READING LATENT STATES: ADAPTIVE HALTING IN CONTINUOUS DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Continuous diffusion language models think in vectors, not words, refining a latent state step by step and decoding text only at the end. Yet each trajectory is run to its final call, and two things are unknown. (1) Whether a latent state can be read at all: it is not text, so there is no candidate to score and no confidence to threshold. (2) Whether the full budget is even necessary. To address both, we split the job between a teacher and a student. Offline, a frozen teacher readout marks where the answer enters and whether the state is good enough to stop. At decision time a student reader sees only the latent prefix and a budget signal and emits HALT or CONTINUE—never text or future states. We apply this to CoLA and ELF latent trajectories. The property under test, trajectory parseability, must hold on three counts: predictability from a prefix, information beyond what the computation clock alone gives, and a usable decision. On WMT14 the policy uses 8.8 of 64 solver calls—86% less computation—and returns BLEU 26.2572 against 26.1466 for full inference. On XSum, 9.5 of 64 calls (85% less) at ROUGE-L 27.9232 against 27.811. Unconditional generation is harder: 24.6 of 32 calls, 23% less, at a small perplexity cost. Trajectories make up their minds early: in CoLA the output identity is in place after the first of four latent blocks for 63.6% of saved trajectories, and WMT14 quality peaks at call 19 of 64. Stopping early is thus less a compromise than a correction, because quality is not monotone in computation. Results are replayed from saved trajectories, not a measured speedup.

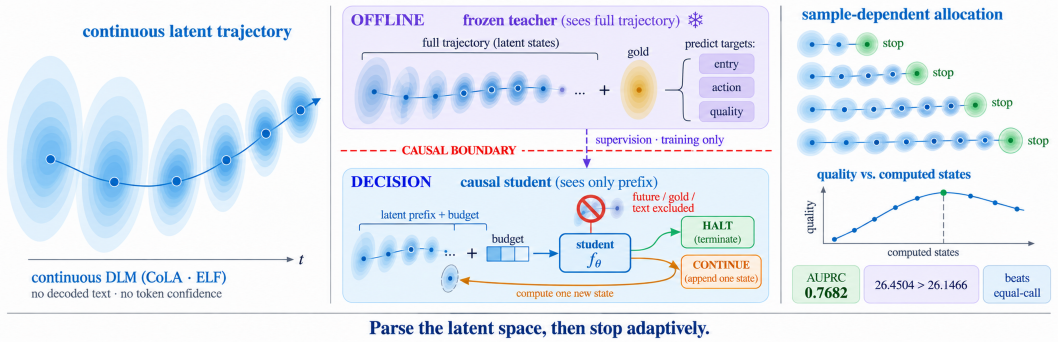

Figure 1: Reading latent trajectories for adaptive computation. A continuous DLM refines a latent trajectory and decodes text only at the end. Offline, a frozen teacher readout sees the whole trajectory and defines the targets: where the answer enters, whether to act, and the quality at stake. A causal boundary then separates what the decision-time student may use—the latent prefix and a budget signal—from future states, the reference, and the text. The student emits HALT or CONTINUE, and different samples stop at different points, which turns a fixed budget into a sample-dependent allocation.

1 INTRODUCTION

Continuous diffusion language models think in vectors rather than words. CoLA builds a sequence of latent blocks before it decodes anything, and ELF evolves an embedding state before mapping it to tokens (Guo et al., 2026; Hu et al., 2026). Because those states are not text, an answer cannot

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.

be read off them the way a token distribution can be read: there is no candidate to score and no confidence to threshold. Meanwhile every step of the trajectory costs another solver call, and the usual way to run these models is to pay for all of them.

The computation, however, is front-loaded. In CoLA the identity the model finally outputs is already in place after the first of four latent blocks for 63.6% of 14,729 saved trajectories. On WMT14 mean quality climbs steeply, peaks at call 19 of 64, and then drifts down. A trajectory tends to make up its mind early and keep polishing afterwards—sometimes past the point where polishing helps. That is the opening for early stopping: if the latent state already says "good enough", the remaining calls can be skipped.

Prior work reads a different surface. One line decodes intermediate states and asks whether the answer has stabilised (Du & Ma, 2026); another turns token-confidence or logit-margin signals into early-exit rules (Li et al., 2026b). Both need text or tokens to look at, and a continuous latent state offers neither. So the question to settle first is simply whether the latent space can be read at all, and whether what it says is worth acting on.

We call the property *trajectory parseability* and define it so that it can fail. A trajectory is parseable if its future can be predicted from a prefix, if that prediction is not merely a reflection of how far the computation has run, and if the prediction can drive a usable action. Our protocol splits the work between a teacher and a student. Offline, a frozen *teacher* readout inspects complete saved trajectories and marks each step with the quantities we care about: where the output identity enters, whether the current state is good enough to stop, how much quality another call would buy. At decision time a causal *student* reader sees only the latent prefix and a budget signal, predicts the teacher's judgement, and emits HALT or CONTINUE. The student is never shown decoded text, references, or future states. We apply the protocol to two continuous families—CoLA block-latents and ELF embedding flows—and, for the interface analysis, to the discrete masked-token model Dream. Figure 2 overviews the setup.

The savings are large wherever the trajectory settles early. On WMT14 the frozen policy stops after 8.8 of 64 solver calls on average, using 13.7% of the computation, and still returns BLEU 26.2572 against 26.1466 for full inference. On XSum it stops after 9.5 of 64 calls, 14.9% of the computation, at ROUGE-L 27.9232 against 27.811 for full inference. Unconditional generation is the difficult case: 24.6 of 32 calls, 23% less, at a small perplexity cost. Early stopping here is less a compromise than a correction, because quality is not monotone in computation; and the readable surface is architecture-dependent, since at Dream's discrete masked-token interface the predictive signal is mostly output-facing, where we make no halting claim.

Our contributions are threefold:

- **Reading the latent space.** We show that continuous-DLM latent trajectories are functionally parseable under a three-part definition, and that the read signal is not merely a restatement of how far the computation has run.
- **Early stopping that pays.** We freeze that signal into policies that cut WMT14 and XSum inference to roughly one seventh of its calls without losing quality, and characterise the harder unconditional-generation case.
- **Two continuous interfaces, one protocol.** We instantiate the same protocol on CoLA and ELF, and show how the observable interface differs on a discrete masked-token model.

Throughout, *reasoning trajectory* refers to iterative internal computation; the tasks do not all assess explicit logical reasoning. Section 2 defines the native interfaces we study, Section 3 presents the protocol, and Section 4 reports the systematic comparisons and analyses.

2 Related Work and Native Interfaces

Native interfaces and computation units. Diffusion language models iterate on two different things: continuous ones refine a latent vector, discrete ones a token canvas. Each exposes a different computation interface that we treat as first-class. Let $s_{1:T}$ be a recorded state trajectory, $u_{1:T}$ its process fields such as block index or solver time, and D_A a frozen readout from intermediate states to output candidates for architecture A. Diffusion-LM and DiffuSeq established continuous denoising for controllable and sequence-to-sequence generation (Li et al., 2022; Gong et al., 2023). CoLA

welds a Text VAE to a block-causal diffusion transformer and conditional decoding (Guo et al., 2026): the state is a growing prefix of continuous latent blocks, the outer block the unit you pay for. ELF runs embedding-space flow matching with shared weights for discretization (Hu et al., 2026): a state is the embedding a completed call leaves behind, the solver call the unit. Dream repaints a masked-token canvas (Ye et al., 2025), laying token, mask, and logit statistics out at each denoising step. Together they let an offline observer watch an answer take shape while the generator keeps computing—and "the computation step" differs in each.

Reading intermediate computations. Reading inside a computation is not new; what it tells you is. The nearest continuous-DLM precedent asks how much meaning survives in ELF states, how sensitive they are to order, and whether a decoder can consume them, with decoder-basin diagnostics, margin-based early exit, and CoLA boundary checks (Du & Ma, 2026). Others read different surfaces: autoregressive hidden states for correctness (Zhang et al., 2025), attention heads for context sufficiency (Xie et al., 2025), functional correctness for offline filtering (Chiu & Liu, 2026), and latent reasoning probed structurally, geometrically, causally (Chang et al., 2026). Readability is thus a research direction; our test adds the pairing: an offline measurement interface, prefix-only decisions, matched inputs, an allocation assay. But internal states also carry how far denoising has run (Rulli et al., 2026), so a reader may only be reading the clock, and accuracy alone is not evidence of sample-specific output information.

Computation–quality dynamics. More computation is not automatically better output. Prefix evaluations separate a trajectory still polishing a settled answer from one that loses an answer it already had (Caldarella et al., 2026), so computation and quality must be measured against each other, not assumed. And because denoising progress is readable from internal states (Rulli et al., 2026), a score that climbs with the step counter does not by itself show the output is improving. We report computation–quality relations separately from readability, and make no claim that our readers detect or prevent harmful continuation.

Adaptive computation and early commitment. Stopping early is easy when you can watch the model's confidence: when the top token is far ahead of the runner-up, the answer is chosen, and Prophet turns that gap into a halt rule (Li et al., 2026b). ACT and PonderNet learn variable computation in recurrent models (Graves, 2016; Banino et al., 2021); earlier DLMs halted adaptively (Vaina et al., 2024). A later family reads that surface more carefully: SchED moves its thresholds with progress, LESS adds temporal stability, Ada-DLM follows confidence to token convergence (Mohamed et al., 2026b;a; Li et al., 2026a); visible reasoning prefixes also support sufficiency assessment and early exit (Xiang et al., 2026). Our readers get none of it: they see a continuous latent prefix and process fields, while the decoded candidates and token-confidence statistics these rules consume are out of reach at decision time. We use termination to ask what that information is worth; the present evidence does not establish a faster or quality-preserving alternative to these systems.

3 Method: A Measurement-and-Action Protocol

We turn intermediate readouts into supervision for a reader that acts without them. Figure 2 draws the split in CoLA: offline, a frozen teacher readout watches a trajectory to the end and leaves notes; at decision time a student reader gets only the prefix. Offline labels on one side, prefix-only decisions on the other, with the generator frozen and only the reader fitted.

3.1 Offline measurement of output and action events

An intermediate state carries no label of its own: no latent block or embedding flow announces that the answer has arrived, so we define targets through a frozen, architecture-specific readout D_A. For sample i, a target has the general form

$$y_{it} = \mathcal{T}_A(s_{i,1:T}, D_A(s_{i,1:T}), r_i)\,, \tag{1}$$

where r_i denotes references or task scores when available. The construction is offline, and the asymmetry is the point: the teacher may look ahead, the reader may not. Targets record what each experiment needs.

Three settings, three kinds of note: the first entry of a decoder-defined reference prediction for CoLA, a task-specific quality/cost action for ELF, and current correctness plus final-answer agreement for Dream (Table 2, Appendix D).

CoLA's primary target is the one-time reference-entry event of Section 4.2: the block where an answer first shows up and stays. Auxiliary heads predict correctness, reference disagreement, nonempty output, decoder stop, completion risk, post-entry position, and future correctness gain. Those labels are written offline and are never input features; Appendix A.1 gives their weights and definitions.

ELF asks a blunter question: is one more call worth buying? With offline candidate quality q_{it} and best strictly future quality $q_{it}^{+} = \max_{t<u\le H} q_{iu}$ within policy horizon H, the teacher labels HALT when $q_{it} \ge q_{it}^{+}$ and CONTINUE otherwise, scoring with sentence chrF++ for WMT14 and ROUGE-L for XSum. OpenWebText has no reference to score against, so a dynamic program trades terminal-output disagreement against call cost instead (Appendix A.3). Read these as distinct decision criteria, not as one shared scale of reasoning sufficiency.

3.2 Learning a prefix-only reader

The reader is the student, and its view is narrow: it predicts $p_\theta(y_{it} \mid z_{i,\le t}, u_{i,\le t})$, where z denotes permitted state features and u process fields. *Causal* here means only that access stops at the current prefix; it does not imply causal identification of the generator. Section 3.3 lists the inputs excluded at decision time. One final output mapping after halting is allowed.

Each interface gets its own reader. CoLA pools raw latent slots and processes the chronological block prefix with a causal Transformer; ELF uses fixed flow-state features followed by a unidirectional GRU. The WMT14 and XSum readers project feature groups to width 32 before a 128-dimensional recurrent state, while OpenWebText uses a separate two-stream summary of the clean state and stochastic update. Each architecture or native task has its own fitted parameters; Appendix B gives full encoders and fitting settings.

On WMT14 and XSum, training charges more for the mistake of stopping too early. With regret $r_{it} = (q_{it}^{+} - q_{it})_{+}$ and mean positive training regret μ, the objective is

$$w_{it} = 1 + r_{it}/\mu, \qquad \mathcal{L}_{\text{action}} = \frac{\sum_{i,t} w_{it}[-\log p_\theta(y_{it} \mid z_{i,\le t}, u_{i,\le t})]}{\sum_{i,t} w_{it}}. \tag{2}$$

The sum covers valid sample-position pairs within a training batch; OpenWebText uses the same weighting form with its dynamic-programming excess stop cost as regret. All positions before H remain supervised, including positions after an earlier teacher halt. It is positionwise: it approximates an absorbing first-stop decision, and better action prediction need not imply better stopping utility.

3.3 Matched identification and the information boundary

Two questions live here, and must stay apart. The first: does a reader's accuracy carry anything specific to this sample, or only restate how far the computation has run? We therefore fit the same student twice: an aligned reader (RB) that reads the current sample's prefix, and a second reader (SB) that reads a coherent prefix borrowed from another sample matched on declared conditions—task, split, and block position for CoLA, split and schedule groups for ELF—with a zero-latent reader (ZB) as an out-of-distribution control, both arms keeping the same process-budget interface. RB–SB identifies sample correspondence under these matches; it does not remove every sample-level nuisance, and no strictly clock-free reader can be claimed while mask, prefix length, and position fields remain available.

The second question is what a decision may read; Table 4 (Appendix D) draws that line stage by stage. Offline label construction may inspect decoded text, references, task scores, and future or final states, because it writes supervision rather than a decision. The reader and the stopping rule may use only the current and historical state prefix, the process fields, and one output mapping after halting. Decoded candidates, token logits, references, task scores, and future or final states are excluded at decision time in continuous settings; Dream is analysed separately because its native interface exposes token, mask, and logit statistics directly.

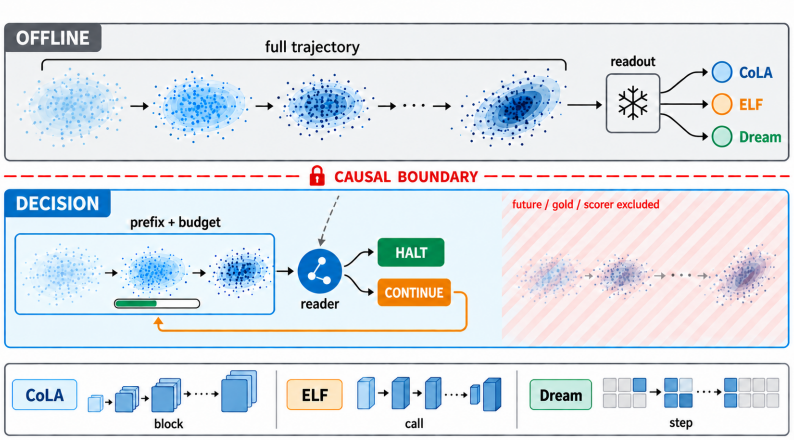

Figure 2: The measurement-and-action protocol. Offline, a frozen teacher readout may walk a complete trajectory and emit architecture-specific targets. A causal boundary separates what a decision-time student may read—the state prefix plus a budget signal—from the future states, the reference output, and the scorer, which stay out of reach. The student calls HALT or CONTINUE, and a frozen policy turns that into an allocation. The lower strip names each architecture's native unit: an outer latent block for CoLA, a solver call for ELF, a denoising step for Dream. Adaptive computation is evaluated by saved-trajectory replay, not online deployment.

3.4 From a readout to a stopping decision

The student speaks once; a frozen policy listens. It stops at its first halt decision, otherwise using the policy horizon:

$$\tau_i = \min\big(\{t < H : h_\theta(z_{i,\le t}, u_{i,\le t}) = \text{HALT}\} \cup \{H\}\big). \tag{3}$$

Each ELF task averages the two action logits from its three aligned best checkpoints, halting when the mean halt logit is at least the mean continue logit. That is the whole rule: no threshold calibration, minimum-call floor, or persistence test. A completed call has already incurred its cost; stopping can remove only later calls. CoLA instead uses a post-hoc mismatch-calibrated policy, evaluated as exploratory evidence.

We evaluate decisions by selecting outputs along fully generated trajectories: a nonadaptive rule with the same average calls tests the value of sample-dependent allocation, and a fixed-horizon anchor measures quality sacrificed relative to a larger budget. Replay makes the outputs directly comparable but does not measure cancelled computation or runtime savings.

4 Experiments

Three questions organise the experiments: whether a trajectory is readable beyond matched computation position, whether more computation improves quality, and whether the readable signal can be frozen into halting policies across native interfaces.

4.1 Experimental setup

Models and tasks. CoLA anchors the identification work: three composite seeds, eight held-out tasks (LAMBADA, MMLU, OpenBookQA, HellaSwag, RACE, Social IQa, SQuAD, Story Cloze), and three reader input conditions—72 fitted readers and 24 aligned–shuffled comparisons. Reader fitting leaves out the evaluation task; the stopping protocol also uses target-task validation. A composite seed varies generation, fitting, splitting, and shuffling and is not an independent dataset replication. Appendix C.1 gives dataset citations, populations, and partition rules.

ELF adds three generation tasks behind frozen official generators (Hu et al., 2026; Bojar et al., 2014; Narayan et al., 2018; Gokaslan et al., 2020): WMT14 German–English translation, XSum summarization, and OpenWebText generation, with 4,096 monitor-training trajectories, 512 development trajectories, and three reader initialization pairs each. Frozen policy evaluation uses 3,003 WMT14 official-test examples, 11,329 XSum test examples after removing five overlapping rows, and OpenWebText seeds 6–11 with 1,000 examples each. WMT14 and XSum have 16-call policy horizons

inside a native 64-call trajectory; OpenWebText has 32. Dream's smaller predictive study appears in Section 4.5.

Matched controls. Section 3.3 defines the aligned (RB), shuffled (SB), and zero-latent (ZB) readers and their declared matches; ELF also matches condition length in bins and keeps the recipient's process fields. RB–SB tests sample correspondence beyond these matches and does not eliminate every sample-level nuisance.

Metrics and references. CoLA reports reference-entry AUPRC and AUROC on held-out tasks; ELF compares development action loss and then task quality under the frozen stopping rule, using validation-selected best checkpoints. Two nonadaptive references serve the ablations below: an equal-call rule with fixed, content-independent prefix lengths matching the policy's total calls, and a longer-horizon anchor, fixed16 or fixed32. Both separate allocation gains from quality preservation. Uncertainty, paired bootstrap procedures, and policy locks are in Appendix C.1–C.3.

4.2 Continuous latent trajectories are parseable

An output landmark in CoLA. Start with the plainest question: at which block does the identity a trajectory finally outputs first appear? The recovered CoLA setting has four outer blocks, and each saved prefix contributes its scored prediction after lowercasing and whitespace normalization. The *reference prediction* is the first nonempty identity repeated in two adjacent blocks, falling back to the final identity when none exists; the *reference-entry block* is its first occurrence. Empty references take block four by convention and are reported separately. Appendix A.1 gives the exact definition.

The answer usually arrives early. Across 14,729 saved test trajectories—eight tasks, three composite seeds, no selection for correctness—the reference first appears in block one for 9,367 trajectories (63.60%) and by block two for 12,609 (85.61%), and in LAMBADA 98.19% enter at block one (Figure 8, Appendix D). A further 1,236 have empty references and take the final-block fallback, which is not evidence that an answer forms late.

A landmark without a fitted reader is useful, but not yet a policy. Because the reference is chosen retrospectively, early entry alone gives neither an online stopping rule nor a guarantee of correctness or later stability. Entries also cluster at particular blocks, so the comparison below holds computation position fixed.

The latent space is readable, and not only the clock. CoLA rules out the obvious rival, the clock: a student predicts that landmark from the current sample's prefix alone. Macro AUPRC is 0.7682, against 0.6251 for matched-shuffled prefixes and 0.6165 for a zero-latent control, and macro AUROC is 0.8837, 0.8492, and 0.8501. The matched-shuffled arm is the closure that the second condition of parseability requires rather than the claim itself: it fixes task, split, and block position, so a 0.1431 AUPRC gap cannot be produced by tracking the computation clock. The effect is task dependent, with seven of eight task means favouring alignment while LAMBADA reverses, and task-cluster intervals cross zero, so the aggregate gain does not establish a uniformly positive effect across task populations. Readability does not certify correctness or identify the semantic variable behind it; Figure 5 (Appendix D) gives the per-task values and the seed ranges.

Action information in ELF. The same question lives again on ELF's embedding flows, where the target is an action, not an identity. Aligned prefixes yield lower development action loss in every initialization pair: mean RB/SB losses are 0.3479/0.3920 on WMT14, 0.42957/0.45881 on XSum, and 0.05753/0.06740 on OpenWebText (Figure 6). Loss definitions and weights differ across tasks, so magnitudes are not comparable. The pairs share trajectories and use checkpoint-selection development data, giving correspondence evidence under those protocols rather than independent held-out confirmation.

Reader perturbations. We perturb inputs to the 24 fitted RB readers, holding their weights fixed. Shuffling complete prefixes across block-matched samples lowers AUPRC by 0.3220 on average, reversing prefix order by 0.2695, zeroing the current block by 0.1822, and removing history by 0.1104: the readers depend on sample correspondence, current content, and chronology, and the

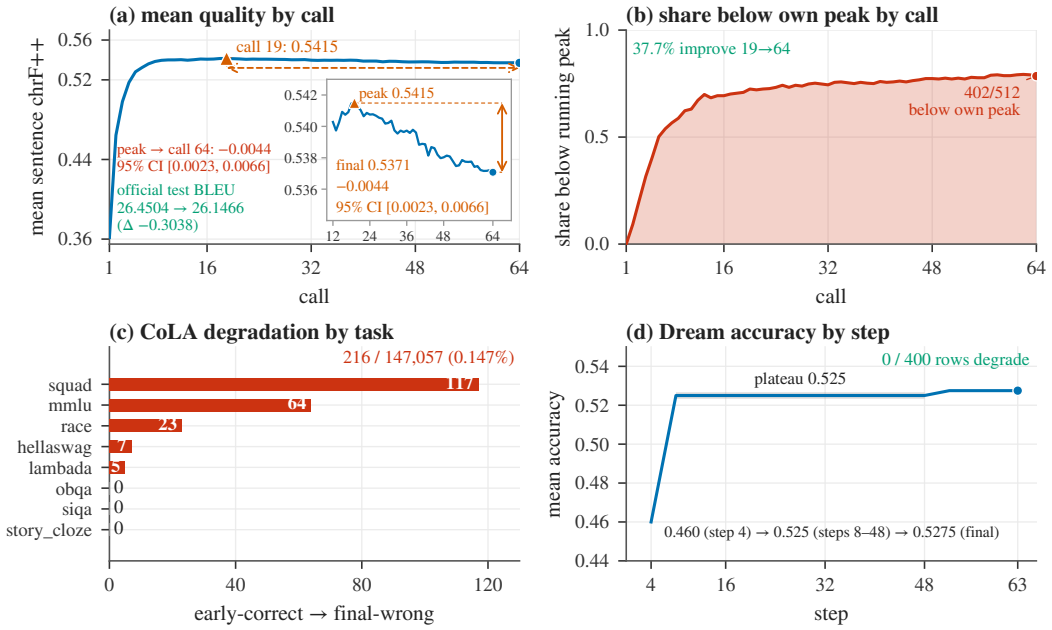

Figure 3: Computation and quality are not monotonically coupled. (a) Mean sentence chrF++ per solver call on the 512-example WMT14 monitor-validation pool: peak 0.5415 at call 19, declining to 0.5371 at call 64; the inset magnifies calls 12–64, where the decline occupies about half the panel height and not the two percent it occupies at full scale. The paired bootstrap interval for the peak-to-final drop is $[0.0023, 0.0066]$, and the official test BLEU endpoint falls from 26.4504 to 26.1466. (b) Share of samples ending below their own running peak, reaching 402/512, while 37.7% improve from call 19 to 64. (c) Earlier-correct/final-wrong CoLA trajectories by task, 216 of 147,057, concentrated in SQuAD and MMLU. (d) Mean step accuracy on Dream's single-seed frontier is a plateau with no trajectory-level reversal. Panels use different populations, metrics, and compute units; values and source digests are in Appendix E.

perturbations act on the reader, not the generator. Effects are task dependent: excluding LAMBADA, prefix shuffling lowers AUROC and AUPRC in all 21 remaining seed–task cells, whereas in LAMBADA zeroing the current block instead improves AUPRC by about 0.2087. Mean-filling explicit process fields barely matters (mean AUPRC drop 0.0020), but mask, length, and position still reveal progress. A geometric assay finds residualized aligned–shuffled structure in the six tasks with adequate within-block support: median oriented centroid cosine 0.8129 across seeds within a task versus 0.1911 across tasks within a seed; LAMBADA and OpenBookQA lack that support. These are task-local, decoder-relative associations, not a universal semantic direction; Figure 4 and Table 11 give per-task drops and cell-level values in Appendix E.

4.3 Computation and Quality are not Monotonically Coupled

More calls do not buy a better answer. This second finding needs no halting policy: quality need not increase with trajectory length, and Table 5 (Appendix D) collects the endpoints available from the saved trajectories.

On WMT14 the last call is the worse one: fixed16 obtains BLEU 26.4504 on the official-test trajectories and the native 64-call endpoint 26.1466, so the shorter readout is better by 0.3038. This aggregate comparison over one set of trajectories does not imply that every sample follows the same pattern, nor does it establish that a learned policy selects each sample's best state, whose quality remains below fixed16.

The same pattern draws a curve on the 512-example WMT14 monitor-validation pool, where mean sentence chrF++ peaks at call 19 and declines to the 64-call endpoint. The decline is a net effect: 37.7% of samples improve between those two calls, and 402 of 512 end below a value they had already reached earlier in their own trajectory (Figure 3).

CoLA shows the same at the level of single answers, though rarely: 216 earlier-correct/final-wrong cases among 147,057 sample-seed trajectories (about 0.147%), concentrated in SQuAD and MMLU,

which contribute 181 of the 216; OBQA, SIQA, and Story Cloze contribute none. Degradation happens, but cases this rare cannot establish systematic detection of harmful continuation, and this denominator belongs to the three-seed dynamics audit rather than to the held-out action population of Section 4.4. Dream's saved trajectories contain no accuracy-level negative future gain, so this boundary is not observed there; Section 4.5 reports what that interface does expose.

One caveat belongs to the curve: the stopping depths of Section 4.4 describe learned allocation heterogeneity, not true sufficient depth, and say nothing about an optimal depth.

4.4 Adaptive halting on continuous interfaces

A readable signal earns its keep only if it changes where the computation goes: each trajectory quits when its own state says so, sparing the rest of the full run (Table 3, Appendix D).

Stopping early is close to free. On WMT14 the policy commits after 8.778 solver calls on average—13.7% of the 64-call trajectory, 86% less computation—and returns BLEU 26.2572 against 26.1466 for full inference, computing less and scoring higher (Table 1). XSum stops at 9.5398 calls, 14.9% of 64 and 85% less, at ROUGE-L 27.9232 against 27.811 for full inference. OpenWebText is the honest exception: 24.6265 calls of its native 32, 23% less, bought GenPPL 26.780 against 24.1432 for the full run. Quality peaks before the last call: on the WMT14 monitor pool mean chrF++ is 0.5407 at call 16, 0.5415 at call 19, and 0.5371 at call 64. A policy that stops on a good state is not cutting a corner—it declines to polish past the peak. These counts are nominal, replayed from saved trajectories, not measured online; no speedup is claimed (Figure 6).

CoLA's action check is exploratory: one seed, a post-hoc reader, no matched-budget claim. On the seed-66 held-out test partitions (4,897 trajectories across eight tasks), the conservative reader selects a task-macro average of 2.640 of four blocks, with 14 final-identity mismatches and no observed correctness loss relative to the final readout. It does not establish equal-call superiority or dominance over prediction stability; the two experiments support readable sample information and its conditional allocation value, but not lossless termination.

Table 1: Early stopping against full inference. Quality is in each task's native metric; the reference column is the same trajectory run to its final call.

Task (metric)	Policy	Full	Calls
WMT14 (BLEU ↑)	26.2572	26.1466	8.778/64
XSum (ROUGE-L ↑)	27.9232	27.811	9.5398/64
OWT (GenPPL ↓)	26.780	24.1432	24.6265/32

Ablations: fixed-budget controls. Better allocation, or merely a shorter run? Neither control below is the paper's main result. Against the longer-horizon anchors—fixed16 on WMT14 and XSum, fixed32 on OpenWebText—the policy trails by -0.1932 BLEU, -0.2008 ROUGE-L, and $+2.637$ GenPPL. Against equal-call rules at the policy's own average, it leads on all three: $+0.4639$ BLEU, $+0.0531$ ROUGE-L, and -0.9072 GenPPL against an exact-call fixed24/25 mixture at 27.687, all six seeds favouring the policy though mean entropy moves by only 0.000884. The ordering survives 1,000 predefined hash assignments, where the equal-call reference never overtakes the policy, and the five overlapping XSum rows, whose full-split fixed16 estimate (ROUGE-L 28.125) only exceeds the independent full64 parity estimate (27.811) as unpaired supplementary context. Intervals and cell values are in Appendix E; Table 3 sets these deltas beside the anchors. The controls confirm that allocation at a fixed budget is the better buy—not the full-inference result above.

4.5 The observable interface depends on architecture

Move the interface and the signal moves: Dream's masked-token state is not ELF's embedding flow. Its single-seed study covers 200 question identities under two conditions, 400 trajectories, and 7,200 frontier events; the matched feature analysis covers 6,800 events with hidden/logit features, and question identity is the unit of split and uncertainty.

Readiness is readable here as well, and it is mostly output-facing. The full reader reaches readiness AUROC 0.8419. Permuting same-event logit statistics lowers AUROC by 0.1453; permuting the five tested hidden summaries does not hurt performance (reported drop -0.0290) and retraining without them provides no evidence of benefit. The result neither exhausts the information in raw Dream hidden states nor completes the correspondence-and-action test run in the continuous settings. This

interface is analysed for predictive observability and feature source; we make no halting or allocation claim at it, so its evidence tier differs from Section 4.4.

4.6 Discussion and limitations

The evidence supports functional parseability: CoLA supplies sample-corresponding output information and ELF connects readable action signals to equal-call allocation gains. The protocol applies across the tested continuous interfaces, while targets and reader parameters stay task or architecture specific; Dream shows that predictive observability is not an identical capability across architectures.

Several limits define that scope. Targets depend on the chosen readout, normalization, or scorer; matching does not remove every nuisance; ELF correspondence tests use checkpoint-selection development data; WMT14's official test has prior use and its 1,920-example reservation stays unopened; and XSum's overlap exclusion does not establish clean monitor training. All three policies lose quality against their horizon anchors, positionwise supervision does not directly optimise absorbing first-exit utility, and replay never cancelled generator calls, leaving online stopping, batch compaction, integrated timing, and reader overhead untested. These gaps separate the observed allocation value from a deployable acceleration claim.

5 Conclusion

A continuous diffusion model thinks in latent vectors and pays for every one of them. We asked whether those vectors can be read early enough to stop, and for the tested models the answer is yes more often than not: on WMT14 and XSum the policy walks away after roughly one seventh of the calls that full inference would spend, and the text it produces is no worse. Unconditional generation is the harder case, saving less and paying a small perplexity cost, and at Dream's discrete masked-token interface we make no halting claim at all. What makes early stopping work is that quality is not monotone in computation: a trajectory usually peaks well before it stops, so its last calls are often its least valuable. Turning this into a measured speedup needs the readers wired into a live sampler, with their own overhead counted.

AI use statement

In this work, we used generative AI tools to aid and polish the writing of the manuscript, and for retrieval and discovery, such as identifying relevant literature. We have not used generative AI tools for any task requiring disclosure: no synthetic datasets were generated, no theoretical models or conceptual frameworks were developed, no mathematical claims or proofs were produced, no hypotheses or experimental methodology were proposed or reviewed, no methods were implemented, no translation was performed, no datasets were cleaned or reformatted, and no qualitative analysis or interpretation of results was assisted by these tools. All code, experiments, and analyses are the authors' own work; the quantitative figures and numerical summaries were produced by the authors' own scripts from recorded artifacts, and no AI tool performed model training, evaluation, or any access to the unopened WMT reservation. We have reviewed all AI-assisted text and take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Ethics statement

Licensing and release. This work analyses publicly released benchmark datasets — WMT14, XSum, OpenWebText, and MuSiQue, together with the eight-task CoLA evaluation suite — under their original terms. We do not redistribute any of these datasets; the companion package contains only our code, configurations, derived numeric summaries, and source digests. Public model checkpoints are used for offline evaluation only. The independent reservation of held-out WMT examples remains unopened.

Research use. The studies involve no human subjects, no new human annotation, and no personal data. Reported outputs are aggregate measurements on saved model trajectories and are intended for

research evaluation of computation allocation, not for deployment, content moderation, or decisions affecting individuals.

REPRODUCIBILITY STATEMENT

Section 3 specifies the measurement-and-action protocol: offline target construction, prefix-only readers, matched identification, and the frozen stopping decisions. Appendix A gives the target definitions, Appendix B the reader implementations, generator settings, and fitting details, and Appendix C the populations, splits, statistical units, matched controls, and frozen first-exit and equal-call references; Appendix E reports per-task and per-seed readouts with source digests. The companion package contains LaTeX sources, the figure and data-extraction scripts, the packaged numeric snapshots with source digests, and the scripts that regenerate them. Historical provenance gaps, the prior use of the WMT official test, the unopened independent reservation, the XSum overlap correction, and the absence of measured online cancellation or integrated timing are disclosed in Appendix F.

REFERENCES

Andrea Banino, Jan Balaguer, and Charles Blundell. PonderNet: Learning to ponder. arXiv preprint, 2021. 8th ICML Workshop on Automated Machine Learning (2021).

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/.

Simone Caldarella, Davide Talon, Rahaf Aljundi, Elisa Ricci, and Massimiliano Mancini. Thinking past the answer: Evaluating harmful overthinking in large reasoning models, 2026. URL https://arxiv.org/abs/2606.02835.

Shuochen Chang, Tong Bai, Xiaofeng Zhang, Qianli Ma, Qingyang Liu, Zhaohe Liao, Yibo Miao, and Li Niu. Unlocking the black box of latent reasoning: An interpretability-guided approach to intervention. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 34019–34032, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1568. URL https://aclanthology.org/2026.acl-long.1568/.

Guan-Ming Chiu and Jeng-Yue Liu. Probing functional correctness in diffusion language models. In Santosh T.Y.S.S., Juan Diego Rodriguez, and Ona de Gibert (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 163–172, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-393-7. doi: 10.18653/v1/2026.acl-srw.15. URL https://aclanthology.org/2026.acl-srw.15/.

Zhicheng Du and Lan Ma. Continuous language diffusion as a decoder-interface problem, 2026. URL https://arxiv.org/abs/2606.08810. arXiv:2606.08810v2.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. OpenWebText. Zenodo, version 1.0, 2020. URL https://doi.org/10.5281/zenodo.3834942.

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, ICLR*, 2023. URL https://openreview.net/forum?id=jQj-_rLVXsj.

Alex Graves. Adaptive computation time for recurrent neural networks. arXiv preprint, 2016.

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, 2026. URL https://arxiv.org/abs/2605.06548. arXiv:2605.06548v1.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He. ELF: Embedded Language Flows, 2026. URL https://arxiv.org/abs/2605.10938. arXiv:2605.10938v2.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082/.

Fan Li, Yu Gu, Zhigang Wang, Fangling Leng, Zhenghao Liu, and Ge Yu. Towards efficient and effective diffusion language model inference via semantic-aware adaptive denoising. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17988–18002, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.819. URL https://aclanthology.org/2026.acl-long.819/.

Pengxiang Li, Yefan Zhou, Dilxat Muhtar, Lu Yin, Shilin Yan, Li Shen, Yi Liang, Soroush Vosoughi, and Shiwei Liu. Diffusion language model knows the answer before it decodes. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 135235–135253, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/daadbff4d4ea884ca3d9d389a1dfc61c-Paper-Conference.pdf.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 4328–4343. Curran Associates, Inc., 2022. doi: 10.52202/068431-0313. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/1be5bc25d50895ee656b8c2d9eb89d6a-Paper-Conference.pdf.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Amr Mohamed, Guokan Shang, and Michalis Vazirgiannis. LESS is more: Mutual-stability sampling for diffusion language models, 2026a. URL https://arxiv.org/abs/2606.16908. arXiv:2606.16908v1.

Amr Mohamed, Yang Zhang, Michalis Vazirgiannis, and Guokan Shang. Fast-decoding diffusion language models via progress-aware confidence schedules. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 35793–35807, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1782. URL https://aclanthology.org/2026.findings-acl.1782/.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In Kevin Knight, Ani Nenkova, and Owen Rambow (eds.),

Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 839–849, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1098. URL https://aclanthology.org/N16-1098/.

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/.

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 Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144/.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Maximo Eduardo Rulli, Thomas Vaitses Fontanari, Simone Petruzzi, Federico Alvetreti, Giorgio Strano, Donato Crisostomi, Giorgos Nikolaou, Tommaso Mencattini, Andrea Santilli, Emanuele Rodolà, Simone Scardapane, and Alessio Devoto. Subliminal clocks: Latent time modelling in diffusion language models, 2026. URL https://arxiv.org/abs/2607.01774. arXiv:2607.01774v2.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *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, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Sofia Maria Lo Cicero Vaina, Nikita Balagansky, and Daniil Gavrilov. Diffusion language models generation can be halted early, 2024. URL https://arxiv.org/abs/2305.10818. arXiv:2305.10818v4.

Yang Xiang, Yixin Ji, Ruotao Xu, Dan Qiao, Zheming Yang, Juntao Li, and Min Zhang. When is thinking enough? early exit via sufficiency assessment for efficient reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23541–23556, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1080. URL https://aclanthology.org/2026.acl-long.1080/.

Roy Xie, Junlin Wang, Paul Rosu, Chunyuan Deng, Bolun Sun, Zihao Lin, and Bhuwan Dhingra. Knowing when to stop: Efficient context processing via latent sufficiency signals. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3afa7f9b7ac8da474b3d915570d58291-Abstract-Conference.html.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7B: Diffusion large language models, 2025. URL https://arxiv.org/abs/2508.15487. arXiv:2508.15487v1.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification, 2025. URL https://arxiv.org/abs/2504.05419.

A OPERATIONAL TARGETS AND OBJECTIVES

All teachers are computed offline from saved trajectories. Their use of future states, decoded candidates, reference answers, or task scores does not authorize those fields as reader inputs. Throughout this appendix, i indexes a sample trajectory, t an already completed computation step, and H the policy horizon. A positive action label denotes HALT; the implementation names this class STOP. The targets below measure different events and are not interchangeable measures of task sufficiency.

A.1 CoLA: A ONE-TIME REFERENCE-ENTRY EVENT

For each of the four outer blocks, let $a_{ib} = N(\hat{a}_{ib})$ be the stored scored prediction normalized by lowercasing, stripping, and collapsing whitespace. This normalizer does not remove punctuation or perform semantic matching. Let j_i be the earliest index satisfying $a_{ij_i} = a_{i,j_i+1} \neq \emptyset$. Define

$$a_i^\star = \begin{cases} a_{ij_i}, & j_i \text{ exists}, \\ a_{i4}, & \text{otherwise}, \end{cases} \qquad b_i^\star = \begin{cases} \min\{b : a_{ib} = a_i^\star\}, & a_i^\star \neq \emptyset, \\ 4, & a_i^\star = \emptyset. \end{cases} \tag{4}$$

The primary label is $C_{ib} = \mathbf{1}[b = b_i^\star]$. Every trajectory therefore has exactly one positive event, including an all-empty trajectory. The auxiliary post-entry label $I_{ib} = \mathbf{1}[b \geq b_i^\star]$ records position relative to this boundary; it does not assert that later strings remain identical. Official correctness, $Q_{ib} = \mathbf{1}[\text{score}_{ib} \geq 1]$, is constructed separately. In particular, C_{ib} uses neither gold correctness nor a requirement of permanent stability.

The CoLA reader has seven active binary heads: reference entry, correctness, difference from the reference prediction, nonempty prediction, decoder stop, completion risk, and post-entry state. Their loss weights are respectively $1, .5, .75, .25, .25, .75, .75$. A future-correctness-gain regression head has weight $.25$. Binary cross-entropies use training-split positive-class weights; regression uses squared error. Completion risk is positive for an empty current prediction with a nonempty reference, or for a strict prefix of that reference; it is zero for an empty reference or an exact match. The reference-difference head tests whether the current normalized prediction differs from $a_i^\star$, provided either is nonempty. These auxiliary targets do not enter the decision-time feature tensor. The empty-answer and answer-format heads available in the code are disabled in the canonical matrix.

Descriptive observation population. Figure 8 uses the saved test trajectories associated with the 24 canonical aligned-reader cells (eight tasks and three composite seeds), counting each trajectory once rather than repeating it for the three input variants. We reconstruct Eq. 4 from the four stored scored predictions and verify agreement with the stored entry labels. The 14,729 sample-seed trajectories yield block counts 9,367/3,242/399/1,721. Of the final-block category, 1,236 have empty references and 485 have nonempty references; 1,219 trajectories have four empty scored predictions. An empty scored prediction can reflect output extraction and must not be interpreted as absence of all generated text. No reader score or correctness filter enters this descriptive aggregation. The distribution is trajectory-weighted, unlike the task-macro reader metrics, and repeated underlying benchmark identities across composite seeds are not independent samples. The companion motivation snapshot and audit provide task/seed counts and source digests.

A.2 ELF WMT14 AND XSUM: FUTURE-QUALITY ACTION SUPERVISION

At completed call t, a frozen shared-weight readout maps the clean embedding prediction $\hat{x}_{it}$ to a candidate. For WMT14, q_{it} is sentence chrF++ divided by 100, implemented with SacreBLEU's CHRF metric and word order 2. For XSum it is sentence ROUGE-L F1. These are teacher scores; the reported WMT14 task metric is corpus BLEU. For $t = 1, \ldots, 15$ and $H = 16$,

$$q_{it}^+ = \max_{t<u\leq H} q_{iu}, \qquad y_{it} = \mathbf{1}[q_{it} \geq q_{it}^+], \qquad r_{it} = \max(0, q_{it}^+ - q_{it}). \tag{5}$$

Equality selects HALT. All supervised positions remain in the training set, including positions after an earlier positive target. Call 16 is a forced boundary and is not a learned action.

Let μ be the mean positive regret over unsafe positions in monitor training data only. For reader logits ℓ_{it}, the canonical quality-protected loss is

$$w_{it} = 1 + \frac{r_{it}}{\mu}, \qquad \mathcal{L}_{\text{action}} = \frac{\sum_{i,t<H} w_{it}\, \text{CE}(\ell_{it}, y_{it})}{\sum_{i,t<H} w_{it}}. \tag{6}$$

The normalization is applied over positions in each batch. Frozen values are $\mu = 0.0669996451$ for WMT14 and 0.0766104629 for XSum. Unit weight is retained everywhere, with an additive penalty for prematurely stopping at unsafe positions. This is the E27 WMT14 objective; earlier losses proportional to regret alone, threshold policies, and minimum-call floors are not used. Optimizing positionwise action loss remains an approximation to first-exit utility: improved action prediction need not preserve the final policy's anchor quality.

A.3 OpenWebText: a decoder-relative cost-adjusted teacher

OpenWebText has no reference text. Let v_{it} be the token sequence produced by the frozen mapping at call t, and v_{iH} the terminal mapping on the same stochastic trajectory, with $H = 32$. Let $n(v)$ count non-padding tokens and $m(v, v')$ count equal, aligned token positions at which both tokens are non-padding. The teacher distance is

$$d_{it} = 1 - \frac{m(v_{it}, v_{iH})}{\max\{n(v_{it}), n(v_{iH}), 1\}}. \tag{7}$$

This distance penalizes aligned disagreement and length mismatch; it is neither a reference-based quality score nor a language-model perplexity. Dynamic programming defines

$$\begin{aligned} V_{iH} &= 0, & V_{it} &= \min\{d_{it}, \lambda + V_{i,t+1}\}, \\ y_{it} &= \mathbf{1}[d_{it} \le \lambda + V_{i,t+1}], & r_{it} &= \max\{0, d_{it} - \lambda - V_{i,t+1}\}. \end{aligned} \tag{8}$$

The teacher charges λ per additional call. A 60-iteration binary search on training trajectories chooses the cost whose first-positive oracle mean is closest to 24 calls, breaking equal-distance ties toward smaller cost. The frozen value $\lambda = 0.0273972703$ gives a training oracle mean of 23.9971 calls and a validation oracle mean of 24.0156. The reader uses Eq. 6 with $\mu = 0.2201549826$, supervised calls 1–31, and a forced call-32 boundary. GPT-2 Large is used only for the frozen GenPPL evaluation, never to construct this teacher or supply reader features.

A.4 Dream: correctness and final-answer agreement

For a decoded step probe $\hat{a}_{it}$ and the final predicted answer $\hat{a}_{iT}$, the Dream target is

$$R_{it} = \mathbf{1}[\mathrm{QA}(\hat{a}_{it}) \ge 1]\, \mathbf{1}[N_D(\hat{a}_{it}) = N_D(\hat{a}_{iT})], \tag{9}$$

where QA is the stored answer/alias scorer and N_D its answer normalizer. Agreement is evaluated at the current event; it does not certify that every later event agrees. This correctness-bearing target differs from CoLA's one-time landmark. Gold answers, aliases, decoded strings, and final-answer identity are offline-only. The loss combines readiness binary cross-entropy, next-probe prediction-change and final-match binary cross-entropies (weight .5 each), and future score-gain squared error (weight .25); only readiness receives training-derived class weighting.

B Readers, generators, and fitting

B.1 CoLA reader and recovered trajectories

Each CoLA block is a 16×16 latent tensor. The reader projects slots to width 64, applies one intra-block Transformer layer with four attention heads, and pools each block into four learned attention summaries plus the explicitly retained final slot (`pma4_last`). A trajectory interaction token combines process information with block summaries, followed by two inter-block Transformer layers and separate readout heads. Dropout is .1. Inputs contain only the available latent prefix, its mask, and four budget fields: block number, maximum budget, remaining blocks, and block fraction. The canonical setting has no task-conditioning or readout-context input. Prefix masks exclude future blocks, but prefix length and positions retain clock information.

Training uses AdamW, learning rate 2×10^{-4}, weight decay 10^{-4}, batch size 1,024, 20 epochs, gradient clipping at 1, and validation every ten optimizer steps. Checkpoint selection maximizes the mean validation AUROC for reference entry, prediction change, completion risk, and post-entry state; when this statistic is undefined, selection falls back to negative validation loss. Only the selected best checkpoint is used for formal evaluation.

The recovered generator setting has a 64-token maximum, four outer blocks of 16 slots, 16 denoising intervals per block, a linear 1000-to-0 noise schedule, CFG 7, temperature 0, top-k 50, top-p .9, and repetition penalty 1.1. Generation uses bfloat16 on CUDA and stores float32 latents. The three composite indices 66/67/68 correspond to generation seeds 20260524/20260525/20260526 and sample-noise seeds 66/67/68. The same indices also affect reader initialization, split assignment, and shuffle mappings; they are not isolated optimizer replications.

B.2 ELF READERS AND NATIVE SAMPLER SETTINGS

All ELF generators, conditioning encoders, and output mappings remain frozen. The official ELF-B checkpoints are `checkpoint_880600` (WMT14 De→En), `checkpoint_39800` (XSum), and `checkpoint_95085` (OpenWebText). The retained upstream source revision is `b29d8833609e9ab7f67cd9da39435ac5cea04837`. WMT14 and XSum use the official 64-call ODE/logit-normal schedule, input CFG 2, and self-conditioning CFG 1; the policy observes its first 16 completed calls. Fixed16 is therefore a prefix of this schedule, not a separately rescheduled 16-call generator. OpenWebText uses 32-call SDE sampling, logit-normal time, self-conditioning CFG 3, and $\gamma = 1.5$.

For WMT14/XSum, `fixed_moment_v2` extracts slot-RMS-normalized mean content, a fixed linear slot-position moment, normalized adjacent-call differences, log-scale moments, and adjacent-state cosine. These use the current and immediately previous clean states only, with zero initial differences. The feature groups are projected to width 32 and fused into a 128-dimensional state with LayerNorm, a linear layer, GELU, and dropout .1. A single unidirectional GRU of hidden width 128 produces two action logits. The five process values are solver time, normalized step index, its complement, normalized condition length, and a latent-channel presence flag. The step denominator remains 63 for the native 64-call schedule; condition-length scales are 64 for WMT14 and 1,024 for XSum. Generated windows are aligned to each sample's condition length.

OpenWebText separately summarizes $\hat{x}_t$ and $z_t^{\mathrm{post}} - \hat{x}_t$, where z_t^{post} is the state after the stochastic update. Each 512-dimensional stream contributes $3 \times 512 + 5$ fixed-moment features, yielding 3,082 values in total. The `raw_mean_v1` reader normalizes and projects this saved summary to width 96, fuses process information, and uses a width-128 unidirectional GRU. Its clock denominator is 31. Keyed noise is fixed by generation seed, sample ID, and call index; this makes saved trajectories addressable but does not by itself establish online batch-compaction equivalence.

All three task-specific readers use AdamW with learning rate 3×10^{-4}, weight decay 10^{-4}, training/validation batch sizes 8/16, dropout .1, bfloat16 training, gradient clipping at 1, and 1,000 optimizer steps. Validation occurs every ten steps. The best checkpoint minimizes validation quality-protected action cross-entropy, with earlier-step tie breaking. The three initialization seeds are 20260720, 20260721, and 20260722. Selected RB steps are 990/920/890 for WMT14, 940/720/820 for XSum, and 920/830/990 for OpenWebText. The six paired runs per task have finite training curves and select checkpoints before the last step. They share task trajectories, and therefore measure initialization sensitivity rather than independent data replication.

B.3 DREAM FEATURES AND FITTING

The full Dream reader has 16 features: normalized step and event indices, availability flags, mask count, changed-token count, mean top-1 probability, mean top-2 margin, entropy, five hidden summaries, and two condition indicators. The hidden summaries are mean, standard deviation, mean absolute value, mean token L2 norm, and last-token L2 norm. Thus this is an output-facing reader, not a latent-only analogue of the continuous readers. Features are standardized using training data. A causal two-layer, four-head Transformer of width 128 and dropout .1 is fitted with AdamW at learning rate 2×10^{-4}, weight decay 10^{-4}, batch size 32, gradient clipping 1, and validation every ten steps for 40 epochs. Checkpoint selection maximizes validation readiness AUROC plus .25 times each of final-match and change AUROC, minus .1 times future-gain MSE. The recorded full-model run selects step 400, which is also its final step. The fourteen feature modes use a single initialization seed, 20260718. Removing or permuting five summaries tests those summaries, not the information content of the unrecorded full hidden tensor.

C Populations, controls, and action comparisons

C.1 Split membership and statistical units

CoLA contains 49,019 task–sample identities: LAMBADA 5,153 (Paperno et al., 2016); MMLU 14,042 (Hendrycks et al., 2021); OpenBookQA 500 (Mihaylov et al., 2018); HellaSwag 10,042 (Zellers et al., 2019); RACE 4,887 (Lai et al., 2017); Social IQa 1,954 (Sap et al., 2019); SQuAD 10,570 (Rajpurkar et al., 2016); and Story Cloze 1,871 (Mostafazadeh et al., 2016). Three generator realizations yield 147,057 sample–seed trajectories and 588,228 block events. Leave-one-task-out fitting uses the other seven tasks. Trajectory-level hash splits allocate 80/10/10% to training, validation, and internal testing; every prefix stays with its trajectory. Target-task policy calibration uses its validation partition, with replay evaluation on its test partition. Consequently, held-out-task discrimination does not imply strictly zero-shot policy calibration.

Each ELF task uses 4,096 monitor-training trajectories and 512 development-validation trajectories. Collection seeds are 50/51 for WMT14, 60/61 for XSum, and 50/54 for OpenWebText. The frozen action benchmarks contain 3,003 official WMT14 test examples, 11,329 XSum examples after removing five overlaps, and six fresh OpenWebText generation seeds (6–11) with 1,000 examples each. Dream contains 200 MuSiQue sample identities (Trivedi et al., 2022) under two conditions: full evidence and a matched textual multi-agent condition. Identity-level splits retain both conditions together, giving 160/20/20 identities and 320/40/40 sequences. The frontier has 7,200 events; 6,800 have the matched hidden/logit statistics used for same-event carrier comparisons.

CoLA sample-cluster intervals express within-task sampling uncertainty; task-cluster intervals express heterogeneity over eight tasks and can cross zero despite favorable sample-level aggregates. Block events and multiple realizations of an identity are not independent task replications. ELF matched-control losses are checkpoint-selection development results. WMT14 policy differences use source-paired corpus-BLEU bootstrap, whereas XSum differences use paired per-example ROUGE-L resampling. OpenWebText reported standard errors are across six generation seeds. Hash-assignment sensitivity assesses the specified equal-call rule, not uncertainty over newly trained readers. Dream inference clusters by sample identity rather than hook event.

C.2 Matched controls and reader dependence

CoLA RB uses aligned raw latents; SB routes a coherent other-sample prefix within task and deterministic split, preserving block positions. A deterministic hashed ordering followed by a one-position rotation avoids self-pairs. ZB zeros raw latents. All arms retain the same budget fields. SB tests sample correspondence under these matches; ZB is an out-of-distribution ablation. Answer identity, difficulty, latent norm, and decoder margin are not matched. The canonical manifest contains all 72 fits: three composite seeds, eight held-out tasks, and three variants.

ELF RB/SB pairs match split, call/time schedule, sampler, CFG, generated-window shape, and condition-length bins, while transferring coherent trajectories for SB. Length matching is binned rather than exact. The recipient's process inputs are retained. These matches preserve major clock and collection conditions, but do not identify generator causality or remove every sample-level nuisance.

The CoLA frozen-reader analysis loads the 24 canonical RB best checkpoints and changes inputs at inference without retraining. Perturbations include sample-prefix shuffle, order reversal, current-block zeroing, history removal, process mean-fill, and slot/dimension changes. Reported drops are baseline minus perturbed performance. They describe dependence of these trained readers; pooling and normalization mediate the effects. In particular, retaining the final slot makes slot rankings architecture-dependent, and process mean-fill leaves mask, prefix-length, and position clocks available. Geometry and entropy associations are likewise observational, not generator interventions.

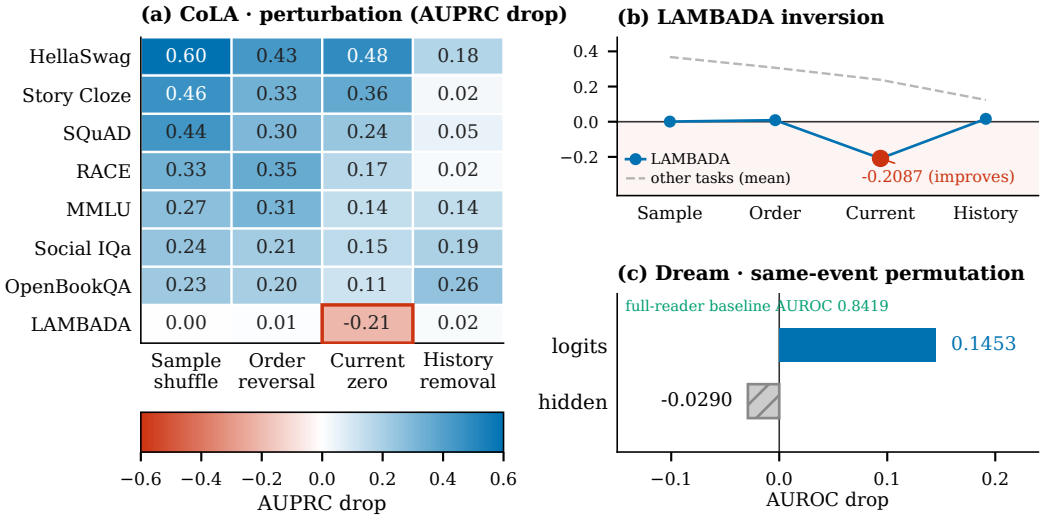

Figure 4: Reader dependence varies across inputs and architectures. (a) AUPRC drop for a frozen CoLA reader under four input perturbations across eight tasks; sample shuffle, order reversal, and current-block zeroing all degrade performance, while history removal is the mildest. (b) LAMBADA is the only inversion: zeroing the current block improves it by 0.2087 AUPRC while every other task's drop stays positive. (c) Dream reports AUROC drops under same-event, same-condition feature permutations in a single-seed study: permuting logit statistics costs 0.1453 against a baseline of 0.8419, whereas permuting the hidden summaries yields -0.0290. Negative drops indicate improvement. The metrics and experimental units differ; these tests characterize reader dependence, not generator causality.

C.3 Frozen first-exit rules and equal-call references

For every ELF task, average the action logits of its three selected RB checkpoints: $\bar{\ell}_{it} = \frac{1}{3}\sum_{k=1}^{3} \ell_{it}^{(k)}$. The policy stops at

$$\tau_i = \min\big(\{t < H : \bar{\ell}_{it,\mathrm{HALT}} \geq \bar{\ell}_{it,\mathrm{CONTINUE}}\} \cup \{H\}\big). \tag{10}$$

There is no threshold search, temperature scaling, logit bias, call floor, or persistence rule. Checkpoints and targets are task-specific. A decision follows an already completed call and can save only later calls. One frozen output mapping is allowed after HALT.

The equal-call WMT14/XSum reference assigns each example a fixed prefix of length $\lfloor\bar{\tau}\rfloor$ or $\lceil\bar{\tau}\rceil$, where $\bar{\tau} = N^{-1}\sum_i \tau_i$. Exactly $\sum_i \tau_i - N\lfloor\bar{\tau}\rfloor$ examples receive the longer prefix, selected by a fixed sample-ID hash namespace, independently of content and quality. Quality is evaluated on the resulting corpus, not interpolated between two corpus metrics. WMT14 official test uses calls 8/9; decontaminated XSum uses calls 9/10. OpenWebText uses an exact-total-call fixed24/25 mixture separately within each generation seed, with its own predefined hash assignments. The 1,000-assignment sensitivity for XSum and OpenWebText varies allocation hashes while preserving total calls; it does not choose a favorable policy threshold.

CoLA's 0.5% final-identity mismatch-cap replay is post-hoc exploratory evidence. On the seed-66 held-out test partitions, comprising 4,897 trajectories across eight tasks, the conservative replay uses a task-macro average of 2.640/4 blocks with 14 final-identity mismatches and no observed correctness loss. This policy population is distinct from the 147,057 trajectories used for the full three-seed dynamics audit and the 14,729 test trajectories used for the motivating observation. The replay does not establish equal-call superiority or dominance over prediction stability. Retrospective oracle depths indicate available headroom and are not deployable policies. All reported action costs are saved-trajectory block/call counts; monitor overhead, online cancellation, batch compaction, and integrated wall-clock acceleration have not been established.

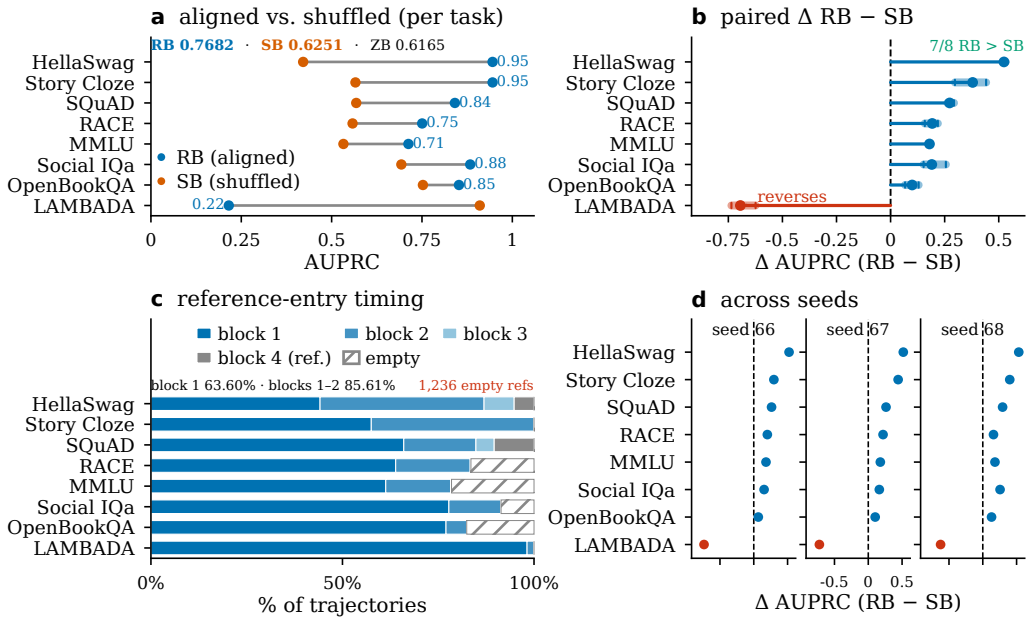

Figure 5: Closure experiment: CoLA prefixes carry output information beyond matched computation position. (a) Per-task AUPRC for aligned (RB) and matched-shuffled (SB) readers; macro AUPRC 0.7682 against 0.6251, zero-latent control 0.6165. (b) Paired RB–SB differences over three composite seeds, as a three-seed mean with the observed seed range: seven tasks are positive for every seed, and LAMBADA reverses the direction. (c) Reference-entry timing over 14,729 saved trajectories: block one for 63.60%, block two for 85.61%, with 1,236 empty-reference trajectories separated out. (d) The RB–SB sign is stable across the three composite seeds. Ranges describe observed seeds, not confidence intervals, and the aggregate gain does not establish a uniformly positive effect across task populations.

D ADDITIONAL TABLES AND FIGURES

Owing to space constraints these tables and figures, which are referenced from the main text, are collected here. Their captions and content are unchanged.

Table 2: Architecture-specific targets and decision interfaces. The protocol is reused across settings; targets and reader parameters are not shared. Dream is a separate predictive analysis with output-facing inputs.

Setting	Reader input	Offline target
CoLA	Latent-block prefix and budget fields	First entry of a decoder-defined reference prediction
ELF	Completed-call flow features and process fields	Task-specific quality/cost action
Dream	Token/mask/logit history and tested hidden summaries	Current correctness and final-answer agreement

E PER-TASK AND PER-SEED READOUTS

This appendix reports the cell-level values behind the aggregates in Section 4. Every number is a direct reduction of the frozen artifacts listed at the end of this appendix; no value is transcribed by hand. Cell-level values are not interchangeable with the macro aggregates in the main text, which combine cells under the stated statistical unit.

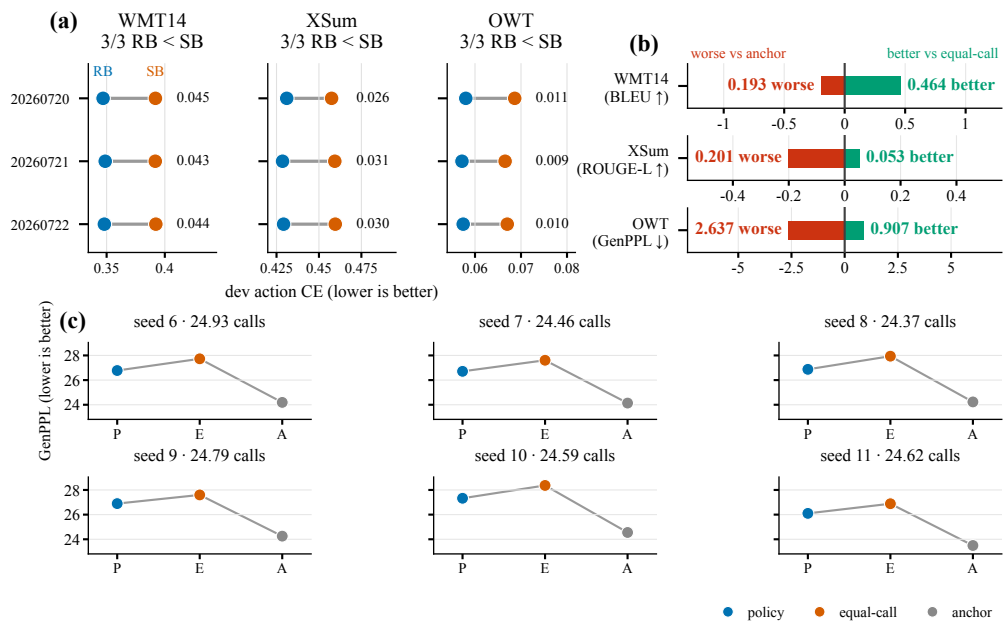

Figure 6: From readable ELF states to allocation at a fixed call budget. (a) Development-only pairing of aligned (RB) and shuffled (SB) prefixes: action cross-entropy is lower for the aligned prefix in all three initializations of every task, and each task keeps its own axis because targets and weighting schemes differ. (b) Frozen saved-trajectory replay: the policy's gain over a non-adaptive equal-call reference and its gap to a longer-horizon anchor, in each task's own metric, direction normalized so rightward is better; the policy leads the reference on all three tasks and trails the anchor on all three. (c) Per-seed OpenWebText generation perplexity for policy, reference, and anchor. Call counts are matched between policy and reference, and no panel measures online runtime.

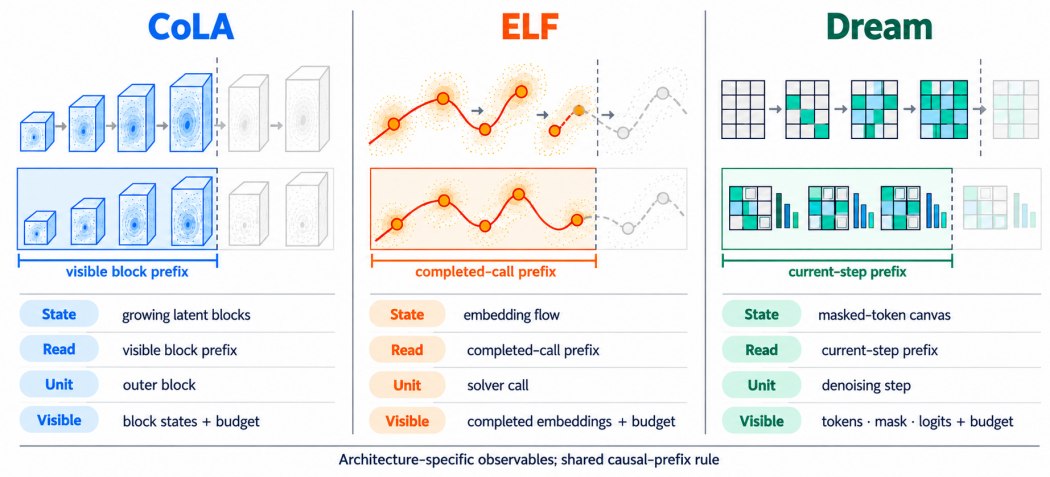

Figure 7: Architecture-specific observables under a shared causal-prefix rule. For each model we give the native state, the prefix a decision-time reader may observe, the computation unit at which a decision is taken, and the observables available at that unit. The readable interface differs across architectures while the causal-prefix rule is shared.

Table 3: Reading and halting across the two continuous interfaces. Both readers consume only a prefix of the native state plus process fields. The CoLA policy is a post-hoc exploratory audit on one seed; the ELF policies are frozen before evaluation. References and intervals are reported in the text and in Table 1.

	CoLA	ELF
Native state	growing latent-block prefix	completed-call embedding-flow prefix
Compute unit	outer block	solver call
Decision input	latent prefix and budget fields	flow features and process fields
Evidence tier	post-hoc exploratory replay	three frozen task-specific policies
Allocation outcome	2.640/4 blocks (task macro), 14 identity mismatches	$+0.4639$ BLEU, $+0.0531$ ROUGE-L, -0.9072 GenPPL over equal-call rules
Longer-horizon reference	no observed correctness loss versus the final readout	-0.1932 BLEU, -0.2008 ROUGE-L, $+2.637$ GenPPL

Table 4: Information boundary by stage. Offline construction may inspect the whole trajectory; decision time may not. Dream's interface is output-facing by construction and is analyzed separately, so it is listed under its own row rather than folded into the continuous boundary.

Stage	Permitted information
Offline measurement	Intermediate decoded text, normalized identity, references, task scores, future and final states
Reader training	Current and historical latent/process prefix, process fields, offline teacher targets
Decision time, continuous settings	Current and historical latent/process prefix and process fields, plus one output mapping after halting
Decision time, excluded	Decoded text, token logits, references, task scores, future or final states, test-label-derived thresholds
Dream analysis	Native token, mask, and logit history, and the tested hidden summaries

E.1 CoLA reference-entry AUPRC by task and composite seed

E.2 CoLA earlier-correct / final-wrong trajectories by task and seed

E.3 ELF allocation and horizon references

E.4 Computation–quality readouts

E.5 Frozen-reader perturbation readouts

Drops are baseline minus perturbed macro AUPRC for the 24 canonical CoLA RB best checkpoints, changing inputs at inference without retraining. A negative value indicates an improvement.

Same-event, same-condition permutation on Dream's single seed changes AUROC by 0.1453 for logit statistics and -0.0290 for the five tested hidden summaries, against a full-reader AUROC of 0.8419. The statistical units are recorded in Appendix C.1; source digests are listed below.

E.6 Source digests

The tables above are generated from the packaged snapshots `figures/data_results.json`, `figures/data_dynamics.json`, and the canonical label inventory. Digests:

- `data_results.json`: `248c73634f16334edd9ab63069bdf097`.
- `data_dynamics.json`: `ed12e6b67d5d7d16f5c5d6778e765d42`.
- `label_inventory.csv`: `f10577f4b85a336b49986858ae6bcc1e`.
- WMT14 monitor pool: 8 shard digests, recorded in `data_dynamics.json`; metric sentence chrF++ (word order 2, character order 6, beta 2), role `monitor_validation`, 512 samples over 64 calls, from `elf_e19_wmt_quality_targets`.
- Dream frontier: `frontier_events.jsonl`, 7,200 events over 400 condition rows; mean step accuracy uses question identity as the unit.

Table 5: Computation–quality endpoints on saved trajectories. WMT14 compares two depths of the same trajectories. XSum compares independently collected estimates and is not paired. CoLA and Dream report trajectory-level reversals. No intermediate depth is interpolated from these endpoints.

Setting	Observation	Direction
WMT14, paired, same trajectories	fixed16 BLEU 26.4504 versus 64-call endpoint 26.1466	shorter better by 0.3038
XSum, not paired	full-split fixed16 ROUGE-L 28.125 versus independent full64 parity 27.811	shorter higher; context only
CoLA, trajectory level	216 of 147,057 earlier-correct/final-wrong	degradation exists, rare
Dream, trajectory level	0 accuracy-level negative future gain	not observed

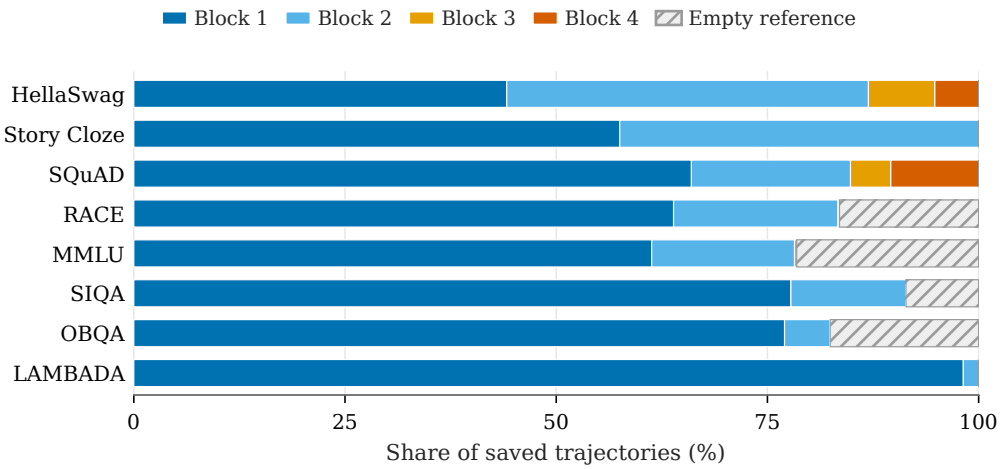

Figure 8: CoLA output-entry timing varies across saved trajectories. The descriptive distribution uses 14,729 canonical test trajectories, pooled across three composite seeds, without correctness filtering. Nonempty reference predictions are grouped by their first occurrence; empty-reference fallback is shown separately. These are offline decoded-identity measurements, not reader predictions, stable-correct-answer rates, or achievable computation savings.

F Qualitative trajectories and how the entry label behaves

The reference-entry event is defined on the decoded trajectory, so it is worth showing what it looks like on individual saved trajectories rather than only in aggregate. Table eftab:examples gives two saved CoLA test trajectories from the same held-out task and the same composite seed: one where the identity is already present in block one, and one whose first block is empty and whose identity therefore enters in block two. Both predictions are stable once the identity appears.

Aggregated over the same population, the label behaves much more like an absorbing event than a drifting one. Of 14,729 saved trajectories, 13,008 take their reference from an adjacent repeat and 1,721 fall back to the final identity because no adjacent repeat exists. Only 1 trajectory changes its decoded identity after the entry block, so the position at which the identity first appears is, in this population, essentially the position from which it no longer moves. The remaining categories qualify that reading: 1,236 trajectories have an empty reference and are assigned the final block by convention, 1,219 of those are empty in all four blocks, and 485 trajectories have a nonempty identity that genuinely appears only in the last block. The two cases must not be pooled: an empty prediction can reflect how the output was extracted rather than an absence of generated text.

Three limits on this measurement follow from how it is constructed. No trained-reader scores used, but targets and scored predictions are serialized alongside scores in canonical student_scores.jsonl. Empty reference is assigned to final block by label definition and must be separated from genuinely late nonempty identity. Reference entry is retrospective, decoder-relative, not correctness or a certified stopping time; no gold-correct selection was applied. Seeds are composite pipeline seeds, not independent benchmark replications. Pooled fractions are trajectory-weighted, not task-macro. The

Table 6: CoLA reference-entry AUPRC per task and composite seed. RB is the aligned reader, SB the matched shuffled reader. The final row is the unweighted mean over the 24 seed–task cells, which reproduces the macro AUPRC reported in Section 4.2.

Task	Seed	RB	SB	RB − SB
HellaSwag	66	0.9309	0.4088	+0.5221
	67	0.9496	0.4317	+0.5179
	68	0.9554	0.4231	+0.5323
Story Cloze	66	0.8609	0.5634	+0.2975
	67	0.9930	0.5513	+0.4417
	68	0.9820	0.5831	+0.3989
SQuAD	66	0.8331	0.5700	+0.2631
	67	0.8217	0.5591	+0.2626
	68	0.8683	0.5751	+0.2931
RACE	66	0.7793	0.5785	+0.2008
	67	0.7524	0.5338	+0.2185
	68	0.7198	0.5615	+0.1582
MMLU	66	0.7245	0.5436	+0.1809
	67	0.6904	0.5121	+0.1782
	68	0.7225	0.5421	+0.1803
Social IQa	66	0.8975	0.7448	+0.1526
	67	0.8471	0.6837	+0.1634
	68	0.9058	0.6502	+0.2556
OpenBookQA	66	0.8752	0.8080	+0.0672
	67	0.7772	0.6748	+0.1024
	68	0.9046	0.7750	+0.1296
LAMBADA	66	0.2092	0.9450	-0.7358
	67	0.2188	0.9421	-0.7233
	68	0.2183	0.8424	-0.6242
Mean over cells	—	0.7682	0.6251	+0.1431

distribution is trajectory-weighted and repeats underlying benchmark identities across composite seeds, so it describes how the label behaves rather than how independent samples are distributed.

G Fitting, selection, and compute protocol

G.1 The CoLA matrix and how cells enter the aggregate

The CoLA comparison is a 72-cell matrix over a scientific unit of seed x held-out task x latent-input control: seeds 66, 67, 68 crossed with the held-out tasks lambada, mmlu, obqa, hellaswag, race, siqa, squad, story_cloze and the latent-input variants raw, zero, shuffle. All 72 expected cells completed. The run used calibration subseed `20260525` and the run tag `20260718`.

Two rules govern how cells are allowed to enter the paper aggregate. First, consume only the explicit canonical_train_dir/canonical_eval_dir in complete cells; never glob partial or interrupted directories into the paper aggregate. Second, the artifact tree is write-once: immutable directories stay immutable, quarantined_and_immutable, and every retry must write to the manifest's recommended new directory. The recorded retry tags are `retry1_after_gpu_recovery_20260718`, `retry2_after_sigterm_20260719`, so the interrupted and GPU-recovered attempts are visible rather than absorbed into the reported numbers.

G.2 Reader configuration and the fitting loop

The ELF readers are the `raw_mean_v1` family with state dimension 96, hidden dimension 128, dropout 0.1, and 2 outputs; the solver horizon is 32 calls and the condition-length scale is 1.

The student sees only `xhat_current`, `step_index`, `t`, `condition_length`; the target-only fields `instantaneous_stop`, `unsafe_regret` are never inputs, and `teacher_fields_available_to_student` is no. The fitting objective is `future_best_quality_protected_instantaneous_action` with loss `all_position_action_`

Table 7: Trajectory-level quality reversals in CoLA. Each cell counts sample–seed trajectories whose reference identity was correct early in the trajectory but wrong at the final readout, over the trajectories in that task and seed. Undefined rates mean no reversals were observed in that cell.

Task	Seed	Reversals	Trajectories	Rate
HellaSwag	66	4	10042	0.00040
	67	1	10042	0.00010
	68	2	10042	0.00020
	aggregate	7	30126	0.00023
Story Cloze	66	0	1871	0.00000
	67	0	1871	0.00000
	68	0	1871	0.00000
	aggregate	0	5613	0.00000
SQuAD	66	42	10570	0.00397
	67	36	10570	0.00341
	68	39	10570	0.00369
	aggregate	117	31710	0.00369
RACE	66	8	4887	0.00164
	67	9	4887	0.00184
	68	6	4887	0.00123
	aggregate	23	14661	0.00157
MMLU	66	24	14042	0.00171
	67	24	14042	0.00171
	68	16	14042	0.00114
	aggregate	64	42126	0.00152
Social IQa	66	0	1954	0.00000
	67	0	1954	0.00000
	68	0	1954	0.00000
	aggregate	0	5862	0.00000
OpenBookQA	66	0	500	0.00000
	67	0	500	0.00000
	68	0	500	0.00000
	aggregate	0	1500	0.00000
LAMBADA	66	1	5153	0.00019
	67	1	5153	0.00019
	68	3	5153	0.00058
	aggregate	5	15459	0.00032

Table 8: ELF frozen policies, equal-call rules, and horizon anchors. Quality is in each task's native metric; calls are mean solver calls on saved trajectories. OpenWebText rows are the six generation seeds behind the arithmetic mean used in the main text.

Task	Arm	Calls	Quality
WMT14	Adaptive policy	8.778	26.2572
	Equal-call rule	8.778	25.7933
	Horizon anchor	16.000	26.4504
XSum	Adaptive policy	9.540	27.9232
	Equal-call rule	9.540	27.8700
	Horizon anchor	16.000	28.1239
OpenWebText	Adaptive policy (mean)	24.626	26.7797
	Equal-call rule (mean)	24.626	27.6869
	Horizon anchor (mean)	32.000	24.1432
seed 6	policy	24.926	26.7758
seed 7	policy	24.465	26.7096
seed 8	policy	24.366	26.8724
seed 9	policy	24.790	26.8937
seed 10	policy	24.594	27.3262
seed 11	policy	24.618	26.1006
seed 6	equal-call	24.926	27.7262
seed 7	equal-call	24.465	27.6084
seed 8	equal-call	24.366	27.9397
seed 9	equal-call	24.790	27.5974
seed 10	equal-call	24.594	28.3663
seed 11	equal-call	24.618	26.8836
seed 6	anchor	32.000	24.1909
seed 7	anchor	32.000	24.1362
seed 8	anchor	32.000	24.2289
seed 9	anchor	32.000	24.2549
seed 10	anchor	32.000	24.5528
seed 11	anchor	32.000	23.4954

Table 9: WMT14 monitor-validation computation–quality readouts. Mean sentence chrF++ and the share of samples below their own earlier maximum, by solver call, on the 512-example monitor-validation pool. The main-text aggregate rests on the official-test endpoints, a different population.

Call	1	4	8	16	19	32	48	64
Mean chrF++	0.3616	0.5174	0.5385	0.5407	0.5415	0.5397	0.5380	0.5371
Below own peak	0.000	0.316	0.566	0.693	0.709	0.744	0.773	0.785

Table 10: Dream frontier accuracy by denoising step. Mean step accuracy over the recorded frontier events at each step, single seed, with question identity as the statistical unit. Step 0 carries no prediction and is omitted. No trajectory-level reversal is observed: 0 of 400 conditionrows falls below its own earlier accuracy.

Step	4	8	16	32	48	63
Mean accuracy	0.4600	0.5250	0.5250	0.5250	0.5250	0.5275

Table 11: CoLA frozen-reader AUPRC drops by task and perturbation. Each cell changes one input aspect while holding the trained reader fixed. The final row is the unweighted mean over the eight task values. These are reader-dependence measurements, not generator interventions.

Task	Sample shuffle	Order reversal	Current zero	History removal
HellaSwag	0.5997	0.4332	0.4843	0.1766
Story Cloze	0.4615	0.3284	0.3646	0.0249
SQuAD	0.4449	0.3045	0.2419	0.0544
RACE	0.3288	0.3545	0.1657	0.0193
MMLU	0.2674	0.3094	0.1435	0.1410
Social IQa	0.2443	0.2140	0.1533	0.1937
OpenBookQA	0.2287	0.2031	0.1130	0.2571
LAMBADA	0.0008	0.0088	-0.2087	0.0164
Mean	0.3220	0.2695	0.1822	0.1104

Table 12: Two saved CoLA test trajectories and their entry labels. Predictions are the decoded identities after normalization, one per outer block; the reference is the first identity repeated in two adjacent blocks. Entry is retrospective and decoder-relative, not a correctness label.

Task	Trajectory	Decoded identity per block	Reference	Entry block	Changed after entry
mmlu	seed 66, sample 2	A / A / A / A	a	1	no
mmlu	seed 66, sample 5	(empty) / A / A / A	a	2	no

`ce_with_additive_unsafe_regret_cost`, and the horizon anchor used to supervise it is call 32, whose semantics are recorded as `forced_budget_boundary_not_learned`.

G.3 Deployment rule and what was deliberately left untuned

At evaluation the policy follows one fixed rule: `first_calls_1_to_31_stop_logit_greater_than_or_equal_continue_logit_else_forced_call32`. No post-hoc adjustment is applied, and the configuration records this explicitly: threshold search, temperature scaling, logit bias, a minimum-call floor, and a persistence rule are all `deployment_threshold_search` = no, `deployment_temperature_scaling` = no, `deployment_logit_bias` = no, `deployment_minimum_call_floor` = no, `deployment_persistence_rule` = no.

The evaluation checkpoint is `best_checkpoint.pt`, selected on validation only, and the reported policy quantities are produced by saved-trajectory replay. The checkpoints carry recorded digests (`best_checkpoint_sha256 6718ef89246ee92c...`), so the fitted artifacts are pinned.

G.4 Compute environment and run hygiene

Training and evaluation ran on two NVIDIA RTX 4090 accelerators on a single host. Before every long-running job the pipeline re-runs a preflight rather than trusting a remembered device: Require a complete nvidia-smi invocation to exit 0 with valid handles for the GPUs selected for the job.; Select GPUs that are healthy and sufficiently idle at launch time; do not rely on a fixed GPU ID or count.; Require torch.cuda.is_available() and a successful small CUDA allocation and matrix multiplication on every selected device.; Require SwanLab cloud mode and a new, non-conflicting target output directory.; Abort launch if any check fails; rerun the preflight before the next launch attempt.. The checks are recorded as required, with frequency `before_each_long_running_training_job`.

The manifest also preserves the recovery that followed a host migration, because a reader reproducing these runs should know that a hardware event intervened: A complete nvidia-smi invocation exited 0 and both migrated-host RTX 4090 GPUs had valid handles. PyTorch reported CUDA available and a small CUDA allocation and matrix multiplication passed twice. A single-GPU SB/shuffle training smoke wrote new diagnostic artifacts and completed with SwanLab cloud mode. The gate is recorded as `passed_after_migration` with scope noted as *This records the completed migration recovery check; it is not a standing guarantee of future GPU health or availability.*. Training

Table 13: Fitting and checkpoint-selection configuration. Values are the recorded run configuration for the OpenWebText readers; the same loop is used for the other tasks.

Item	Value
Optimizer steps (max / best)	`1000 / 920`
Batch size (train / validation)	`8 / 16`
Learning rate	`0.0003`
Weight decay	`0.0001`
Gradient clip norm	`1`
Mixed precision (bf16)	`yes`
Validation interval	`10`
Validation points audited	`101`
Selection metric	`validation_quality_protected_action_ce`
Selection direction	`min`
Tie break	`earlier_optimizer_step`
Best is last checkpoint	`no`
First / best / last selection loss	`0.7412646413 / 0.05799686536 / 0.06209835783`
Last minus best selection loss	`0.004101492465`

therefore has a documented interruption and a documented resume, rather than an unbroken provenance.

Each fitted reader is tracked with its own run identifier and pinned checkpoint digests (`swanlab_run_id` = `voihbsqkrk26vxwpt8pq2` for the reader summarised above), so a reported policy can be traced back to the exact weights that produced it. The registry audit itself performed no training, optimisation, or model inference; it only read frozen artifacts.

H Canonical validation counts

Before any paper number is produced, the recovered artifacts are checked against a canonical inventory. Every entry below is recorded with an expected value, an observed value, and, where the quantity is a rate, a tolerance; all checks pass. The table is generated directly from the registry, so it is the same list the analysis consumed.

The registry also fixes the counting units: CoLA spans 147,057 trajectories, 588,228 events, and 49,019 independent clusters over 3 generation seeds, while the Dream study spans 400 trajectories, 7,200 recorded events, 6,800 canonical step events, 2 conditions, and 200 independent clusters. Two related quantities deserve to be read side by side: CoLA contains 147,057 trajectories but a smaller number of clustered units, so trajectory counts and independent-sample counts are not interchangeable.

I Claims and their boundaries

Each claim examined in this project is recorded together with the boundary of what its evidence supports. Recording the boundary alongside the claim is what keeps the paper's wording honest when the evidence is partial, and it also documents one claim that the evidence does not support.

Statistical units are fixed per setting: CoLA uses *task::sample_id clustered across three generation noise seeds*, and the Dream inference uses *sample_id clustered across the two prompt conditions*. The registry itself is an audit artifact and records its own execution boundary: local-only offline artifact audit; no training, optimizer, or model inference; no SwanLab run; no source frontier artifact mutation.

J Data integrity and reproducibility limits

WMT14 official test was used previously for generator parity and older policy checks. The frozen E27 benchmark therefore is not globally untouched confirmation. Its 1,920-example independent reservation remains unopened because the strict fixed16 quality gate failed. No reservation data are used in this manuscript. XSum content hashing found five of 11,334 test rows overlapping monitor

Table 14: Canonical artifact checks. Expected versus observed values for the quantities the analysis consumes. Rates are compared at the recorded tolerance; counts are exact.

Quantity	Expected	Observed	Tolerance
cola trajectories	147,057	147,057	–
cola events	588,228	588,228	–
cola final accuracy	0.21592988	0.2159298775	1e-07
cola fixed-1 accuracy	0.179916631	0.179916631	1e-12
cola fixed-1 losses	5,494	5,494	–
cola fixed-1 mismatches	41,251	41,251	–
cola stability-2 mean block	2.511624744	2.511624744	1e-12
cola stability-2 gains	4	4	–
cola stability-2 mismatches	24	24	–
cola earliest-final-identity mean block	1.632428242	1.632428242	1e-12
cola recovery events	5,729	5,729	–
cola degradation events	221	221	–
cola recovery trajectories	5,494	5,494	–
cola degradation trajectories	216	216	–
dream trajectories	400	400	–
dream events	7,200	7,200	–
dream canonical step events	6,800	6,800	–
dream final accuracy	0.5275	0.5275	1e-12
dream fixed-8 accuracy	0.525	0.525	1e-12
dream fixed-4 losses	27	27	–
dream fixed-4 mismatches	51	51	–
dream sampled-event stability-2 mean step	8.47	8.47	1e-12
dream sampled-event stability-2 mismatches	4	4	–
dream earliest-final-identity mean step	4.7775	4.7775	1e-12
dream raw positive future gain events	460	460	–
dream raw negative future gain events	0	0	–

Table 15: Claims and the boundary of their evidence. The final row records a claim this study does not support; no halting or allocation claim is made at Dream's masked-token interface.

Claim	Boundary of the evidence
`C1_readiness_precedes_final_compute`	oracle/final-identity evidence is not an online stopping policy
`C2_native_latent_adds_incremental_readiness_information`	full students mix latent/hidden content with time, budget, logits, and process features
`C3_dream_adaptive_policy_beats_fixed_budget`	not supported in this study: a Dream policy that beats a fixed budget is tied by a fixed step-8 baseline, so no halting claim is made at this interface
`C4_early_correctness_can_degrade`	Dream D4 contains no accuracy-level negative future gain
`C5_real_online_speedup`	all current policy results are offline counterfactual replay

training/validation sources, including two exact source–reference groups. The primary analysis excludes these five rows; full-split results are a sensitivity analysis. The reader has not been retrained on deduplicated monitor data, so this correction does not establish strict training-data cleanliness.

The retained CoLA model-cache snapshot resolves to `9ee16d14bf4c869201daa08dd6911ebe9f92b9cd`, and Dream to `05334cb9faaf763692dcf9d8737c642be2b2a6ae`. These recovered hashes are not proof that historical downloads explicitly pinned revisions. Exact external CoLA prompt-template/scorer source commits are not recorded in the released summaries, and the legacy external files are absent after migration. Consequently, the recovered labels can be audited, but bitwise regeneration of prompt rendering and scoring is not yet established. Source-shard batch configurations also differ from the seed-66 merged directory's batch-size label; directory names must not be treated as frozen generation settings. Dream frontier summaries omit some extraction/scoring arguments, limiting exact reconstruction from the summaries alone.

The repository environment specifies Python 3.11, PyTorch 2.11.0 with CUDA 12.8, and Transformers 4.57.6. Training artifacts retain local metrics, best checkpoints, and cloud run identifiers;

hardware varies between recovered runs. No aggregate GPU-hour or integrated deployment-timing claim is made. Frozen protocol files distinguish training-time code digests from canonical runtime digests; the latter must not be silently substituted for the historical source state. An audit of fourteen E27 frozen runtime digests found ten matches and four mismatches in the current checkout; the accompanying audit names the affected files. Saved-artifact comparisons remain inspectable, but exact historical reruns require recovery of the recorded source versions. The manuscript's accompanying methods audit indexes the executable target builders, reader implementations, protocol files, and inspected summaries. These remaining provenance and deployment gaps qualify reproducibility and external validity without changing the reported saved-artifact comparisons.