and floor 10^{-3}. Descriptive coverage follows Equation 5; online intervention waits for detection, which need not imply motion in the final update.

Total probability-path length $\ell_{a:b} = \frac{1}{2}\sum_{n=a+1}^{b} \|p_n - p_{n-1}\|_1$ measures accumulated probability motion.

Metrics are averaged over eligible positions or position–window pairs within each document, then equally across documents. Paired contrasts average within-document differences, and 95% intervals resample whole documents while retaining paired arms.

B.5 Intervention protocols

Velocity damping. Equation 6 initializes $\widetilde{v}_0 = v_0$, keeps source/prompt strength zero, and retains native predictions and memory refreshes. Pulses apply $\gamma_0 = .5$ for one call at $b+1$; schedules apply $.25$ over the first half of calls or $.5$ to four contiguous groups with staggered, reversed, or random release. Plaid modifies deterministic drift only, replaying identical noise across arms. In interval merging, Plaid, whose ancestral update has no explicit velocity, reuses the prediction $\widehat{X}_k$ over the combined interval under its native update rule. In Cola-DLM, which has no prediction memory, history selectors use velocity history instead. Merging reuses the applied damped velocity; realized

dose is $\|\Delta t_k(\widetilde{v}_k - v_k)\|_2$ over affected positions. The history selector uses five memory states:

$$P_{k,i} = \sum_{j=k-3}^{k} \|S_{j,i} - S_{j-1,i}\|_2, \qquad R_{k,i} = \frac{P_{k,i} - \|S_{k,i} - S_{k-4,i}\|_2}{P_{k,i} + 10^{-8}}. \tag{10}$$

R measures cancellation, not categorical revisitation; thresholds on R, P, and last motion $M_{k,i} = \|S_{k,i} - S_{k-1,i}\|_2$ are frozen on development data, with ranking by R or M at the first eligible boundary. Readout selection uses the first window satisfying Equation 9 at its default thresholds.

Cola-DLM velocity controls. The 32-prompt LAMBADA panel pairs all arms with native sampling in the same execution environment, retaining prompt constraints and block boundaries. History and readout selectors apply one strength-.5 velocity pulse, with random positions matched in admission time, count, and strength. Both pulse selectors act only in the first block; uniform and staggered schedules act in both 16-interval blocks, and Tail omits indices 9, 11, 13, and 15 in each. Pulse outcomes use selected positions in the first fully released $W = 4$ window ending at $b + 5$, assigning zero to nontriggers; schedule outcomes average calls 13–16 over generated positions in the first block. Warm timing uses the first 16 prompts with three balanced repeats, excluding passive diagnostics but including online selection. Coefficients are matched, but realized vector doses need not be.

Interval merging. Paired branches start from the same full state at the candidate boundaries specified in Appendix E.1. Two fine Euler updates are compared with one over their summed increment, then continued natively to the same terminal time. Fixed 28 merges pairs starting at zero-based indices 20, 22, 24, and 26, preserving the final interval. Online decisions use completed calls only; all policies preserve source constraints, diffusion-time coordinates, and block boundaries. For samplers other than deterministic Euler, grids remove the same evaluation points while keeping each native update rule, guidance, and source constraints.

B.6 Validation, uncertainty, and computational cost

Observer validation compares outputs, states, RNG state, and caches against native execution on separate implementation-test samples and measurement runs, including budget controls. Interventions undergo operator-specific replay checks. Unless stated otherwise, intervals use 2,048 paired whole-document resamples; LangFlow and Plaid budget contrasts use 4,096.

Diagnostics reuse native logits where available or add decoder calls; their reductions and tracing costs are excluded from deployment timing unless required by the online policy.

Timing comparisons are paired within device; individual panels specify hardware and timing scope.

C Mathematical Properties and Derivations

C.1 Observed commitment and the observation clock

The terminal observation guarantees existence of stable commitment. Writing $C_{n,i} = \mathbf{1}\{n \geq \tau_i^{\text{stable}}\}$ and setting $\alpha_{N+1,i} = \alpha_{N+2,i} = 0$ gives the equivalent partition

$$\begin{aligned} Q^{\mathsf{R}}_{n,i} &= 1 - \prod_{r=0}^{2}(1 - \alpha_{n+r,i}), \\ Q^{\mathsf{C}}_{n,i} &= C_{n,i} \prod_{r=0}^{2}(1 - \alpha_{n+r,i}), \\ Q^{\mathsf{E}}_{n,i} &= (1 - C_{n,i}) \prod_{r=0}^{2}(1 - \alpha_{n+r,i}). \end{aligned} \tag{11}$$

A detection covers $n-2, n-1, n$; incomplete windows supply no detection. The product assigns revisitation first, then C separates commitment from exploration, making the indicators disjoint and exhaustive. At N, $C_{N,i} = 1$ but $Q^{\mathsf{C}}_{N,i} = 1 - \alpha_{N,i}$.

Now retain $\mathcal{N}_c \subseteq \mathcal{N}$ with the same terminal observation, masks, and retained readouts. Observed stable time has no fixed ordering under subsampling: subsampling clocks $(0, 1, 2)$ to $(0, 2)$ moves stability from 2 to 0 for identities $(\mathtt{a}, \mathtt{b}, \mathtt{a})$, but from 1 to 2 for $(\mathtt{b}, \mathtt{a}, \mathtt{a})$. Cadence controls therefore concern observed commitment (Appendix D.3); changing the readout channel, terminal identity, or EOS mask changes the estimand.

C.2 Relative revisitation and window size

For reference j, let $h_{j,b} = h(p_j, p_b)$ and $E_{j:b} = \max_{j<n\le b} h(p_j, p_n)$. Since the maximum includes b, $0 \le h_{j,b} \le E_{j:b}$; for $E_{j:b} > \epsilon \ge 0$,

$$1 - \frac{h_{j,b}}{E_{j:b}} = \frac{E_{j:b} - h_{j,b}}{E_{j:b}} \in [0, 1]. \tag{12}$$

This recovered radial fraction equals one at exact recovery and zero at maximal distance, irrespective of eligible excursion size. For (p, p', p, p) with $h(p, p') > \epsilon$, $O = 1$ and $\alpha = 1$ despite a stationary final update. Detection thus concerns the window, whereas the amplitude-gated selector also requires final-step motion.

Extending $a{:}b$ backward to $a'{:}b$ leaves existing reference suffixes, eligibility, and scores unchanged while adding candidates, so

$$O^{\epsilon}_{a':b} \ge O^{\epsilon}_{a:b}. \tag{13}$$

Thresholding preserves this ordering; changing the maximizing reference can nevertheless change its excursion and amplitude-gated eligibility.

C.3 Velocity damping and realized displacement

From Equation 6,

$$\widetilde{v}_{k,i} - \widetilde{v}_{k-1,i} = (1 - \gamma_{k,i})(v_{k,i} - \widetilde{v}_{k-1,i}).$$

Damping contracts the change from the previously applied velocity by $1 - \gamma_{k,i}$ without necessarily reducing the velocity norm; the state-proposal displacement is $|\Delta t_k|\gamma_{k,i}\|\widetilde{v}_{k-1,i} - v_{k,i}\|_2$. For deterministic Euler and $\Delta t_{k-1} \neq 0$, substituting $\widetilde{v}_{k-1,i} = (Z_{k,i} - Z_{k-1,i})/\Delta t_{k-1}$ shows that damping mixes the native proposal with

$$R^{Z}_{k+1,i} = Z_{k,i} + \frac{\Delta t_k}{\Delta t_{k-1}}(Z_{k,i} - Z_{k-1,i}).$$

This equals $2Z_{k,i} - Z_{k-1,i}$ only for equal signed increments. Stochastic displacement includes noise and cannot replace stored deterministic velocity; Plaid damps only that velocity component. Calls and native memory refreshes remain; changed accepted states can affect later predictions even if the current readout is unchanged.

C.4 One merged interval versus two native intervals

Merging as velocity blending with $\gamma = 1$. For deterministic Euler with memory refresh $S_{k+1} = \widehat{X}_k$, setting $\gamma_k = 0$ and $\gamma_{k+1} = 1$ at all generated positions yields $\widetilde{v}_{k+1} = \widetilde{v}_k = v_k$ and the merged update in Equation 7. Skipping call $k + 1$ also skips its memory refresh, leaving $S_{k+2} = \widehat{X}_k$; damping with $0 < \gamma < 1$ retains both.

Consider deterministic Euler branches starting from identical complete sampler states, diffusion time, and conditioning, after any intervention has ended. Both first return $(v_k, \widehat{X}_k)$. The fine branch sets $Z^{\mathrm{fine}}_{k+1} = Z_k + \Delta t_k v_k$ and $S^{\mathrm{fine}}_{k+1} = \widehat{X}_k$, then evaluates $(v^{\mathrm{fine}}_{k+1}, \widehat{X}^{\mathrm{fine}}_{k+1})$ and obtains

$$Z^{\mathrm{fine}}_{k+2} = Z_k + \Delta t_k v_k + \Delta t_{k+1} v^{\mathrm{fine}}_{k+1}, \qquad S^{\mathrm{fine}}_{k+2} = \widehat{X}^{\mathrm{fine}}_{k+1}.$$

Subtracting from the merged branch gives

$$\begin{aligned} Z^{\mathrm{merge}}_{k+2} - Z^{\mathrm{fine}}_{k+2} &= \Delta t_{k+1}(v_k - v^{\mathrm{fine}}_{k+1}), \\ S^{\mathrm{merge}}_{k+2} - S^{\mathrm{fine}}_{k+2} &= \widehat{X}_k - \widehat{X}^{\mathrm{fine}}_{k+1}. \end{aligned} \tag{14}$$

These local discrepancies use future outputs available only offline.

For ELF generated coordinates,

$$a_k = \max(1 - t_k, \epsilon_t), \qquad v_k = \frac{\widehat{X}_k - Z_k}{a_k}, \qquad \widehat{X}_k = Z_k + a_k v_k, \tag{15}$$

where $\epsilon_t > 0$ is the denominator floor. Guidance preserves this relation by applying the same affine weights to both outputs at shared state and time. Substituting the first Euler update yields

$$\begin{aligned} \widehat{X}_k - \widehat{X}^{\text{fine}}_{k+1} &= (a_k - \Delta t_k) v_k - a_{k+1} v^{\text{fine}}_{k+1} \\ &= a_{k+1}(v_k - v^{\text{fine}}_{k+1}) + (a_k - \Delta t_k - a_{k+1}) v_k. \end{aligned} \tag{16}$$

If $1 - t_k \geq \epsilon_t$ and $1 - t_{k+1} \geq \epsilon_t$, the final coefficient vanishes. Writing ΔZ and ΔS for merged-minus-fine state and memory differences,

$$\Delta S = (1 - t_{k+1})(v_k - v^{\text{fine}}_{k+1}) = \frac{1 - t_{k+1}}{\Delta t_{k+1}} \Delta Z, \qquad \Delta t_{k+1} \neq 0. \tag{17}$$

The two local discrepancies are proportional, with matching positional norm ranks when the factor is common and nonzero. Across a native Euler step, the same unfloored relation gives $\widehat{X}_k - \widehat{X}_{k-1} = (1 - t_k)(v_k - v_{k-1})$. Since $S_k = \widehat{X}_{k-1}$, $\delta S_{k,i}$ ranks like unnormalized $\|v_{k,i} - v_{k-1,i}\|_2$. For normalized η, instead $\delta S_{k,i} = (1 - t_k)(\|v_{k,i}\|_2 + 10^{-8})\eta_{k,i}$, whose position-dependent factor need not preserve ranks. An active floor requires the additional term in Equation 16.

The identities assume generated coordinates, native source constraints, and deterministic Euler; stochastic updates, higher-order solvers, and block caches require separate derivations. Merging m disjoint pairs removes m solver evaluations (guidance/readouts counted separately) and changes observation spacing.

D Supplementary Experiments on Prediction Dynamics

Configurations and measurement rules are in Appendix B; descriptive analyses use α, while explicitly identified amplitude-gated analyses retain χ.

D.1 Commitment, probability motion, and mode composition

Motion after commitment. Table 5 uses 64 runs per model, with Cola-DLM retaining its LAMBADA continuation cohort. Each document's postcommitment fraction sums probability-path length strictly after commitment, excluding the entering edge, and divides by its total path length. The separate unconditional Cola-DLM fractions are $30.964\ [29.483, 32.429]\%$ and $18.214\ [16.916, 19.464]\%$ for blocks 0 and 1.

Table 5: Document-mean probability-path fractions after commitment (%), with 95% bootstrap intervals; Cola-DLM uses the LAMBADA continuation cohort.

Model	After commitment
LangFlow	$26.558\ [24.465, 28.767]$
Plaid	$12.830\ [12.486, 13.193]$
ELF-base	$3.889\ [3.714, 4.065]$
Cola-DLM, block 0	$20.547\ [18.197, 22.987]$
Cola-DLM, block 1	$18.716\ [17.545, 19.816]$

Detection and coverage. Table 6 gives complete-panel detection rates, position reach, and document reach: rates average over eligible $W = 4$ windows within documents, whereas reach counts any qualifying detection. The detector is $O^{\epsilon} > 1/3$, $\epsilon = 10^{-3}$, with the floor applied per reference before maximization; descriptive coverage assigns $n - 2, n - 1, n$ with revisitation priority. LangFlow, Plaid, and ELF use 2,048 document-bootstrap resamples, stratified by collection stage where applicable; unconditional Cola-DLM uses 4,096 whole-draw resamples. Cola-DLM's table rows are unconditional; the displayed continuation first block instead reaches 63/565 positions and 33/64 prompts. Thresholds were frozen before the new unconditional Cola-DLM draws.

Table 6: Detection rates under $O^\epsilon > 1/3$ with $\epsilon = 10^{-3}$, with document-mean rates and 95% bootstrap intervals.

Model	Detections (%)	Positions	Docs.
LangFlow	6.13 [5.35, 7.02]	6,557/8,192	64/64
Plaid	4.01 [3.89, 4.12]	7,466/8,192	64/64
ELF	1.55 [1.46, 1.64]	2,030/8,192	64/64
Cola-DLM block 0	0.17 [0.09, 0.27]	21/1,024	15/64
Cola-DLM block 1	1.93 [1.57, 2.33]	179/1,024	60/64

D.2 Candidate structure and observation controls

Certified contests. The Cola-DLM candidate analysis uses a separate 64-draw unconditional panel. Episodes group detections until four observable inactive endpoints; missing observations and termination censor unfinished episodes. The reference precedes the first detection by three observations; its leading token is tracked to its first minimum through the observed episode end, including quiet closing observations. Exact geometry requires that token in every top-10 cache and a loss above 10^{-6}. The rival maximizes gain among candidates cached at both endpoints; certification requires its gain to exceed all incompletely observed candidates' upper bounds within 10^{-7}, using $[0, p_{10}]$ for missing probabilities. Of 21,738, 23,333, and 2,292 episodes in LangFlow, Plaid, and ELF, 132, 1,523, and 222 are censored; Cola-DLM blocks 0/1 have 21/183 episodes with 5/118 censored. Table 7 gives certified counts and document-mean statistics, using the bootstrap rules above. Large contests require reference excursion $\geq .1$ and leading-token loss $\geq .02$. Recurrence uses consecutive eligible contests at the same position with the same leading token, restarting after missing geometry, uncertified rivals, or nonpositive loss; neither Cola-DLM block supplies an eligible pair in the original 64-draw panel. At the leading token's minimum, 87.4%, 69.1%, and 72.7% of certified LangFlow, Plaid, and ELF contests are at or after commitment; this need not hold throughout the episode. An expansion adds 192 draws to the original 64 with the same first-block analysis, yielding 86 episodes and 57 certified positive-loss contests across 256 draws but still no eligible consecutive pairs. In the second block, the expansion supplies seven eligible pairs from seven draws, with recur-

Table 7: Full candidate statistics: all and large certified contests, with document-mean estimates and 95% bootstrap intervals.

Model	Subset	Contests / documents	Eligible pairs / documents	Rival takes $> 1/2$ (%)	Pair mass (%)	Same rival (%)
LangFlow	All	10,980 / 64	4,609 / 64	86.0 [84.3, 87.7]	94.2 [93.5, 94.9]	76.1 [73.8, 78.4]
	Large	663 / 56	88 / 29	63.4 [57.7, 69.1]	74.7 [71.9, 77.4]	48.5 [31.4, 63.6]
Plaid	All	11,771 / 64	3,743 / 64	79.2 [78.2, 80.0]	74.9 [74.1, 75.6]	50.6 [48.6, 52.6]
	Large	4,746 / 64	968 / 64	78.5 [77.3, 79.8]	59.2 [58.3, 60.1]	52.8 [49.4, 56.0]
ELF	All	1,507 / 64	95 / 46	98.2 [97.4, 98.9]	98.4 [98.1, 98.7]	69.1 [58.3, 78.9]
	Large	617 / 64	12 / 10	97.6 [96.1, 98.9]	96.9 [96.3, 97.5]	95.0 [85.0, 100.0]
Cola-DLM	All	13 / 10	0 / 0	60.0 [30.0, 85.0]	85.5 [69.5, 95.7]	–
	Large	5 / 4	0 / 0	62.5 [25.0, 100.0]	68.8 [37.3, 91.0]	–

Large contests have excursion $\geq .1$ and a leading-token loss $\geq .02$. Pair mass is the combined probability of the leading token and the rival at the leading token's lowest point. Recurrence compares consecutive eligible contests at the same position with the same leading token. Counts are pooled; estimates average within contributing documents, then across documents, so pooled count ratios need not match the reported percentages. Cola-DLM uses its unconditional first block and has no eligible recurrence pairs ("–").

rence 57.1 $[14.3, 85.7]\%$; this history-conditioned result is reported separately from the first-block analysis.

Plaid budget and cadence. Sixty-four paired draws compare 128 and 512 loops, keeping model, length, temperature, and precision fixed but not coupling differently timed ancestral noise paths. Coarse observations of the 512-loop path retain indices $\iota(j) = \mathrm{round}(511j/127)$, $j = 0, \ldots, 127$, and terminal index 512, approximating the native grid within half a fine interval without interpolation. Statistics and masks are recomputed at each cadence; commitment uses progress $1 - t$ with both terminal observations at one. With the current detector and matched 129 readouts, the detection rate rises from .039 to .052 ($+.013$ $[.011, .015]$), while mean commitment progress changes by $-.010$ $[-.025, +.003]$.

ELF budget and cadence. On 64 paired noise draws, ELF uses 128 instead of 32 uniform intervals, observed densely or every fourth update on the original grid. Dense observation raises position reach by 15.34 $[14.05, 16.59]$ percentage points relative to coarsened observation, whereas coarse–native differs by -1.25 $[-2.76, .26]$ points. At matched cadence, commitment advances (Figure 6c), while single-rival concentration remains high across native, dense, and coarse observations. More intervals lower gPPL but raise repetition and lower entropy.

LangFlow budget and cadence. On 64 paired noise draws, LangFlow runs 512 instead of 128 Euler–EDM steps, observed densely or every fourth update to match the native cadence. At matched cadence, commitment advances while detection and position-reach intervals include zero; Figure 6c–d reports the commitment and detection contrasts. Dense observation instead raises detection by .145 $[.133, .159]$ and position reach by .147 $[.124, .171]$, demonstrating sensitivity to observation cadence.

D.3 Step-size controls in Cola-DLM

Paired 16/32-interval runs share initial noise on 32 LAMBADA and 32 SIQA prompts, generating two 16-token blocks with CFG7, temperature zero, repetition penalty 1.1, and no EOS stopping. Commitment and revisitation use common-eight observations $\kappa(n) = Kn/8$, $n = 1, \ldots, 8$, without interpolation or mask changes. Under the current detector, the equal-task mixture reduces detection from 3.70% to 3.07% ($-.628$ $[-1.040, -.238]$ percentage points) and advances mean commitment progress by .0128 $[.0064, .0189]$ at 32 rather than 16 intervals. Both tasks have the same direction; intervals use 4,096 task-stratified paired prompt resamples, averaging positions and blocks within each prompt. Doubling intervals preserves all normalized LAMBADA answers and official SIQA choices. These estimates supply Figure 6c–d.

D.4 Returning feedback

A provider is a perturbed generated position; recipients are the other generated positions. A one-call provider-memory perturbation ($\gamma_0 = .5$, event boundary $b \leq 25$) creates native and changed branches. At the next boundary, provider/recipient carriers independently come from the native (0) or changed (1) branch, giving $00, 10, 01, 11$ with native source coordinates. Contrasts $01 - 00$ and $11 - 10$ replace recipients while holding the provider fixed and measure next-provider prediction L2 distance. Figure 8 reports the excess response over an equal-dose random provider perturbation for joint (Z, S), S-only, and Z-only replacements. The panels contain 32 reused prompts (18 triggers) and 64 different source articles with new noise (47 triggers). Channels share events; all-prompt estimates assign zero to nontriggers. Native/sham replay and fixed-provider checks pass, but recipient S/Z doses differ (mean L2 ratios 24.11/24.72), precluding intrinsic channel ranking. Joint replacement has positive aligned–random excess in both panels; the replicated memory-only interval includes zero, despite a positive direct S-minus-Z contrast.

The independent LangFlow evaluation contains 64 unconditional runs and 49 triggers, with a single provider and all other generated positions as recipients. For native-provider replacements, trigger-conditioned joint and memory-only aligned–random prediction responses are .115 $[.017, .232]$ and .144 $[.023, .282]$; memory-only excess also remains positive with the perturbed provider input fixed. Relative-dose-matched replacements rescale recipient directions to a shared relative norm across

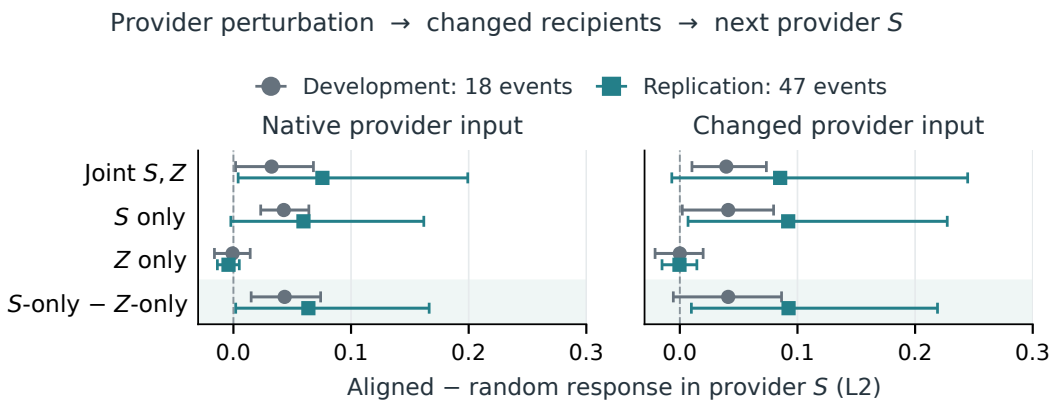


Figure 8: Recipient-state replacements reveal a feedback pathway to the next provider prediction in ELF, with paired 95% intervals and the direct S-only minus Z-only contrast shaded.

carriers and perturbation directions without extrapolation; the corresponding joint and memory-only intervals include zero.

D.5 Context holds at detected revisitation

Both models use the current detector ($W = 4$, per-reference excursion $> 10^{-3}$, $O > 1/3$), selecting the largest-O generated position at the first eligible observation n, with ties resolved by position index. All branches share call $n + 1$; before call $n + 2$, Hold branches restore the surrounding S, Z, or both to their values entering $n + 1$, preserving the selected position's native inputs, time, and RNG state. The outcome is the hold-minus-native change in $h(p_{\text{probe}}, p_{n-1}) - h(p_{\text{probe}}, p_n)$; the next $W = 4$ score includes the shared observation at $n + 1$. Each model uses four implementation checks, 16 development draws, and 64 independent evaluation draws with frozen rules and 4,096 paired document-bootstrap resamples. Checks verify native-prefix/sham replay, held values, and unchanged target inputs; all evaluation draws trigger with anchor separation above .001 and no missing outcomes.

LangFlow uses 128 slots and calls, reading p_{probe} directly from forward logits. ELF uses 128 generated positions, 32 intervals, and CFG1/SC1; its fresh denoiser prediction is decoded at time 1 with zero decoder memory. Thus, the ELF probe measures the composite denoiser/readout pathway, with fixed focal denoiser inputs but potentially changed predicted-clean output. Figure 6a–b shows that holding S shifts preference toward p_n in both models, while Z-only intervals include zero; holding both has a positive mean, with uncertainty spanning zero in ELF. All six next-window O intervals include zero, so the directional effects do not establish reduced subsequent revisitation.

E Supplementary Experiments on Sampling Controls

Appendix B specifies the operators and frozen selection rules.

E.1 Continuous-history gate: calibration and ablation

Disjoint 32-prompt development and calibration panels compare fine/merged branches at $k \in \{12, 16, 20, 24, 28\}$. The original shared mask retains $k > b + 2$ after perturbation admission at $b \leq 22$, or all boundaries without admission, giving 156 native branches per split. For $U \in \{Z, S\}$, the generated-position error is

$$e_U = \frac{\text{RMS}(U_{k+2}^{\text{merge}} - U_{k+2}^{\text{fine}})}{\max\{\text{RMS}(U_{k+2}^{\text{fine}} - U_k), 10^{-8}\}}, \qquad e_{\text{merge}} = \max\{e_Z/.00966728, e_S/.621673\},$$

where the scales are native calibration medians.

Ridge regression predicts $\log(e_{\text{merge}} + 10^{-8})$ from diffusion time and positional maxima/95th percentiles of $\eta, \delta S$. Four document-grouped folds search penalties 10^{-4}–10^{2} with development-fitted scaling. The deployed fit includes native and feedback-perturbed prefixes; penalty 10 and threshold .02818591 target 30% calibration acceptance and allow one merge without categorical readouts. On native prefixes, adding velocity history to diffusion time reduces document-grouped cross-validation RMSE for $\log(e_{\text{merge}} + 10^{-8})$ from .315 to .194; adding prediction-memory motion further reduces it to .181.

E.2 Fixed-grid quality and cost

Fixed K omits native boundaries; Uniform K rebuilds the grid. On 64 ELF-XSum prompts, Fixed 28/24 omit $\{21, 23, 25, 27\}/\{15, 17, \ldots, 29\}$. Timing uses three interleaved A5000 repeats, includes terminal readout, excludes source encoding/scoring, and uses 2,048 paired resamples.

Placement and output sensitivity. At matched 24-update budgets on two reused 64-prompt ELF-XSum panels, Tail changes fewer texts than Mid, Head, or Uniform. A separate 29-boundary scan on 64 prompts relates terminal changes to boundary intercepts and the native final-body fraction not yet committed. Grouped-fold AUC is .836 with boundary alone, .867 with retrospective commitment, and .835 with online history; commitment excludes source/EOS suffixes and does not use revisitation-priority modes.

Unconditional grids. Table 8 pairs 64 draws per model. Head/Tail omit alternating early/late interior points, preserving endpoints and gaps of at most two intervals; Retained-uniform selects rounded native indices. Plaid keeps its ancestral rule and seeded noise stream, without coupling Brownian paths across budgets.

GPT-2-large scores each token after the first once, with context 1,024, stride 512, no added BOS, BF16 weights, FP32 logits, and FP64 accumulation. gPPL weights tokens; paired NLL weights documents; 4,096 resamples recompute both arms. Warm timing uses 16 draws, one warmup and five cyclic repeats on V100s (LangFlow/ELF) or A5000 (Plaid), including model calls and token-ID readout but excluding loading, transfers, string decoding, and scoring.

ELF favors Tail over uniform placement at 75% budget; LangFlow and Plaid do not establish the same ranking. LangFlow's low gPPL accompanies high repetition.

Supplementary Retained-uniform evaluations cover 64 runs each for LangFlow, Plaid, ELF-OWT, ELF-XSum, ELF German-to-English, and Cola-DLM LAMBADA; XSum and translation reuse existing panels. Unlike rebuilt Uniform, this policy permits paired O comparisons on all complete $W = 4$ windows at shared retained indices, with the per-reference 10^{-3} floor. Sampling time ranges from .750 to .772 of native (16 runs, three warm repeats), while generated-token mismatch ranges from 2.63% to 84.97%. Revisitation decreases in Plaid and increases in ELF-OWT, translation, and Cola-DLM; LangFlow and XSum intervals include zero. ELF-OWT NLL increases by $.358$ $[.269, .450]$; the other NLL, ROUGE-L, and BLEU intervals include zero, and Cola-DLM loses two correct answers out of 64 (-3.125 $[-7.813, 0]$ accuracy percentage points). These results concern Retained-uniform and do not replace the rebuilt Uniform outcomes in Table 1.

Independent confirmation and Cola-DLM grids. Table 9 evaluates frozen grids on 64 new ELF noise draws and 64 unused Cola-DLM LAMBADA articles. Cola-DLM grids, developed on 32 separate prompts, omit $1, 3, 5, 7$ (Head) or $9, 11, 13, 15$ (Tail) within each block; Uniform uses 13 points from 1000 to 0. Both blocks preserve native conditioning and paired noise.

ELF retains near-native diversity; Cola-DLM's output agreement and answer accuracy need not move together. All outputs are nonempty; official/strict Cola scoring and all-slot/pre-EOS ELF views agree.

E.3 Online selection and observation costs

Table 10 separates LangFlow/ELF development gates from independent readout-policy evaluations; timing includes online decisions and output transfer. LangFlow's history gate does not recover its overhead, whereas ELF saves time without an established advantage over fixed placement. For

Table 8: Unconditional grid quality and paired placement contrasts on 64 shared draws per model.

Model	Grid	Evaluations	gPPL [95% CI]	Token entropy	Rep-4 (%)
LangFlow	Native	128	7.93 [6.54, 9.61]	2.543	63.92
	Head 75	96	8.96 [7.63, 10.61]	2.693	63.11
	Tail 75	96	7.78 [6.47, 9.39]	2.529	63.88
	Uniform 75	96	8.19 [6.96, 9.59]	2.602	62.10
	Retained-uniform 75	96	8.55 [7.20, 10.08]	2.638	61.74
	Tail 62.5	80	7.57 [6.33, 9.07]	2.508	64.31
Plaid	Native	129	45.99 [41.22, 51.29]	4.271	1.00
	Head 75	97	49.25 [44.76, 54.10]	4.290	0.75
	Tail 75	97	50.23 [44.48, 56.76]	4.278	0.80
	Uniform 75	97	47.53 [41.95, 53.69]	4.304	0.81
	Retained-uniform 75	97	44.03 [38.87, 49.96]	4.299	1.00
	Tail 62.5	81	54.35 [48.87, 60.89]	4.293	0.79
ELF	Native	33	436.79 [365.36, 519.45]	4.142	2.99
	Head 75	25	1003.82 [860.60, 1169.16]	4.312	0.50
	Tail 75	25	444.89 [374.40, 527.14]	4.143	2.88
	Uniform 75	25	613.19 [512.82, 734.58]	4.206	1.44
	Retained-uniform 75	25	630.40 [531.48, 748.63]	4.225	1.40
	Tail 62.5	21	610.34 [517.02, 718.04]	4.193	2.18

Model	First minus second	ΔgPPL [95% CI]
LangFlow	Head 75 – Tail 75	1.19 [−0.03, 2.35]
	Tail 75 – Uniform 75	−0.41 [−1.46, 0.69]
	Tail 75 – Retained-uniform 75	−0.77 [−2.14, 0.55]
	Tail 75 – Native	−0.15 [−0.82, 0.25]
Plaid	Head 75 – Tail 75	−0.97 [−8.83, 6.59]
	Tail 75 – Uniform 75	2.70 [−4.07, 9.28]
	Tail 75 – Retained-uniform 75	6.19 [−0.35, 12.76]
	Tail 75 – Native	4.24 [0.30, 8.52]
ELF	Head 75 – Tail 75	558.93 [462.09, 677.02]
	Tail 75 – Uniform 75	−168.30 [−233.33, −115.93]
	Tail 75 – Retained-uniform 75	−185.50 [−242.89, −132.67]
	Tail 75 – Native	8.10 [−4.21, 19.75]

Suffixes give nominal retained-grid percentages; evaluations include terminal readouts. Uniform rebuilds the lower-budget grid, whereas Retained-uniform selects rounded native indices. Entropy and repetition are averaged within then across 128-slot outputs. gPPL uses token-weighted GPT-2-large NLL; 95% intervals use 4,096 paired document resamples, recomputing both arms rather than subtracting marginal intervals.

LangFlow/Plaid, stability over four top-1 readouts triggers stride-two sampling while preserving terminal evaluation; thresholds 1.0 and $79/128$ are selected by development call counts and frozen before confirmation. For ELF-XSum, the fraction of slots below confidence .9 triggers Tail at $\leq 1/24$, or a single merge at zero. Shuffled switching permutes switch times across samples as a diagnostic control.

E.4 History signals and velocity damping

On 64 ELF-XSum development prompts, within-boundary motion signals predict later native revisitation (AUC .825–.832) better than path cancellation (.226), but predict categorical admission weakly (.499–.563).

Table 11 reports the shared ELF-XSum damping panel. Native revisitation has already ended at 34/37 readout-selected and 28/29 history-selected positions when pulses act. The readout-selected first-window reduction does not persist; unequal realized doses and random-position responses limit attribution to the selector. Dose-matched release-order comparisons likewise show no reliable quality advantage.

Table 9: Quality and sampling time under frozen fixed grids on independent ELF and Cola-DLM confirmation panels.

Model	Grid	Quality	Time (s)	Time / native [95% CI]	Paired quality change [95% CI]
ELF	Native	335.49	.749	1.0000	reference
	Tail	338.26	.569	.7595 [.7575, .7615]	+.00754 [−.01858, +.03403]
Cola-DLM	Native 16	34/64	9.347	1.0000	reference
	Head 12	36/64	7.133	.7631 [.7605, .7647]	+3.125 [0, +7.813]
	Tail 12	32/64	7.128	.7626 [.7596, .7647]	−3.125 [−7.813, 0]
	Uniform 12	34/64	7.147	.7647 [.7645, .7648]	0 [0, 0]

Quality is token-weighted GPT-2-large gPPL for ELF and correct answers out of 64 for Cola-DLM. Paired quality change is document-mean NLL for ELF (lower is better) and accuracy in percentage points for Cola-DLM (higher is better). Timing uses 16 samples/model with three warm repeats; repeats are averaged within samples before 4,096 paired-sample bootstrap resamples. ELF times request initialization through CPU token IDs; Cola-DLM times the native generation call, including prefix VAE/cache operations, guidance, and final VAE decoding. Their timing scopes differ, so seconds are compared only within each model. All 64 outputs per arm are nonempty and scorable; the policies and samples were fixed before confirmation access.

Table 10: Online merging policies and fixed controls, with paired quality and sampling-time intervals.

Policy	Mean evaluations	Quality change [95% CI]	Time / native [95% CI]
LangFlow: 32 development runs; quality = ΔNLL			
Readout gate (single)	127.656	−0.0020 [−0.0055, +0.0000]	1.0080 [1.0015, 1.0153]
History gate	127.625	+0.0000 [+0.0000, +0.0000]	1.0073 [1.0037, 1.0107]
Fixed single	127.000	−0.0123 [−0.0492, +0.0136]	0.9922 [0.9895, 0.9956]
ELF: 32 development runs; quality = ΔNLL			
Readout gate (single)	36.156	+0.0017 [−0.0048, +0.0081]	1.0957 [1.0813, 1.1105]
History gate	32.000	+0.0054 [−0.0002, +0.0118]	0.9750 [0.9713, 0.9788]
Fixed single	32.000	−0.0012 [−0.0090, +0.0059]	0.9738 [0.9704, 0.9771]
LangFlow: 64 independent runs; quality = ΔNLL			
Fixed Tail	96.00	+0.0039 [−0.0223, +0.0314]	0.7520 [0.7487, 0.7573]
Readout gate (Tail)	90.06	+0.0179 [−0.0122, +0.0594]	0.7924 [0.7547, 0.8373]
Shuffled switch times	90.06	−0.0022 [−0.0490, +0.0452]	0.8004 [0.7629, 0.8424]
Plaid: 64 independent runs; quality = ΔNLL			
Fixed Tail	97.00	+0.0732 [−0.0253, +0.1675]	0.7520 [0.7516, 0.7524]
Readout gate (Tail)	95.73	+0.1556 [+0.0451, +0.2619]	0.7494 [0.7237, 0.7772]
Shuffled switch times	95.73	+0.0858 [−0.0112, +0.1914]	0.7524 [0.7268, 0.7784]
ELF-XSum: 64 independent prompts; quality = ΔROUGE-L			
Fixed Tail	49.00	+0.139 [−0.017, +0.352]	0.755 [0.754, 0.757]
Fixed Head	49.00	−1.962 [−3.312, −0.684]	0.756 [0.754, 0.757]
Fixed single	63.00	+0.000 [+0.000, +0.000]	0.970 [0.969, 0.972]
History gate	63.56	+0.000 [+0.000, +0.000]	0.977 [0.971, 0.983]
Readout gate (single)	67.12	+0.000 [+0.000, +0.000]	1.013 [1.004, 1.022]
Readout gate (Tail)	49.47	+0.054 [−0.039, +0.213]	0.702 [0.673, 0.736]
Shuffled switch times	49.47	+0.272 [+0.054, +0.542]	0.773 [0.714, 0.838]

Evaluations include decision readouts and terminal decoding; lower NLL and higher ROUGE-L are favorable. Sampling-time ratios use the first 16 runs, three balanced warm repeats, and within-run medians; quality and time intervals use 4,096 paired resamples. LangFlow/ELF single-merge gates are fitted separately on development data; fixed controls use the same panel as their gate. Shuffling preserves the full-panel switch-time distribution, not each run's budget or the timing subset's budget. On ELF-XSum, readout Tail and shuffled policies each average 49.47 evaluations on all 64 prompts but 45.44 and 50.00 on the timing subset; the history and readout single gates average 63.38 and 65.62 on that subset. Plaid retains its seeded ancestral noise stream; changing the grid does not preserve an identical stochastic path.

Velocity damping across settings. Table 1 separates changes in dynamics from quality: readout-selected pulses do not establish lower revisitation than matched random positions outside ELF-XSum, and schedule effects on revisitation do not consistently improve quality.

Table 11: Velocity damping and combined policies on 64 shared ELF-XSum prompts, with paired 95% intervals.

Selector	Trig.	First window	Remaining	Dose	ΔRL	G/T/L
Readout (χ)	37	-3.97 $[-10.60, 0.00]$	-1.20 $[-3.54, +0.02]$	.033	$+.03$ $[.00, +.08]$	1/63/0
Random	37	$+0.60$ $[0.00, +1.80]$	-0.31 $[-1.14, +0.37]$	.022	$+.03$ $[.00, +.08]$	1/63/0
History (R)	29	-3.20 $[-9.60, 0.00]$	-0.18 $[-0.53, 0.00]$	.008	.00 [.00, .00]	0/64/0
Random	29	0.00 [0.00, 0.00]	$+0.95$ $[-0.01, +2.85]$	.011	.00 [.00, .00]	0/64/0
Paired contrasts at the selected positions (10^{-3})						
$\chi -$ random		-4.57 $[-11.20, 0.00]$	-0.89 $[-3.58, +0.88]$			
$R -$ random		-3.20 $[-9.60, 0.00]$	-1.13 $[-3.32, 0.00]$			

Fixed schedule	Mean ΔO (10^{-3})	Dose	ΔRL	G/T/L	Changed
Uniform	0.00 $[-0.25, +0.24]$	2.35	$-.04$ $[-.37, +.26]$	4/56/4	24
Staggered, left to right	-0.05 $[-0.31, +0.19]$	2.18	.00 $[-.34, +.29]$	6/53/5	27
Reversed order	$+0.09$ $[-0.18, +0.36]$	5.95	$-.21$ $[-.64, +.17]$	5/51/8	36
Random order	$+0.11$ $[-0.17, +0.37]$	4.48	$-.19$ $[-.66, +.27]$	4/53/7	31
Reversed, dose-matched	—	1.96	$-.12$ $[-.42, +.12]$	3/57/4	25
Random order, dose-matched	—	2.02	$+.12$ $[-.20, +.41]$	5/56/3	24

Merging or combined policy	ΔRL	G/T/L	Changed
Tail (no damping)	$+.14$ $[-.02, +.35]$	5/57/2	27
Staggered + Tail	$+.06$ $[-.24, +.33]$	6/53/5	39
Gate (no damping; 46 merges)	.00 [.00, .00]	0/64/0	3
Uniform + gate (44)	$-.11$ $[-.42, +.13]$	3/57/4	26
Staggered + gate (50)	.00 $[-.34, +.29]$	6/53/5	28
History + gate (49)	.00 [.00, .00]	0/64/0	4
Random + gate (47)	.00 [.00, .00]	0/64/0	4

Pulse effects (10^{-3}) compare selected positions with native in the first fully released $W = 4$ window and remaining observations; nontriggers contribute zero. Schedule effects average all generated positions. Dose is $\sum_k \|\Delta t_k(\widetilde{v}_k - v_k)\|_2$; ΔRL compares ROUGE-L with native, G/T/L counts gains/ties/losses, and Changed counts different summaries.

Arms share prompts, noise, and native outputs; intervals use 4,096 paired document resamples. Values and interval bounds are rounded to two decimal places. Dashes mark unmeasured dose-matched dynamics; Table 12 reports combination dynamics.

Dose-matched strengths scale by .4375 (reversed) or .5625 (random), fixed on 32 development prompts. Their ROUGE-L contrasts against left-to-right release are $-.12$ $[-.29, .00]$ and $+.12$ $[-.04, +.35]$; the left-to-right replay reproduces all 64 outputs.

Merge-only and combined-policy ΔO use fixed primary prefixes (16 LangFlow runs; 32 otherwise), except merge-only Cola-DLM (64 prompts) and a separate LangFlow single-merge panel. Single-merge quality/cost uses 32 LangFlow/ELF runs and 64 Plaid/Cola-DLM runs; Cola damping/combinations use 32 prompts, and LangFlow/Plaid readout switching uses separate 64-run panels. All available combined-policy extensions are reported, irrespective of effect direction. Dashes indicate unevaluated policies or unavailable paired outcomes; rebuilt Uniform has no paired ΔO because its grid does not retain native readout times.

All ΔO outcomes use complete $W = 4$ windows and the per-reference 10^{-3} floor before maximization, with frozen intervention selectors and no coverage expansion. They measure continuous-score changes, not detection-indicator changes. Merge comparisons restrict native readouts to retained input calls, excluding the extra terminal Plaid decoder. Cola merge-only scores average generated positions in the first block; Staggered + Tail uses later windows (21.33 $[11.00, 32.08] \times 10^{-3}$), not the full-run secondary outcome. ELF-OWT schedule outcomes use calls 20–32, specified during supplementary analysis to match the final-40% window used for LangFlow/Plaid. Their uniform/staggered effects are $+.394$ $[-.354, 1.170]$/$+.0004$ $[-.779, .732]$ in 10^{-3} units; all-window estimates are $+.629$/$+.345$.

Timing and quality retain their original panels after score recomputation. LangFlow schedule/history timing uses separate paired panels (zero-strength ratios .996/1.06); Plaid pulse timing includes online selection. ELF-XSum/translation timing uses the first 16 prompts except XSum Tail and Uniform; Uniform uses three-repeat median timing on all 64 confirmation prompts. Token mismatch preserves panel-specific masks: Cola grids include all 32 sampled slots, including prompt overlap, whereas ΔO excludes prompt positions. Translation-only Tail/Head mismatch is .71%/16.88%.

Combined-policy dynamics at matched calls. Table 12 separates fixed-prefix primary estimates from nested 64-run extensions. Staggered + Tail uses all-position later windows (fully released in Plaid); History + gate uses selected positions in the first released window, with nontriggers contributing zero. All saved combined outputs reproduce; XSum uses V100 replay, excluding the device-sensitive A5000 replay.

Table 12: Combined-policy changes in revisitation at matched retained calls ($10^3\Delta O$), separating the fixed-prefix primary analysis from the full-panel extension.

Setting	Policy	Primary	Extension ($n = 64$)	Triggers primary/64
LangFlow	Staggered + Tail	$+0.485\ [-4.576, +6.309]$	$+0.011\ [-1.959, +2.049]$	—
LangFlow	History + gate	$+2.040\ [-3.400, +10.711]$	$-0.888\ [-4.289, +2.171]$	16/64
Plaid	Staggered + Tail	$-6.293\ [-8.499, -3.929]$	$-5.896\ [-7.472, -4.259]$	—
ELF-OWT	Staggered + Tail	$+0.307\ [-0.488, +1.086]$	$+0.571\ [-0.143, +1.313]$	—
ELF-OWT	History + gate	$0\ [0, 0]$	$0\ [0, 0]$	3/9
ELF-XSum	Staggered + Tail	$-0.127\ [-0.555, +0.272]$	$-0.218\ [-0.473, +0.029]$	—
ELF-XSum	History + gate	$0\ [0, 0]$	$-3.201\ [-9.604, +0.000]$	20/30
ELF de→en	Staggered + Tail	$+0.171\ [-0.141, +0.474]$	$+0.199\ [-0.010, +0.405]$	—
ELF de→en	History + gate	$0\ [0, 0]$	$0\ [0, 0]$	9/22

Primary panels contain 16 runs for LangFlow and 32 for other settings. Brackets give 95% paired document-bootstrap intervals (4,096 resamples). Triggers count pulses under each combined policy, not standalone damping; fixed schedules need no trigger. Nontriggering runs contribute zero to history-policy means. Each extension contains its primary prefix, so the two columns are not independent replications.

On the primary Plaid panel, combined-minus-Tail on the same grid is $-2.134\ [-4.262, -.017] \times 10^{-3}$.