TOWARDS VALUE-AWARE UNMASKING: LOCAL ACTION VALUE IN DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion language models (DLMs) must decide which masked position to commit next, yet existing endpoint predictors estimate trajectory-level success, and lookahead methods compare candidate futures. We ask whether the frozen representation already encodes value differences between alternative commitments. We introduce Local Action-Value Readout (LAVR), a framework that formalizes policy-relative action value for unmasking decisions, estimates it through matched forced-and-reference rollout branches with candidate centering, and reads it from frozen hidden states against output-and-trajectory controls. In an 800-prompt GSM8K study on Nemotron-Diffusion-3B, the hidden-state readout improves global state-value prediction over the output-statistics baseline ($R^2 = 0.4812$ vs. 0.3380) and supports selective generation. For local decisions, candidate-specific path log-likelihood (Path-LL) value replicates on a disjoint 600-document pool and improves pairwise ranking on an independent 400-document study across all six model–policy/target conditions. By contrast, task-correctness action gains are not reliably positive, and frozen transfer decreases concordance. This contrast shows that readable local continuation value does not automatically imply task-correctness value, making reward alignment central to task-aligned unmasking. Code and evaluation artifacts are available in an anonymous repository.

1 INTRODUCTION

Diffusion language models (DLMs) generate text by progressively resolving masked positions, enabling generation orders beyond left-to-right decoding (Sahoo et al., 2024; Lou et al., 2024). Their decoding policy must determine both what to predict and which positions to commit. Each commitment changes the context for subsequent predictions, making unmasking order a consequential control variable in decoding (Jazbec et al., 2026; Liu et al., 2026). Immediate confidence describes the current proposal, whereas its downstream value depends on the continuation it induces. If the current hidden representation already distinguishes how alternative candidate commitments affect expected continuation value, it could provide a candidate-level signal without explicitly evaluating each candidate's future continuation.

Two lines of work approach this problem at different levels. Future-aware decoding explores alternative continuations to compare decisions (Liu et al., 2026; Lee et al., 2025; Cui et al., 2026; Ye et al., 2026), while representation-based predictors estimate eventual success or trajectory quality from an unfinished generation's frozen hidden states (Kadavath et al., 2022; Chiu & Liu, 2026; Yadav et al., 2026). These approaches motivate a more specific representation question: whether the current frozen representation distinguishes the expected continuation values of alternative candidate commitments available from the same state. Candidates from the same state share the prompt, partial sequence, mask pattern, and decoding progress, so local evaluation must isolate how the choice of commitment changes expected continuation value. Selecting among commitments therefore requires candidate-level discrimination. The corresponding action-value target compares the expected reward after forcing a candidate with that of the reference continuation under the same policy. It is estimated using matched forced/reference rollouts.

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.

To address this question, we introduce Local Action-Value Readout (LAVR) to measure this policy-relative local quantity (Figure 1). A candidate commitment a in state s fixes the model's current most probable token at a masked position. Under continuation policy π, its advantage is $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$, where Q^π is its forced-action value and V^π the reference value. Matched forced/reference rollouts estimate this advantage. Candidates in the same state share additive state-level information, motivating within-set centering of targets and features. Frozen readouts test whether hidden representations predict candidate-specific value variation beyond output and trajectory controls. Rollouts provide offline supervision and evaluation labels. At inference, fitted readouts operate on current-state features.

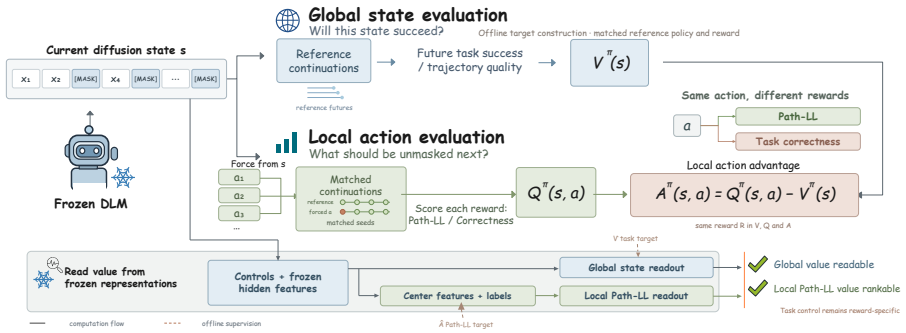


Figure 1: **LAVR overview.** Offline rollouts supervise current-state readouts for global trajectory selection and local candidate ranking.

Our evaluation moves from global state value to local path log-likelihood (Path-LL) ranking and task-correctness action evaluation, then compares Path-LL and task-correctness advantages for reward alignment. On GSM8K (Cobbe et al., 2021), frozen Nemotron representations improve global state-value R^2 from 0.3380 to 0.4812 and improve selective generation. We next test candidate-specific Path-LL value on the Masked Diffusion Language Model (MDLM) and Score Entropy Discrete Diffusion (SEDD). A frozen-protocol replication yields positive R^2 gains in all six conditions, ranging from $+0.0049$ to $+0.0207$, and an independent $K = 64$ study yields positive pairwise-concordance gains from $+0.0022$ to $+0.0094$. By contrast, task-correctness action gains have confidence intervals spanning zero on GSM8K and SVAMP (Patel et al., 2021), while frozen GSM8K transfer decreases concordance by 0.0356. These results separate state-level evaluation, local continuation ranking, and task-correctness action evaluation.

Our contributions are summarized as follows:

❶ **Local action-value formulation.** LAVR defines policy-relative commitment value, estimates it with matched forced/reference rollouts, and isolates within-state variation. We prove additive-state cancellation and noise cancellation in the fitted squared-error contrast under conditionally unbiased held-out labels.

❷ **A progression from state value to candidate ranking.** Global state-value readouts support trajectory selection. Frozen-protocol replication and independent ranking show that candidate-specific Path-LL action value is readable across models and policies.

❸ **Reward specificity of local action value.** The task-correctness and transfer experiments evaluate how continuation-value readability relates to task-correctness preferences, highlighting reward alignment as a consideration for future task-aligned unmasking policies.

2 Related Work

Future-aware diffusion decoding. Recent surveys cover decoding, inference efficiency, and controllability as active directions in diffusion language modeling (Li et al., 2026; Yu et al., 2025). Future-aware decoders obtain information in different ways. Path Likelihood Optimization (Liu et al., 2026) trains a value estimator for expected future Path-LL and uses it in sequential Monte Carlo search. Lookahead Unmasking (Lee et al., 2025) proposes paths and verifies their uncertainty before

importance-sampling among them. AdaLook (Cui et al., 2026) adapts rollout depth and branch expansion to intermediate candidate scores. Ripple-Pivot Search (Ye et al., 2026) uses lookahead to select a pivot position and token assignment. Trajectory-level speculative decoding (Pan et al., 2026) drafts denoising updates and verifies them in parallel. These inference procedures use path scoring, learned future-value prediction, exploration, or verification to guide decoding. These methods establish the utility of explicit future evaluation during decoding. LAVR measures reward-specific candidate value accessible in frozen current-state representations relative to output and trajectory statistics.

Future prediction from internal representations. Language-model hidden states can predict eventual correctness before generation is complete (Kadavath et al., 2022). In DLMs, correctness probing further shows how this information varies across denoising stages and layers, supporting early difficulty estimation and selective generation (Chiu & Liu, 2026); hidden-state ranking has also been used to improve reliability under input noise (Yadav et al., 2026). These results show that frozen representations contain useful information about an unfinished generation, but they primarily concern state-level outcomes. Related probing work emphasizes comparison against explicit controls (Hewitt & Liang, 2019), while frozen masked-model features can capture pairwise conditional dependence for parallel generation (Sharma et al., 2026). Our setting is finer-grained: candidates from the same state share the prompt, partial sequence, mask pattern, and decoding progress, so predictive state-level information need not distinguish which commitment is better. LAVR therefore centers within each candidate set and tests whether hidden representations explain candidate-specific action-value variation beyond output-and-trajectory controls.

Learning local decisions and control. A separate line of work uses future-oriented signals to change decoding behavior rather than only predict outcomes. Learned unmasking policies optimize generation order using confidence features and reinforcement learning (Jazbec et al., 2026). TraceLock (Sun et al., 2026) uses later token persistence as supervision for commitment decisions, while Self-Aware Scheduling (Xu et al., 2026) learns an order policy from a pathwise reward with the denoiser kept fixed. More broadly, Value Prediction Networks (Oh et al., 2017) learn return-predictive representations for planning. These methods show that future-derived signals can supervise local decisions, but they use those signals as part of controller learning or planning. LAVR isolates a prior measurement question: before learning a new controller, does the frozen DLM already expose reward-specific action value for alternative commitments in the current state? We test this under Path-LL and task-correctness rewards using matched rollout targets and frozen readouts.

3 LAVR: LOCAL ACTION-VALUE READOUT

LAVR starts from the policy-relative value of an unmasking commitment and estimates it using matched forced/reference rollouts. Local action evaluation depends on differences among candidates within the same state, so candidate-level Path-LL targets and features are centered before readout fitting. Frozen readouts then test whether current-state hidden representations predict these differences relative to the evaluated output and trajectory controls. Section 3.4 formalizes how centering and finite-rollout evaluation determine the interpretation of held-out predictive gains.

3.1 VALUE FORMULATION

A state s contains the prompt, current token assignments, mask, and decoding progress. A reference procedure π specifies continuation, including sampling and randomness conventions. A candidate action $a = (i, \hat{x}_i)$ commits the model's current most probable token $\hat{x}_i$ at masked position i. The token at each candidate position is fixed to the model's current argmax prediction, so candidates differ only in which masked position is committed.

For a reward functional R of the subsequent trajectory, define

$$V^\pi(s) = \mathbb{E}_\pi[R \mid s], \qquad Q^\pi(s,a) = \mathbb{E}_\pi[R \mid s, a], \qquad A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s). \tag{1}$$

Here Q^π is defined by forcing a and then continuing under π. Final-answer correctness defines the reward used for global state value and task-correctness action value. Finite-horizon Path-LL defines the reward used for local Path-LL action value. The common reference value cancels in pairwise action comparisons. Global state value V^π measures whole-state prospects and supports trajectory-level decisions such as selective generation, whereas advantage A^π measures how a commitment

changes those prospects relative to the reference continuation, matching the within-state comparison used by lookahead.

3.2 MATCHED ROLLOUT TARGETS

Task state value is the mean correctness of K completions, $\widehat{V}_{\text{task}}(s) = \frac{1}{K}\sum_{k=1}^{K} R_k$. Task advantage subtracts the reference reward from the forced-action reward, paired by rollout seed, with both branches completing all remaining positions.

We use two Path-LL contrasts. The per-token target includes the decision-point contribution. The downstream target $\widehat{A}_{\text{future}}$ removes that contribution and isolates continuation utility. Following the path-utility perspective of Liu et al. (2026), each committed token contributes its log probability under the full, untruncated model distribution p_θ. We use conditional-expectation scoring (stepwise Rao–Blackwellization), replacing the sampled token score by

$$b(s,i) = \sum_{v} \pi(v \mid s,i) \log p_\theta(v \mid s). \tag{2}$$

With horizon H, let ℓ_a be the forced-action score, $S_{Q,k}$ the sum of the next H forced-branch scores, $S_{V,k}$ the sum of $H+1$ reference scores, and $b^{(1)}_{V,k}$ the first reference score. Then

$$\widehat{A}_{\text{pertok}}(s,a) = \frac{1}{K}\sum_{k=1}^{K} \frac{\ell_a + S_{Q,k} - S_{V,k}}{H+1}, \tag{3}$$

$$\widehat{A}_{\text{future}}(s,a) = \frac{1}{K}\sum_{k=1}^{K} \frac{S_{Q,k} - (S_{V,k} - b^{(1)}_{V,k})}{H}. \tag{4}$$

Here K counts paired rollouts. Both branches use the same $H+1$ commitment budget and matched randomness, coupling order and token draws by seed, absolute position, and token identity (Appendix B).

3.3 CANDIDATE-CENTERED FROZEN READOUTS

With the backbone fixed, local readout evaluation compares candidates within the same state. We subtract each state's candidate mean from both features and labels, removing additive state-only components while preserving candidate ordering. For any candidate-level quantity $z(s,a)$ and candidate set $A(s)$, define

$$\widetilde{z}(s,a) = z(s,a) - \frac{1}{|\mathcal{A}(s)|}\sum_{a' \in \mathcal{A}(s)} z(s,a'). \tag{5}$$

Path-LL fitting and evaluation use this centered representation. The sampled candidate mean defines the zero point, isolating relative candidate variation while preserving within-state ordering.

The global readout uses pooled hidden features to predict state value, with its primary analysis comparing the hidden-state readout against the output-statistics baseline. Candidate Path-LL readouts compare control-only and control-plus-hidden models using hidden vectors at the proposed action position. Task-correctness readouts combine candidate-local hidden features with the pooled global state representation. Rollouts provide supervision and evaluation labels only. All inference-time features come from the current decoding state (Figure 2).

3.4 INTERPRETATION AND GUARANTEES

Centering removes every additive state-only component exactly, including $V^\pi(s)$: $\widetilde{A}^\pi = Q^\pi - \overline{Q}^\pi$, where $\overline{Q}^\pi$ is the candidate mean. State–candidate interactions remain part of the local problem. When the forced and reference branches follow their specified continuation procedures and use the matched horizon, the rollout difference is conditionally unbiased for the forced-action advantage. The branch continuation laws determine its expectation, while pairing changes variance.

We next relate held-out squared-error gains under finite-rollout labels to latent candidate-centered advantage. Let $Y = \widetilde{A}^\pi$, $\widehat{Y} = Y + \eta$, and let f, g be the fitted control and hidden-augmented

predictions. Let $\mathcal{T}$ contain training and selection, and $\mathcal{X}$ the current state, candidate set, evaluated action, and inputs. If $\mathbb{E}[\eta \mid \mathcal{X}, \mathcal{T}] = 0$ and the predictors use no evaluation rollout outcomes, then

$$\mathbb{E}[(\widehat{Y} - f)^2 - (\widehat{Y} - g)^2 \mid \mathcal{T}] = \mathbb{E}[(Y - f)^2 - (Y - g)^2 \mid \mathcal{T}]. \tag{6}$$

Under conditionally unbiased held-out labels, evaluation noise cancels from the expected hidden-versus-control squared-error difference. A positive population gain therefore indicates improved prediction of latent candidate-centered value under the evaluated fitted readouts. The mathematical appendix gives the formal conditions and proofs.

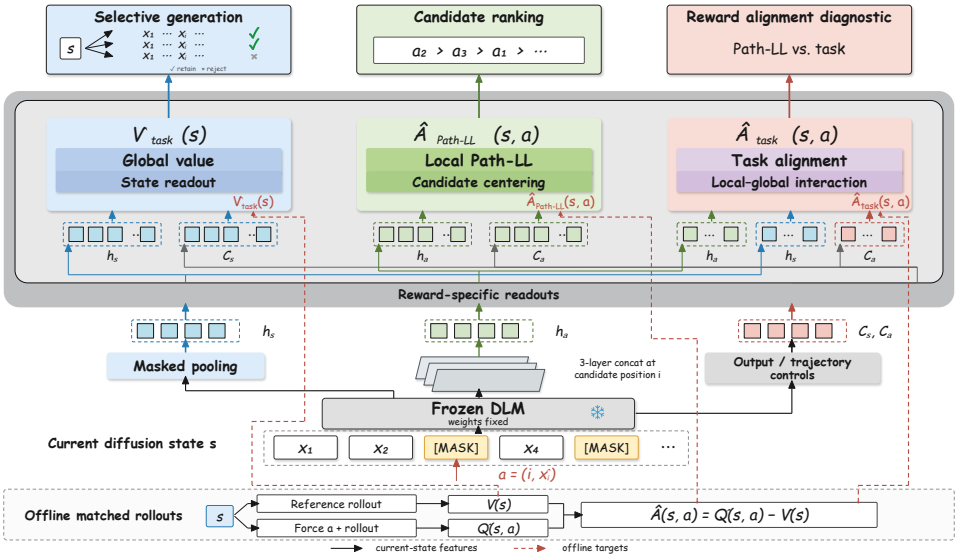


Figure 2: **Frozen-readout architecture.** Reward-specific readouts use frozen DLM states and controls, with matched rollouts for offline supervision.

4 Experimental Setup

We evaluate trajectory selection through global task-correctness value, within-state decisions through local Path-LL value and ranking, and task alignment through local task-correctness value. Table 1 summarizes the original settings; replication A and independent ranking study B extend the Path-LL evaluation.

Table 1: **Evaluation settings and original-study scale.** Path-LL conditions share 400 OpenWebText (OWT) documents, while replication A and study B use separate pools. Table 3 gives full counts.

Level	Backbone	Data	Target	Scale
State	Nemotron-3B	GSM8K	V_{task}	800 prompts
Local path	MDLM-169.6M	OWT	$A_{\text{Path-LL}}$	400 documents
Local path	SEDD-169.6M	OWT	$A_{\text{Path-LL}}$	400 documents
Task action	Nemotron-3B	GSM8K / SVAMP	A_{task}	950 prompts

4.1 Models, Data, and Evaluation Settings

Models. Nemotron-Labs-Diffusion-3B (Fu et al., 2026) uses a 160-position generation region with 32-position blocks, confidence-threshold commitments, and stochastic token sampling. MDLM-169.6M (Sahoo et al., 2024) and SEDD-169.6M (Lou et al., 2024) commit one token at a time with top-50 sampling at temperature 1.0. Both use ancestral ordering based on a fixed random key per position. MDLM also uses confidence ordering with a persistent per-position Gumbel key reused across the rollout.

Data and populations. Global state value uses 800 GSM8K prompts and 4,411 states. Task-correctness actions use disjoint GSM8K batches C and D (750 prompts total) and 200 SVAMP prompts. The original Path-LL study shares 400 OWT documents per model–policy condition, using 256-token windows with 64 observed prefix tokens. Replication A and study B use separate pools of 600 and 400 new documents, disjoint from the original and selection data.

Rollout protocol. Nemotron states are recorded at six decoding stages and task actions at four; correctness compares final extracted numbers within 10^{-6}. Path-LL records six stages from 10% to 60% of suffix commitments. Each state has three uniform and three confidence/instability-prioritized candidates, with $H = 16$ subsequent commits. The original study and replication A use $K = 24$ paired rollouts per candidate; study B uses $K = 64$. Task labels use eight completions paired for action contrasts. Representations, candidates, targets, and analysis choices were frozen before fresh data collection (Appendix B).

4.2 Readouts and metrics

Readouts. The 82-dimensional state baseline combines output statistics with a 64-dimensional full-log-probability projection; the 100-dimensional candidate baseline adds local confidence, margins, position, visible context, and trajectory features. State hidden features use validation-selected layer and pooling; Path-LL readouts concatenate local vectors from a selected layer and two flanking layers. Two-block ridge regression scales control and hidden features separately, with zero hidden weight in its validation grid. The task-correctness readout tests rank-4 interactions between candidate-local and pooled state projections fitted with principal component analysis (PCA), against control-only versions of the same architecture. Backbone weights remain frozen. The headline global comparison is hidden-state versus output-statistics readout; a nested hidden-plus-full-control analysis is secondary.

Metrics. Global prediction uses R^2 and area under the receiver operating characteristic curve (AUC) for states whose eight-rollout success rate exceeds 0.5; local value uses candidate-centered R^2. Ranking uses pairwise concordance C on identical within-state pairs with unequal rollout labels, giving half credit to prediction ties. Task-correctness actions use five-fold out-of-fold predictions and frozen C→D transfer, fitted on C and evaluated on disjoint D. Splits are grouped by problem or document: 480/120/200 prompts for state value; 50 repeated 240/60/100-document splits for original Path-LL; fixed 360/90/150 and 240/60/100 splits for replication A and study B.

4.3 Statistical inference

State-value and selective-generation intervals resample prompts, preserving all their states. Stage-specific fits retain the selected representation and repartition the available prompts at each stage. Original Path-LL intervals use the Nadeau–Bengio correction (Nadeau & Bengio, 2003) for overlapping splits, with variance factor $1/50 + 100/240$. Benjamini–Hochberg adjustment (Benjamini & Hochberg, 1995) covers four prespecified hidden-feature variants within each model–policy condition, target, and metric. This analysis conditions on the selected representation.

Replication A and study B use 10,000 document-cluster percentile bootstrap draws on their held-out test documents. Task-correctness action intervals use 10,000 prompt-cluster draws with fitted predictions fixed. Replication A and study B intervals are pointwise. Study B prespecifies downstream concordance as primary and per-token concordance as secondary. Appendix C details resampling, selection, and pair handling.

5 Results

Table 2 summarizes the progression from global state value and trajectory selection to local Path-LL prediction and ranking, followed by task-correctness action evaluation.

5.1 Global state value is readable and supports selection

Frozen hidden-state readouts improve state-value prediction throughout decoding. The primary comparison evaluates a hidden-state readout against output-statistics controls. The layer-18 masked-position readout raises R^2 from 0.3380 to 0.4812, with $\Delta R^2 = +0.1433$ $[+0.0847, +0.2042]$, and

AUC from 0.8229 to 0.8896 (Table 2). AUC measures discrimination of states with a majority of successful completion rollouts. The hidden R^2 reaches approximately 48.5% of the estimated state-label noise ceiling (Appendix D).

Table 2: **Main results.** Global rows compare hidden-state with output-statistics readouts; local rows compare control-plus-hidden with control-only. Gains have 95% confidence intervals (CIs). Replication A value and study B ranking use per-token (PT) and downstream (DS) targets. Values are rounded.

Setting	Metric	Control	Hidden	Gain [95% CI]
Global state value: Nemotron / GSM8K				
GSM8K	R^2	0.3380	0.4812	+0.1433 [0.0847, 0.2042]
GSM8K	AUC	0.8229	0.8896	+0.0666 [0.0378, 0.0982]
Selective generation: Nemotron / GSM8K				
70% coverage	Correct (%)	74.87	77.23	+2.36 [0.25, 4.23] pp
50% coverage	Correct (%)	80.78	86.91	+6.12 [2.47, 9.26] pp
30% coverage	Correct (%)	82.07	90.01	+7.94 [2.99, 13.68] pp
Coverage area	Area	0.7722	0.8186	+0.0464 [0.0171, 0.0812]
Local Path-LL: replication A (600 docs, $K = 24$), study B (400 docs, $K = 64$)				
Model / policy	Target	A: ΔR^2 [95% CI]		B: ΔC [95% CI]
MDLM anc.	PT	+0.0049 [0.0022, 0.0076]		+0.0031 [0.0010, 0.0052]
MDLM anc.	DS	+0.0088 [0.0045, 0.0131]		+0.0049 [0.0022, 0.0076]
MDLM conf.	PT	+0.0089 [0.0056, 0.0122]		+0.0030 [0.0006, 0.0054]
MDLM conf.	DS	+0.0168 [0.0112, 0.0224]		+0.0062 [0.0031, 0.0093]
SEDD anc.	PT	+0.0097 [0.0061, 0.0133]		+0.0022 [0.0001, 0.0043]
SEDD anc.	DS	+0.0207 [0.0149, 0.0265]		+0.0094 [0.0060, 0.0128]
Task-correctness action concordance: Nemotron				
Setting	Metric	Control	Hidden	Gain [95% CI]
GSM8K cross-fit	C	0.4868	0.5010	+0.0143 [-0.0091, 0.0369]
SVAMP cross-fit	C	0.4454	0.4587	+0.0133 [-0.0369, 0.0654]
GSM8K C→D	C	0.5138	0.4783	-0.0356 [-0.0567, -0.0151]

With the representation fixed, R^2 gains have positive intervals at all six stages, whereas AUC gains have positive intervals at five, starting at approximately 10% progress (Figure 3; Appendix D).

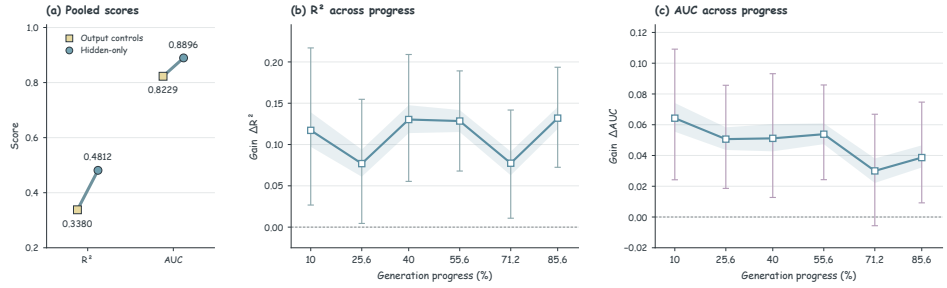


Figure 3: **Global state value remains readable across decoding.** The left panel compares pooled readouts. The remaining panels show hidden-minus-control gains with prompt-cluster 95% CIs in the 800-prompt GSM8K study.

A separate nested hidden-plus-control comparison yields an additional R^2 gain of $+0.1317$ over the full controls and $\Delta\text{AUC} = +0.0655$. Gaussian features and training-label permutation remove the benefit (Appendix D). A secondary within-prompt centering analysis leaves essentially no hidden-state gain, indicating that the global state-value signal primarily tracks between-prompt variation (Appendix D).

We evaluate trajectory selection through offline selective generation, applying the global state-value readout to choose which in-progress trajectories to retain. Mean rollout correctness rises by $+2.36$,

+6.12, and +7.94 percentage points at 70%, 50%, and 30% retained-state coverage, respectively. All three prompt-cluster 95% intervals exclude zero (Figure 4). The accuracy–coverage area increases by $+0.0464$ $[0.0171, 0.0812]$. Appendix E provides absolute scores and the separate computation-accounting scenario. Global state value guides trajectory selection by identifying states with better continuation prospects.

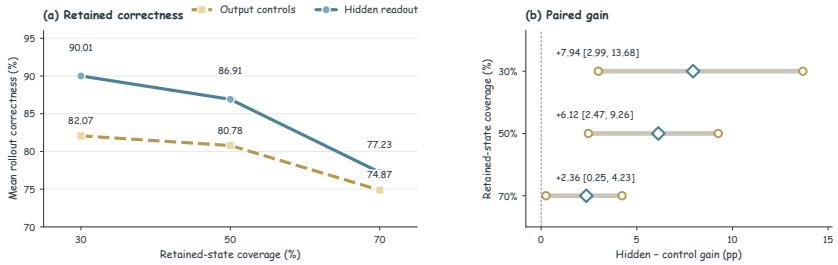


Figure 4: **State-value ranking improves retained correctness on held-out states.** The right panel reports paired gains with prompt-cluster 95% CIs.

5.2 Local Path-LL value replicates and supports ranking

Candidate-specific Path-LL value reproduces on a new document pool and improves within-state ranking. Global state value ranks states. Local candidate selection requires within-state candidate ranking. Candidate-centered Path-LL evaluation isolates this distinction. Frozen-protocol replication A reproduces the effects in all six model–policy/target conditions on 600 new documents at $K = 24$ (Figure 5a; Table 2). All six pointwise 95% intervals exclude zero, retaining approximately 85–89% of the original effect magnitudes under representation and analysis choices fixed before collection. All 18 replication-A null-control intervals include zero (Table 10).

On a separate 400-document pool at $K = 64$, study B yields pairwise-ranking gains of $+0.0022$ to $+0.0094$ with pointwise intervals positive in all six model–policy/target conditions (Figure 5b). These comprise three prespecified downstream primary endpoints and three per-token secondary endpoints. Downstream gains exceed per-token gains in every model–policy condition.

Downstream value gains exceed per-token gains in every model–policy condition of replication A and the original study (Table 2; Appendix F.2). The downstream target excludes the immediate decision-point score, so the measured gain reflects prediction of subsequent continuation utility. Original corrected effects and candidate-identity controls agree (Tables 6–7). Full replication and ranking comparisons appear in Tables 8–9. Finite-state validation agrees with exact advantages at the Monte Carlo error scale and reduces estimator variance by 4.13–9.88× (median 6.15×; Appendix F.1).

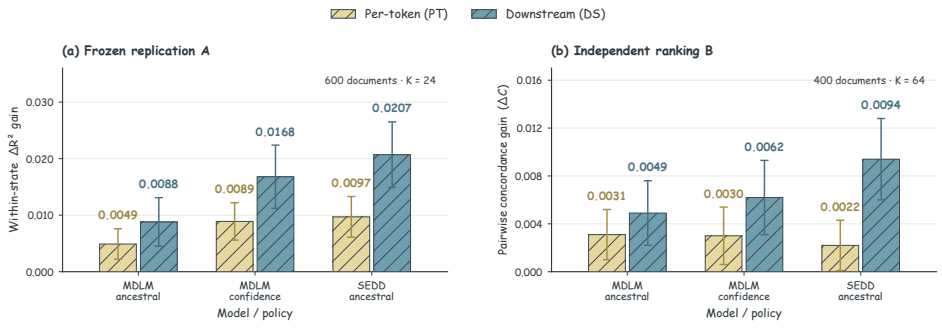


Figure 5: **Replication A and study B Path-LL results.** Both studies show positive gains across all six model–policy/target conditions. Points and bars give estimates and pointwise document-cluster 95% CIs.

The fresh studies use disjoint document pools and fixed document-level splits: 360/90/150 for replication A and 240/60/100 for study B. Each document contributes six decoding stages with six candidates per state. Held-out intervals resample documents as clusters, keeping their stages and candidate pairs together. The two studies thus test frozen-protocol value replication and ranking on independent populations; their differences reflect both population and rollout budget (Appendix F.2).

5.3 Local action-value readability is reward-specific

Task-correctness readouts yield cross-fitted concordance gains of $+0.0143$ ***on GSM8K C+D and*** $+0.0133$ ***on SVAMP, with both 95% intervals spanning zero.*** The rank-4 task-correctness readout tests whether readable local information supports the task-correctness reward (Table 2). Removing its global-state interaction changes concordance by -0.0022 $[-0.0220, 0.0174]$ on GSM8K C+D and $+0.0109$ $[-0.0262, 0.0499]$ on SVAMP (Appendix F.3).

The frozen task-correctness readout yields a negative concordance increment over its control baseline under C→D transfer. The representation layer, projection, normalization, and scoring weights selected on batch C are frozen and applied to disjoint GSM8K batch D. The increment is -0.0356 $[-0.0567, -0.0151]$. Replication A and study B reproduce Path-LL ordering gains. Task-correctness gain intervals span zero under the evaluated readouts. Pearson correlations between Path-LL and task-correctness candidate advantages are $+0.0493$ $[-0.0222, 0.1208]$ on GSM8K C and $+0.2246$ $[0.1391, 0.3094]$ on SVAMP (prompt-cluster 95% intervals), showing that reward alignment varies across the evaluated populations (Appendix F). Two additional GSM8K diagnostic batches reported in Appendix F yield correlations of $+0.0957$ $[0.0248, 0.1727]$ and -0.0174 $[-0.1059, 0.0667]$, extending the reward-alignment diagnostic across four populations.

6 Discussion

Global state prediction and local action selection are different decision problems. Endpoint readouts assess trajectory prospects (Kadavath et al., 2022; Chiu & Liu, 2026), while candidate centering isolates differences among commitments within a state. The local hidden-versus-control readout comparison (Hewitt & Liang, 2019) detects readable candidate-specific Path-LL action value beyond output and trajectory features. Global state value supports selective generation by choosing promising trajectories.

Local value is useful with respect to the reward it predicts. Path-LL gains reproduce and support candidate ranking, whereas task-correctness cross-fit intervals include zero and the frozen C→D transfer is negative. Path-LL/task-correctness advantage correlations vary across the evaluated populations, indicating different local preferences. Lookahead methods evaluate future paths (Liu et al., 2026; Lee et al., 2025), while learned policies and value-based planning optimize an explicit objective (Jazbec et al., 2026; Oh et al., 2017). The observed differences in local preferences motivate checking reward alignment when continuation-value readouts are used for task decisions.

The present evidence characterizes frozen-readout prediction. Closed-loop intervention is a subsequent empirical question. Global and local gains should be interpreted within their respective backbones and populations. Study B changes both the document population and K relative to replication A. The readouts measure predictive information. Future closed-loop experiments can evaluate task improvement under intervention. Additional scope and design limitations are summarized in Appendix A.

7 Conclusion

LAVR formulates and evaluates policy-relative action value for diffusion unmasking decisions. Global state value supports trajectory selection, while candidate-specific Path-LL value replicates on a new document pool and supports within-state ranking. Task-correctness experiments distinguish continuation-value ranking from task-aligned action evaluation, showing that readable local value depends on the reward being predicted. More broadly, LAVR provides a framework for asking not only whether future value is encoded in a frozen DLM, but also which decision-level value that representation makes accessible for decoding.

ARTIFICIAL INTELLIGENCE USE STATEMENT

In this work, generative AI tools were not used for any research tasks requiring disclosure, including developing theoretical or conceptual frameworks, formulating or proving mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, generating or modifying experimental data, cleaning or reformatting datasets, performing qualitative or thematic analysis, interpreting results, or translating research materials. Survey- and interview-related tasks are not applicable to this work.

Generative AI tools were used only for manuscript-level assistance, including editing text for clarity and readability, organizing author-provided experimental results for presentation in the manuscript, and checking textual consistency such as abbreviation definitions and usage. The tools did not generate new experimental findings or scientific interpretations. The authors reviewed all AI-assisted edits and take responsibility for the final content of the work, including all text and presentation choices made with the aid of generative AI.

ETHICS STATEMENT

This work studies value information in frozen diffusion language model representations and evaluates it using public benchmarks, public text corpora, and released model checkpoints. The experiments do not involve human-subject data collection, user interaction, or deployment of an autonomous decision-making system. LAVR is evaluated as an offline measurement framework: matched rollouts provide supervision and evaluation targets, while the reported readouts operate on frozen model representations. As with other methods for value-aware decoding, the broader impact of using such signals depends on the reward being optimized and the deployment context. Our results specifically highlight this dependence by showing that value readability is reward-specific, motivating explicit consideration of reward alignment when such signals are used to guide future decoding policies.

REPRODUCIBILITY STATEMENT

Section 3 defines LAVR's estimands and readouts, and Section 4 specifies the experimental design and grouped evaluation. Appendix B documents decoding, collection, and feature specifications. Appendix C describes readout fitting, uncertainty, and selection scope. Appendix D reports extended state and temporal evidence, Appendix E reports selective computation from global state value, and Appendix F reports local Path-LL and task-utility experiments. Appendix G provides mathematical guarantees, with supplementary mathematical details in Appendix H. The supplementary source package contains figure data, saved result summaries, editable figure sources, and executable checks of repeated-split effects and aggregate comparisons. Code and evaluation artifacts are also available in the anonymous repository, providing the released implementation and evaluation resources associated with the reported experiments. These materials reproduce the reported numerical summaries and visualizations; model checkpoints are specified in Appendix B.1, decoding and collection in Appendix B.2, and implementation and computational resources in Appendix B.4.

REFERENCES

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 01 1995. ISSN 0035-9246. doi: 10.1111/j.2517-6161.1995.tb02031.x. URL https://doi.org/10.1111/j.2517-6161.1995.tb02031.x.

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/.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John

Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Yingqian Cui, Wei Deng, Lantao Mei, Hang Li, Charu C. Aggarwal, Hui Liu, and Yue Xing. Adaptive multi-step lookahead decoding for diffusion language models, 2026. URL https://arxiv.org/abs/2607.15655.

Yonggan Fu, Lexington Whalen, Abhinav Garg, Chengyue Wu, Maksim Khadkevich, Nicolai Oswald, Enze Xie, Daniel Egert, Sharath Turuvekere Sreenivas, Shizhe Diao, Chenhan Yu, Ye Yu, Weijia Chen, Sajad Norouzi, Jingyu Liu, Shiyi Lan, Ligeng Zhu, Jin Wang, Jindong Jiang, Morteza Mardani, Mehran Maghoumi, Song Han, Ante Jukić, Nima Tajbakhsh, Jan Kautz, and Pavlo Molchanov. Nemotron-labs-diffusion: A tri-mode language model unifying autoregressive, diffusion, and self-speculation decoding. Technical report, 2026. URL https://arxiv.org/abs/2607.05722.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. 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. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Metod Jazbec, Theo X. Olausson, Louis Béthune, Pierre Ablin, Michael Kirchhof, João Monteiro, Victor Turrisi, Jason Ramapuram, and Marco Cuturi. Learning unmasking policies for diffusion language models, 2026. URL https://arxiv.org/abs/2512.09106.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know, 2022. URL https://arxiv.org/abs/2207.05221.

Sanghyun Lee, Seungryong Kim, Jongho Park, and Dongmin Park. Lookahead Unmasking Elicits Accurate Decoding in Diffusion Language Models. arXiv preprint arXiv:2511.05563, 2025. URL https://arxiv.org/abs/2511.05563.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models, 2026. URL https://arxiv.org/abs/2508.10875.

Xuejie Liu, Yap Vit Chun, Yitao Liang, and Anji Liu. Lookahead path likelihood optimization for diffusion llms, 2026. URL https://arxiv.org/abs/2602.03496.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Claude Nadeau and Yoshua Bengio. Inference for the Generalization Error. *Machine Learning*, 52: 239–281, 2003. doi: 10.1023/A:1024068626366. URL https://mlanthology.org/mlj/2003/nadeau2003mlj-inference/.

Junhyuk Oh, Satinder Singh, and Honglak Lee. Value prediction network. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/ffbd6cbb019a1413183c8d08f2929307-Paper.pdf.

Tianxiang Pan, Baitao Gong, Mo Guang, Hongwei Yong, Tianpeng Jiang, Yaqian Li, Zheng Cao, and Kaiwen Long. Trajectory-level speculative decoding for diffusion language models, 2026. URL https://arxiv.org/abs/2608.27514.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168/.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184. Curran Associates, Inc., 2024. doi: 10.52202/079017-4135. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/eb0b13cc515724ab8015bc978fdde0ad-Paper-Conference.pdf.

Jai Sharma, Yifan Wang, and Bryan Li. Neural estimation of pairwise mutual information in masked discrete sequence models, 2026. URL https://arxiv.org/abs/2605.20187.

Bohang Sun, Max Zhu, Francesco Caso, Jindong Gu, Junchi Yu, Philip Torr, Pietro Liò, and Jialin Yu. The path matters: Learning a token-commitment policy for diffusion language models, 2026. URL https://arxiv.org/abs/2605.24697.

Jiawei Xu, Minghui Liu, Aakriti Agrawal, Yifan Chen, and Furong Huang. Scheduling thoughts: Learning the order of thought in diffusion language models, 2026. URL https://arxiv.org/abs/2606.23567.

Saurabh Yadav, Badri Narayana Patro, and Vijay Srinivas Agneeswaran. Unsure but certain: Uncovering the representation-confidence gap in diffusion language models, 2026. URL https://arxiv.org/abs/2608.08791.

Yushi Ye, Xu Chen, Haoyun Jiang, Jinsong Lan, Haihong Tang, Xiangtao Li, Mingming Gong, Ivor Tsang, Yanfeng Wang, and Jiangchao Yao. Ripple-pivot search: Active parallel decoding for diffusion large language models, 2026. URL https://arxiv.org/abs/2608.11742.

Runpeng Yu, Qi Li, and Xinchao Wang. Discrete diffusion in large language and multimodal models: A survey, 2025. URL https://arxiv.org/abs/2506.13759.

CONTENTS OF APPENDIX

A Limitations **14**

B Decoding, collection, and feature specifications **14**

B.1 Backbones and task populations 14

B.2 Reference branches and randomness 15

B.3 Feature definitions 16

B.4 Implementation and computational resources 16

C Readout fitting, uncertainty, and selection scope **16**

C.1 Linear readouts and centering 16

C.2 Dependence-corrected inference 17

C.3 Task-action fitting and pairwise evaluation 17

D Extended state and temporal evidence **18**

E Selective computation from global state value **18**

F Local Path-LL and task utility **19**

F.1 Finite-state validation of the rollout estimator 19

F.2 Path-LL replication and independent $K = 64$ ranking 20

F.3 Task-action state interactions 21

G Mathematical guarantees for local action-value measurement **23**

G.1 Setup and exact within-state cancellation 23

G.2 Unbiased matched-rollout estimation 23

G.3 Coupling, shared references, and precision 24

G.4 Conditional-expectation scoring 24

G.5 Population predictive accessibility 25

G.6 Linear residual characterization and fitted ridge 25

G.7 Unbiased evaluation noise preserves the fitted risk difference 26

G.8 Pairwise ordering 26

G.9 Finite-rollout ranking, margins, and eligibility 27

H Supplementary mathematical details **28**

H.1 Sequential scoring and regularized parameterization 28

H.2 Eligibility and implemented weighting 28

A LIMITATIONS

LAVR is designed to characterize reward-specific value information accessible from frozen DLM representations, and the present experiments establish this at both global state and local candidate levels under the evaluated settings. The resulting claims are therefore scoped to the studied backbones, decoding policies, populations, and reward definitions rather than assuming that the same readout transfers unchanged across models or objectives. In particular, the consistent Path-LL results across MDLM and SEDD establish readable local continuation value, while the task-correctness and transfer experiments show that such readability is reward-specific rather than universally interchangeable across objectives. Replication A and study B further provide independent evidence across disjoint document pools, although study B changes both the population and rollout budget K, so differences between the two studies are not attributed to K alone. More broadly, LAVR evaluates the predictive information already present in frozen representations; using these signals for closed-loop commitment selection is a natural next step for testing how representation-level value translates into task-level decoding improvements.

B DECODING, COLLECTION, AND FEATURE SPECIFICATIONS

Table 3: **Original-study populations and rollout counts, before splitting.** The three Path-LL arms share 400 OWT documents. GSM8K C/D are disjoint; E denotes SVAMP.

Backbone	Continuation policy	Dataset	Quantity measured	Prompts/ docs	States	Candidates	Rollouts
Nemotron-3B	Block	GSM8K	V_{task}	800	4,411	—	8
MDLM-169.6M	Ancestral	OWT	$A_{\text{Path-LL}}$	400	2,400	14,400	24
MDLM-169.6M	Confidence	OWT	$A_{\text{Path-LL}}$	400	2,400	14,400	24
SEDD-169.6M	Ancestral	OWT	$A_{\text{Path-LL}}$	400	2,400	14,400	24
Nemotron-3B	Block	GSM8K C	A_{task}	180	720	4,320	8
Nemotron-3B	Block	GSM8K D	A_{task}	570	2,280	13,680	8
Nemotron-3B	Block	SVAMP E	A_{task}	200	800	4,800	8

B.1 BACKBONES AND TASK POPULATIONS

The MDLM backbone is the released `kuleshov-group/mdlm-owt` checkpoint (Sahoo et al., 2024); SEDD uses `louaaron/sedd-small`, with an absorbing graph (Lou et al., 2024). Both have approximately 169.6 million parameters, 12 transformer blocks, hidden width 768, and Generative Pretrained Transformer 2 (GPT-2) tokenization augmented with a mask token. Their transformer designs are closely related, but their training objectives and time conditioning differ. The SEDD wrapper converts its score-ratio outputs into the conditional token distribution over non-mask symbols at masked positions. The released model identifiers specify the backbones used in the experiments.

Nemotron uses `nvidia/Nemotron-Labs-Diffusion-3B`, pinned to checkpoint revision `0d51902da1f8869f83413ce642fab402fa5641e0` (Fu et al., 2026). It has 26 transformer layers, hidden width 3,072, vocabulary size 131,072, and mask token identifier 100. Hidden states are captured from transformer-layer outputs. The state-value study records all 26 layers; the task-action study stores every second layer, giving 13 stored indices. Layer numbers for task-action readouts below index that stored array. Models are evaluated with frozen weights and no gradient propagation through rollouts.

Original-study OpenWebText windows are sampled from the public `stas/openwebtext-10k` collection. The window builder shuffles documents with seed 0, filters for sufficient length, and selects a 256-token window from each qualifying document; its first 64 tokens form the observed prefix. Each original arm uses the same 400 documents with independent model/policy trajectories. Readout test documents are held out from readout training. Overlap of this in-distribution corpus with backbone pretraining is unmeasured.

The 800 state-value prompts are the first 800 indexed problems in the GSM8K test partition, collected in two disjoint batches of 400. They supply 4,411 available states. Task-action collection instead

enriches for decisions with nonconstant outcomes: an initial eight-rollout screen identifies problems with both successful and unsuccessful outcomes, and draws a mixture from that set and from the remaining screened problems. Batch C contains 180 problems from a 260-problem screen beginning at index 440 (72 mixed-outcome and 108 other problems). Batch D contains 570 problems from a 600-problem screen beginning at index 700. SVAMP contains 200 problems from a 300-problem screen. D and SVAMP request up to 70% mixed-outcome problems, subject to availability. Thus these task-action samples characterize the specified proposal populations. C and D use disjoint problem sets. Some state-value prompts overlap the task-action collections. The overlapping state-value and task-action collections are analyzed as separate studies.

Table 4: Original-study decoding and label-collection settings. Progress is measured by positions committed. Each candidate action forces the current top-probability token at one position.

Setting	MDLM / SEDD Path-LL	Nemotron task utility
Generation region	192 suffix tokens	160 appended tokens
Policy granularity	One token per step	Blocks of 32; multiple commits permitted
Token sampling	Top 50; temperature 1.0	Top 50; temperature 0.2
Position selection	Uniform order; or log-confidence/2 plus persistent position Gumbels	Confidence ≥ 0.9 in active block; otherwise maximum-confidence fallback
Recorded fractions	0.10, 0.20, 0.30, 0.40, 0.50, 0.60	State: see Table 5; action: 0.15, 0.35, 0.55, 0.75
Candidates per state	3 uniform + 3 prioritized	3 uniform + 3 prioritized for C/D/E
Rollout replicates	24 paired replicates	8; paired for action labels
Horizon	Forced action + 16 continuation commits versus 17 reference commits	Completion of all remaining positions
Reward	Mean full-distribution log probability; conditional expected stochastic-step score	Extract final number; absolute error below 10^{-6}

B.2 Reference branches and randomness

For each recorded state, a reference branch starts from the unchanged partial sequence. A candidate branch sets one masked position to the current argmax token, then uses the same continuation procedure. Seeds couple token noise by absolute position and token identity, so a position shared by two branches receives the same random variate even when its top-50 token ranks differ. Ancestral order uses a fixed random key per position. The confidence arm retains its position Gumbel key across the rollout while updating log confidence from the evolving sequence. This persistent-randomness procedure defines its reference values.

The two Path-LL estimands in Equations 3–4 compare equal numbers of total commitments. The per-token target includes the deterministic contribution of the forced proposal. The downstream target deletes the reference branch's own first contribution as well as the forced proposal's contribution. The latter is consequently a contrast of expected continuation utility after two different decision-point choices. At the latest Path-LL checkpoint roughly 77 suffix positions remain, exceeding the 17-position matched horizon. All six primary conditions therefore use the same fixed horizon.

For stochastic commitments, sampling from the truncated policy and scoring under the full model distribution are separate operations. Equation 2 averages the current log-probability contribution over the sampling distribution, while the sampled token determines subsequent states. The implementation also retains realized-token scores for diagnostics. The reported targets use the conditional expected contributions throughout.

Nemotron appends two newline characters followed by the exact instruction: "End your reply with the final numeric answer alone on the last line." The question and suffix form the user message passed to the tokenizer's chat template. The scorer extracts the final numeral, removes thousands separators, and compares it with the gold number. The procedure supplies a reproducible task reward based on

final-answer correctness. Reasoning-step validity lies outside this reward definition. Both reference and forced-action branches fill the entire generation region under the same block rule.

B.3 Feature definitions

The state-value control vector has 82 components. Its 18 scalar features comprise normalized progress and mask counts; mean, standard deviation, minimum, maximum, and quartiles of top-token probability over masked positions; mean and spread of the top-50 entropy contribution and top-two log margin; mean and spread of top log probability; mean second-token probability; and observed-region/progress summaries. A fixed Gaussian projection of the mean full log-probability vector adds 64 components. Entropy here refers to the summed contribution on the retained top-50 support. Some progress/count summaries are redundant; they are retained as collected. Hidden features average a layer's vectors over either all positions or currently masked positions. The primary state representation is the masked-position mean at recorded layer 18.

The 100-dimensional candidate control vector contains eight output scalars (top probability and log probability, top-50 entropy contribution, probability and log-probability margins, second probability, maximum logit, and spread of top logits); 12 progress, position, locality, and trajectory features; and 80 distributional features. Locality includes the observed-token fraction in a radius-8 window and distances to visible context. Trajectory features include a top-support temporal divergence, changes in confidence, proposal flips, and persistence. Distributional features are 16 sorted top log probabilities and a fixed 64-dimensional full-log-probability projection. The projection seed is 20260819.

For the saved primary Path-LL arms, the flip and persistence counters follow the collection-time convention before incorporation of the current transition. The analyses measure the hidden increment over those recorded controls. Recollection with current-transition-aligned counters would test the stronger control configuration. State-value features use their separate state-level feature set. Task C/D/E were collected with the current-transition convention.

B.4 Implementation and computational resources

Collection runs locally through PyTorch on CUDA devices. Nemotron uses the Transformers loader with the pinned checkpoint revision, bfloat16 (BF16) weights, evaluation mode, and forward hooks for hidden states. MDLM and SEDD keep weights in 32-bit floating point (FP32) and use BF16 autocasting in transformer blocks. The documented Path-LL environment specifies PyTorch 2.7.0+cu126 and Transformers 4.53.2; the Nemotron loader requires Transformers 5.0.0 or later.

The state-value collection launcher exposes one graphics processing unit (GPU) per job. The two 400-prompt batches record 8,887 and 9,024 seconds of elapsed collection time, totaling 4.98 single-GPU job-hours. These timings cover the primary 800-prompt state study. The task-action launch record assigns separate single-GPU workers to GSM8K D and uses automatic model sharding over two devices for SVAMP. Rollout microbatches are reduced after memory-allocation failures.

The ridge implementation uses CUDA when available and otherwise falls back to NumPy; saved task-readout settings specify CUDA. Original-study bootstrap resampling uses NumPy on the central processing unit (CPU). The exact solver backend for each ridge fit, per-run GPU model and memory, peak memory, complete software versions, and replication A/study B resource allocations and collection times are not retained in the available run metadata. Sampling and analysis seeds are specified in Appendices B.2 and C.

C Readout fitting, uncertainty, and selection scope

C.1 Linear readouts and centering

Ridge fits standardize each feature using training-set means and standard deviations, treating scales below 10^{-8} as one. Regularization is selected from 34 logarithmically spaced values between 10^{-2} and 10^{9} by validation R^2. Two-block fits concatenate standardized controls C and scaled hidden features γH, with

$$\gamma \in \{0, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3\}.$$

The zero value nests the control-only model. The state-value split uses seed 0 and 480/120/200 prompts, producing 2,655/658/1,098 states. The selected ridge penalties are approximately 21.54 for controls and 4,641.59 for hidden features. AUC is computed from the ridge scores against $\mathbb{1}[\widehat{V}_{\text{task}} > 0.5]$, corresponding to at least five successes among eight rollouts.

The candidate Path-LL objective uses Equation 5 on both labels and features. Centering maps a global state vector to zero, so candidate discrimination is carried by candidate-varying features. Centering is performed within complete states that never cross readout partitions. Model fitting uses training labels. Test labels define the centered evaluation target, and candidate feature centering uses the six evaluated current-state vectors.

Original Path-LL representation selection uses document split seed 99. The chosen center layer is 9 for MDLM ancestral, 8 for MDLM confidence, and 9 for SEDD ancestral, for both reward definitions. The three-layer blocks are respectively $\{6, 9, 12\}$, $\{5, 8, 11\}$, and $\{6, 9, 12\}$. These choices are held fixed across split seeds 0–49, each with 240 training, 60 validation, and 100 test documents. Representation selection is fixed from split seed 99 and reused across the repeated outer splits on the same documents. Inference conditions on that selected configuration.

C.2 Dependence-corrected inference

For repeated-split effects $d_1, \ldots, d_J$, the Nadeau–Bengio (NB) corrected standard error (SE) is

$$\mathrm{SE}_{\mathrm{NB}}(\bar{d}) = \sqrt{\left(\frac{1}{J} + \frac{n_{\text{test}}}{n_{\text{train}}}\right) \frac{1}{J-1} \sum_{j=1}^{J} (d_j - \bar{d})^2}, \tag{7}$$

with t_{J-1} critical values (Nadeau & Bengio, 2003). At $J = 50$, $n_{\text{train}} = 240$, and $n_{\text{test}} = 100$, this standard error is about 4.67 times the naive independent-split standard error. Table 6 reports the corrected intervals. Benjamini–Hochberg adjustment (Benjamini & Hochberg, 1995) uses the corrected p-values across four signal feature variants separately for each arm, target, and metric. The variants are local hidden features, three-layer local features, local plus global features, and a projected interaction block. The six headline intervals quantify pointwise uncertainty for the specified effects with dependence correction.

The state-value headline intervals use 2,000 prompt-cluster bootstrap replicates, with random seed 1. Stage-specific and selective-ranking intervals use 1,500 replicates; the stage seeds are $2 + r$ for stage index r, selective operating points use seed 9, and the accuracy–coverage area uses seed 11. Each draw resamples prompts and includes all their available states. Intervals condition on the fitted readouts. The six stage fits retain the main layer/pooling configuration but split the currently available prompts anew using seed 0. As late-stage availability changes, so do the partitions; stage-level results quantify temporal persistence conditional on the shared representation selection and available prompt populations.

Independent recomputation from raw state features agrees with the state point estimates to well below 10^{-4}. Platform-dependent ordering among nearly tied baseline scores affects AUC-bootstrap endpoints at approximately 4×10^{-5}. Recomputing the 50-split means and corrected intervals from per-split effects agrees to numerical precision.

C.3 Task-action fitting and pairwise evaluation

For task actions, PCA fits 64-dimensional local and global projections on training features. The rank-4 bilinear readout combines linear control/local/global terms with four multiplicative local–global components. Models minimize mean squared error using learning rate 10^{-3}, weight decay 10^{-4}, batch size 512, and at most 25 epochs, choosing the checkpoint on validation data. Five-fold evaluation shuffles problem identifiers with seed 20260901. In each fold, 20% of problems are held out, and one quarter of the remaining problems is reserved for validation, giving 60/20/20% overall train/validation/test proportions.

Layer selection is performed inside each fold. The selected stored layer indices are $(11, 12, 11, 12, 11)$ for GSM8K C+D and $(3, 0, 10, 12, 0)$ for SVAMP. The no-state-interaction model zeros the global input; the control-only model zeros both hidden inputs, using the same scoring architecture. These

models share the nominal parameterization, while the amount of active input information differs. The frozen transfer readout is selected on batch C, with stored layer index 0, rank 4, and fixed projection, normalization, and model weights before application to D.

For each state, evaluate unordered candidate pairs with unequal action labels. Concordance scores a correct predicted ordering as one, an incorrect ordering as zero, and a prediction tie as one half. The reported statistic pools eligible pairs. GSM8K C+D has 892 states with a nontrivial decision among 3,000 states, and 7,157 eligible pairs. Frozen D has 670 such states among 2,280. Eligible decision pairs require unequal action labels.

Task-action intervals use 10,000 problem-cluster bootstrap replicates, with the out-of-fold or frozen predictions held fixed. The bootstrap resamples per-problem correct-pair and total-pair counts. Bootstrap resampling preserves each state's candidate set and replicate multiplicity. It quantifies sampled-problem uncertainty conditional on the fitted predictions.

D Extended state and temporal evidence

The hidden-plus-control state ablation attains $R^2 = 0.4697$ and AUC 0.8885, corresponding to increments of $+0.1317$ and $+0.0655$. The hidden-plus-control result is a secondary nested analysis; the primary comparison uses the hidden-state readout and output-statistics baseline. Gaussian hidden features add approximately -0.000069 in R^2, and a permuted-training-label fit attains $R^2 = -0.00185$. The estimated state-label noise ceiling is 0.9913, so the primary hidden R^2 accounts for approximately 48.5% of this estimated ceiling. This ceiling characterizes noise in the empirical state-value labels.

Prompt-centering yields observed within-prompt variance 0.014775 and estimated label-noise variance 0.001922; their difference is 0.012853 (approximately 0.01285). Centering both features and labels within each prompt gives essentially identical $R^2 \approx 0.0013$ for hidden-augmented and control readouts. A direct centered fit adds essentially zero hidden R^2 across the evaluated layers. Thus global predictive success and within-prompt trajectory tracking remain distinct measurements.

Secondary state-value analyses use selected task-action populations. GSM8K C has hidden increments of $+0.0698$ $[-0.0611, 0.2008]$ in R^2 and $+0.0252$ $[-0.0121, 0.0624]$ in AUC; SVAMP has $+0.0430$ $[-0.0354, 0.1214]$ and $+0.0398$ $[-0.0129, 0.0925]$. All four Nadeau–Bengio intervals over 50 problem-level splits include zero. These smaller, differently selected populations delimit replication beyond the primary 800-prompt GSM8K study.

Table 5: Full temporal state-value results. N is the number of available prompts before splitting. Pointwise intervals use prompt resampling.

Progress	N	ΔR^2 [95% CI]	ΔAUC [95% CI]
10.0%	800	+0.1171 [0.0267, 0.2170]	+0.0643 [0.0242, 0.1091]
25.6%	800	+0.0769 [0.0045, 0.1547]	+0.0506 [0.0186, 0.0857]
40.0%	800	+0.1303 [0.0554, 0.2090]	+0.0511 [0.0127, 0.0932]
55.6%	800	+0.1284 [0.0679, 0.1891]	+0.0538 [0.0243, 0.0858]
71.2%	770	+0.0774 [0.0108, 0.1418]	+0.0300 [-0.0057, 0.0668]
85.6%	441	+0.1320 [0.0724, 0.1935]	+0.0386 [0.0092, 0.0747]

E Selective computation from global state value

State-value scores improve the selection of generations for further computation. We retain the highest-scoring fraction of held-out states, using current features for ranking and future rollouts to evaluate retained correctness. On the pooled test states, retaining 70%, 50%, or 30% raises mean rollout correctness by $+2.36$, $+6.12$, and $+7.94$ percentage points over the output-statistics ranking, with 95% intervals $[0.25, 4.23]$, $[2.47, 9.26]$, and $[2.99, 13.68]$. Retained correctness increases from 74.87% to 77.23%, 80.78% to 86.91%, and 82.07% to 90.01%, respectively (Figure 6). The paired gains establish a use of global state information at the coarse granularity of selecting generations.

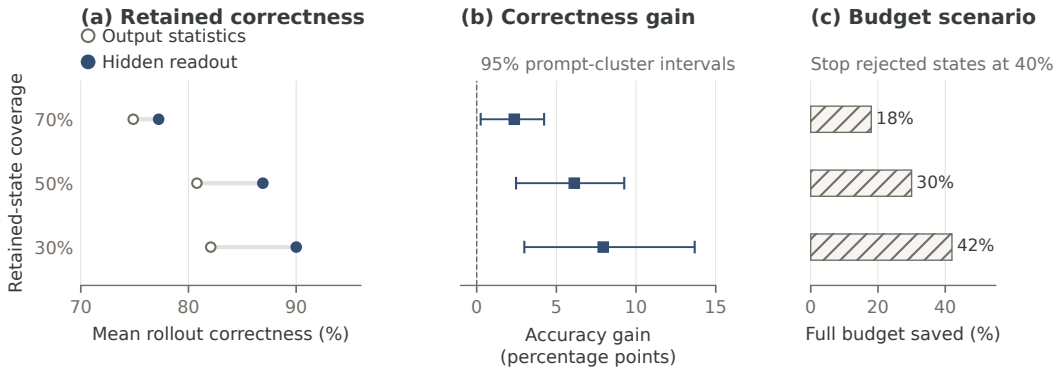


Figure 6: **Selective generation and a fixed-progress budget scenario.** Left: retained correctness on pooled test states. Center: paired gains with prompt-cluster 95% intervals. Right: budget savings $(1-c)(1-0.4)$ at coverage c when rejected states stop at 40% progress.

A separate fixed-progress scenario relates this selection decision to computation. If fraction $1-c$ of generations stops at progress f, the fraction of a full decoding budget saved is $(1-c)(1-f)$. At $f=0.4$, the three coverages correspond to 18%, 30%, and 42% savings. The pooled-state experiment measures ranking quality, while this accounting relation specifies the budget consequence of stopping at a common checkpoint. This separates the demonstrated ranking benefit from its prospective computational use: current-state value supplies the priority signal for allocating future decoding.

At coverage c, each readout independently retains its top $\lfloor cN \rfloor$ of $N = 1{,}098$ pooled test states and averages their eight-rollout success rates. Multiple retained states can belong to one problem. Full-pool mean accuracy is approximately 0.5667. The accuracy–coverage area increases from 0.7722 to 0.8186: $+0.0464$ $[0.0171, 0.0812]$. The accuracy–coverage area integrates retained accuracy over coverage, with larger values indicating better selective ranking.

The separate 40%-progress accounting scenario gives full-budget reductions of 0.18, 0.30, and 0.42 at retained fractions 0.7, 0.5, and 0.3. The corresponding fractions of the *remaining* budget removed are 0.3, 0.5, and 0.7. The accounting scenario and the multi-stage ranking population serve these separate measurements.

F Local Path-LL and task utility

Table 6 gives both primary within-state R^2 and secondary concordance increments. Table 7 reports the candidate-identity controls. All 18 intervals for Gaussian, within-state-shuffled, and across-row-shuffled hidden controls cross zero; training-label permutation produces negative increments. The positive Path-LL readouts therefore depend on candidate-specific feature structure.

Path-LL and task reward are separate utilities. Candidate-advantage Pearson correlations are $+0.0957$ $[0.0248, 0.1727]$ and -0.0174 $[-0.1059, 0.0667]$ in two earlier GSM8K batches, $+0.0493$ $[-0.0222, 0.1208]$ in C, and $+0.2246$ $[0.1391, 0.3094]$ in SVAMP, using 10,000 prompt-cluster resamples. Earlier batches serve this alignment diagnostic alone. These correlations characterize reward alignment across the evaluated populations. Task-accuracy effects are reported in the task-action experiments.

F.1 Finite-state validation of the rollout estimator

An enumerable environment tests the rollout estimator against exact dynamic programming. The environment has a three-token vocabulary plus a mask symbol, a six-position sequence, and state-dependent token probabilities. The production rollout and paired-branch routines use ancestral position selection, full-vocabulary sampling, and horizon $H = 2$. Across four partial states and 18 candidate commitments, each branch uses 2,000 Monte Carlo replicates. The largest standardized discrepancies from exact advantage are 2.13 Monte Carlo standard errors for the per-token target and 1.87 for the downstream target. In these finite-state tests, replacing realized-token scores with

Table 6: Complete local Path-LL increments. CI denotes the Nadeau–Bengio corrected 95% interval over 50 document splits. q adjusts within each four-variant signal feature family.

Model / policy	Target	ΔR^2 [95% CI]	q for ΔR^2
MDLM / ancestral	per-token	+0.00553 [0.00331, 0.00775]	2.9e-05
MDLM / ancestral	downstream	+0.00997 [0.00574, 0.01420]	3.4e-05
MDLM / confidence	per-token	+0.01045 [0.00733, 0.01358]	2.4e-08
MDLM / confidence	downstream	+0.01902 [0.01354, 0.02449]	2.9e-08
SEDD / ancestral	per-token	+0.01143 [0.00801, 0.01486]	2.6e-08
SEDD / ancestral	downstream	+0.02319 [0.01737, 0.02900]	7.3e-10
Model / policy	Target	Δ concordance [95% CI]	Positive splits
MDLM / ancestral	per-token	+0.00330 [0.00069, 0.00592]	46/50
MDLM / ancestral	downstream	+0.00482 [0.00074, 0.00891]	47/50
MDLM / confidence	per-token	+0.00316 [-0.00068, 0.00701]	42/50
MDLM / confidence	downstream	+0.00603 [0.00018, 0.01188]	46/50
SEDD / ancestral	per-token	+0.00125 [-0.00307, 0.00557]	35/50
SEDD / ancestral	downstream	+0.00927 [0.00380, 0.01474]	50/50

Table 7: Candidate-identity controls: mean incremental within-state R^2. All 18 corresponding corrected 95% intervals include zero. The signal readout uses the same control baseline.

Model / policy	Target	Gaussian	Within-state shuffle	Across-row shuffle
MDLM / ancestral	per-token	-0.000077	-0.000044	-0.000029
MDLM / ancestral	downstream	-0.000282	-0.000036	-0.000105
MDLM / confidence	per-token	+0.000006	-0.000198	-0.000046
MDLM / confidence	downstream	-0.000057	-0.000369	-0.000099
SEDD / ancestral	per-token	-0.000075	-0.000035	-0.000046
SEDD / ancestral	downstream	-0.000127	-0.000162	-0.000162

conditional expectations reduces empirical estimator variance by factors of 4.13–9.88, with median 6.15. This finite-state comparison validates the matched-horizon advantage calculation and quantifies variance reduction in an environment with exactly known continuation values.

F.2 Path-LL replication and independent $K = 64$ ranking

The original three Path-LL arms share 400 OpenWebText documents. Experiment A is a frozen-protocol independent replication on 600 new documents at $K = 24$. Experiment B evaluates ranking on a separate 400-document pool at $K = 64$. The two document pools are disjoint from each other, the original pool, and all prior representation- and hyperparameter-selection data. Within each experiment, the three arms use its document pool; the six arm/target conditions remain separate evaluations.

Frozen design and populations. Before replication A and study B traces were collected, we fixed the checkpoints, policies, window construction, recorded stages, candidate construction, forced argmax-token action, $H = 16$ horizon, matched randomness, conditional-expectation estimator, both targets, 100-dimensional controls, hidden blocks, preprocessing, ridge grid, state centering, split proportions, endpoints, null controls, and inference procedure. The retained hidden blocks are $\{6, 9, 12\}$ for MDLM ancestral, $\{5, 8, 11\}$ for MDLM confidence, and $\{6, 9, 12\}$ for SEDD ancestral. These configurations were frozen before either study's data collection and used unchanged throughout both studies. Readouts use each experiment's training and validation partitions under the fixed model-selection procedure.

Each document supplies states at 10%, 20%, 30%, 40%, 50%, and 60% generation progress. Each state supplies six candidates: three uniform positions and three confidence/instability-prioritized positions. A therefore contains 3,600 states and 21,600 candidates per arm; B contains 2,400 states and 14,400 candidates per arm. A uses a fixed 360/90/150-document training/validation/test split, and B uses 240/60/100. All stages, candidates, and paired rollouts from one document remain in

the same partition. B retains the frozen configuration with $K = 64$ paired rollouts per candidate. Its prespecified primary endpoint is downstream pairwise concordance in each of the three arms; per-token concordance is secondary.

Held-out document-level inference. Replication A and study B 95% intervals are percentile intervals from 10,000 document-cluster bootstrap draws on the 150 held-out test documents in A and the 100 in B. Each draw includes all observations from each sampled document and recomputes the paired hidden-plus-control minus control-only metric, conditional on the fitted readouts. For ΔR^2, the target is the within-state action-value contrast. For concordance, both readouts use the same eligible within-state candidate pairs; all stages and pairwise comparisons from a document are resampled together. These intervals quantify held-out document sampling variation. The original repeated-split intervals instead use the correction described above. Their intervals are pointwise 95% intervals for the prespecified conditions.

Replication-A null controls. Experiment A runs Gaussian hidden features, within-state candidate shuffle, and across-row hidden-feature shuffle in each of the six conditions, giving 18 tests. Their measured mean ΔR^2 across conditions is $+0.0002$, -0.0003, and $+0.0001$, respectively. All 18 condition-specific 95% intervals include zero (Table 10). These means summarize null behavior descriptively; condition-specific effects and intervals are the units of inference, with no pooling of observations across conditions.

Table 8 reports replication-A hidden increments over output-and-trajectory controls. Relative to the unrounded original estimates, the six magnitudes retain 84.83–89.28% of their original values. Table 9 compares pairwise gains at the two rollout counts. All downstream intervals exclude zero at $K = 24$; all six do so at $K = 64$. The point estimates stay positive in both experiments. A/B interval differences combine population and rollout-count changes. Larger downstream point estimates recur across arms and metrics; these comparisons describe the observed ordering of effects.

Table 8: **Original and replication-A Path-LL value increments.** Hidden-minus-control within-state ΔR^2; replication-A estimates have pointwise 95% confidence intervals. Both studies use $K = 24$.

Model / policy	Target	Original ΔR^2	Replication A ΔR^2 [95% CI]
MDLM ancestral	Per-token	+0.00553	+0.0049 [+0.0022, +0.0076]
MDLM ancestral	Downstream	+0.00997	+0.0088 [+0.0045, +0.0131]
MDLM confidence	Per-token	+0.01045	+0.0089 [+0.0056, +0.0122]
MDLM confidence	Downstream	+0.01902	+0.0168 [+0.0112, +0.0224]
SEDD ancestral	Per-token	+0.01143	+0.0097 [+0.0061, +0.0133]
SEDD ancestral	Downstream	+0.02319	+0.0207 [+0.0149, +0.0265]

Table 9: **Pairwise-ranking increments in replication A and independent study B.** Hidden-minus-control concordance on rollout labels, with pointwise 95% confidence intervals; $K = 24$ for A and $K = 64$ for B.

Model / policy	Target	A: $K = 24$	B: $K = 64$
MDLM ancestral	Per-token	+0.0028 [+0.0002, +0.0054]	+0.0031 [+0.0010, +0.0052]
MDLM ancestral	Downstream	+0.0044 [+0.0007, +0.0081]	+0.0049 [+0.0022, +0.0076]
MDLM confidence	Per-token	+0.0026 [-0.0005, +0.0057]	+0.0030 [+0.0006, +0.0054]
MDLM confidence	Downstream	+0.0055 [+0.0004, +0.0106]	+0.0062 [+0.0031, +0.0093]
SEDD ancestral	Per-token	+0.0018 [-0.0017, +0.0053]	+0.0022 [+0.0001, +0.0043]
SEDD ancestral	Downstream	+0.0085 [+0.0032, +0.0138]	+0.0094 [+0.0060, +0.0128]

F.3 TASK-ACTION STATE INTERACTIONS

Table 10: **Replication-A candidate-identity controls.** Experiment A ($K = 24$): within-state ΔR^2 over output-and-trajectory controls, with condition-specific pointwise document-cluster 95% intervals.

Model / policy	Target	ΔR^2	95% CI
Gaussian			
MDLM / ancestral	per-token	-0.00010	[-0.00150, +0.00140]
MDLM / ancestral	downstream	+0.00040	[-0.00140, +0.00210]
MDLM / confidence	per-token	+0.00030	[-0.00110, +0.00180]
MDLM / confidence	downstream	+0.00000	[-0.00170, +0.00170]
SEDD / ancestral	per-token	+0.00050	[-0.00100, +0.00200]
SEDD / ancestral	downstream	+0.00010	[-0.00160, +0.00190]
Within-state candidate shuffle			
MDLM / ancestral	per-token	-0.00060	[-0.00200, +0.00080]
MDLM / ancestral	downstream	-0.00010	[-0.00180, +0.00160]
MDLM / confidence	per-token	-0.00040	[-0.00180, +0.00100]
MDLM / confidence	downstream	-0.00020	[-0.00190, +0.00150]
SEDD / ancestral	per-token	-0.00050	[-0.00190, +0.00090]
SEDD / ancestral	downstream	+0.00000	[-0.00180, +0.00180]
Across-row shuffle			
MDLM / ancestral	per-token	+0.00020	[-0.00120, +0.00160]
MDLM / ancestral	downstream	-0.00010	[-0.00180, +0.00160]
MDLM / confidence	per-token	+0.00030	[-0.00110, +0.00170]
MDLM / confidence	downstream	+0.00000	[-0.00170, +0.00180]
SEDD / ancestral	per-token	+0.00040	[-0.00100, +0.00180]
SEDD / ancestral	downstream	-0.00020	[-0.00190, +0.00150]

Table 11: **Global-state interaction effects on task-action concordance.** Concordance increment over the same readout architecture with its global-state input removed. Five-fold cross-fitting; 10,000 prompt-cluster bootstrap draws.

Evaluation	Prompts	ΔC over no-state interaction [95% CI]
GSM8K C+D	750	-0.0022 [-0.0220, 0.0174]
SVAMP	200	+0.0109 [-0.0262, 0.0499]

G MATHEMATICAL GUARANTEES FOR LOCAL ACTION-VALUE MEASUREMENT

G.1 SETUP AND EXACT WITHIN-STATE CANCELLATION

Fix the frozen model, state s, m candidates $\mathcal{A}(s)$, and continuation procedure π, restarted with persistent random keys. Candidates and features precede rollouts. Condition on candidate-set randomness. Action $a = (i, \hat{x}_i)$ forces the current argmax token; $Q_r^\pi(s, a)$ is defined as the forced-branch expectation under action a and continuation procedure π.

Let $Z_t = \log p_\theta(X_t \mid S_t, I_t)$ be the realized token score. Assume at least $H + 1$ masks remain. The action branch forces commitment 0 with score ℓ_a, then makes H stochastic commitments; the reference makes $H + 1$. Define per-token ($r = \mathrm{p}$) and downstream ($r = \mathrm{d}$) rewards:

$$R_{\mathrm{p}}^a = \frac{\ell_a + \sum_{t=1}^H Z_t^a}{H+1}, \qquad R_{\mathrm{p}}^0 = \frac{\sum_{t=0}^H Z_t^0}{H+1}, \tag{8}$$

$$R_{\mathrm{d}}^a = \frac{\sum_{t=1}^H Z_t^a}{H}, \qquad R_{\mathrm{d}}^0 = \frac{\sum_{t=1}^H Z_t^0}{H}. \tag{9}$$

Both rewards use the same commitment budget; downstream scoring assigns zero weight to commitment 0 in both branches. Write $V_r^\pi(s) = \mathbb{E}[R_r^0 \mid s]$, $Q_r^\pi(s, a) = \mathbb{E}[R_r^a \mid s, \mathcal{A}(s), a]$, and $A_r^\pi = Q_r^\pi - V_r^\pi$. Task rewards use the same forced/reference definition with both branches completed.

Assume finite second moments. Predictive risk samples documents, recorded states, and candidates uniformly, equivalent to row averaging in the fixed six-stage, six-candidate design. Let $P_s = I_m - \mathbf{1}\mathbf{1}^\top/m$; tildes denote multiplication by P_s.

Every candidate shares the same additive baseline.

Lemma G.1 (Exact cancellation of additive state-only components)**.** *For every state, candidate set, and reward,*

$$\widetilde{A}_r^\pi(s, a) = Q_r^\pi(s, a) - \overline{Q}_r^\pi(s), \qquad A_r^\pi(s, a_i) - A_r^\pi(s, a_j) = Q_r^\pi(s, a_i) - Q_r^\pi(s, a_j), \tag{10}$$

where $\overline{Q}_r^\pi$ is the uniform candidate mean. Centering eliminates any additive state-only label, feature, or prediction component.

Proof. Since $P_s\mathbf{1} = 0$, applying P_s to $\mathbf{A}_r = \mathbf{Q}_r - V_r^\pi(s)\mathbf{1}$ eliminates the shared baseline. The same identity removes $g(s)\mathbf{1}$ for any state-only g. Subtracting two coordinates cancels every shared term. □

State–candidate interactions remain: centering $d(s)u(s, a)$ gives $d(s)\widetilde{u}(s, a)$. They are part of the local prediction problem. Uniform candidate centering also differs from a policy-weighted baseline: $\widetilde{A} > 0$ denotes value above the sampled candidate mean. $V^\pi(s)$ is the reference-policy baseline.

G.2 UNBIASED MATCHED-ROLLOUT ESTIMATION

Each scored commitment has the same expectation as its realized-token score.

Proposition G.2 (Correct branch marginals identify the advantage)**.** *Suppose both branches have their specified continuation laws and commitment counts. At each stochastic commitment let $\mathcal{G}_t$ contain the pre-token information, including the selected position but excluding unrevealed token noise. Let π_t be the conditional sampling law given this information, and assume*

$$b_t = \sum_v \pi_t(v) \log p_\theta(v \mid S_t, I_t) = \mathbb{E}[Z_t \mid \mathcal{G}_t]. \tag{11}$$

Then the estimators in Equations 3–4 satisfy

$$\mathbb{E}[\widehat{A}_{r,K}(s, a) \mid s, \mathcal{A}(s), a] = A_r^\pi(s, a), \qquad r \in \{\mathrm{p}, \mathrm{d}\}. \tag{12}$$

Proof. The tower property replaces each b_t by Z_t in expectation; the forced score remains ℓ_a. Linearity across branches and replicates gives $Q_r^\pi - V_r^\pi$ under the stated branch marginals and permits arbitrary dependence between paired branches. Candidates were selected before rollout outcomes. □

Under ideal independent streams, top-50 Gumbel-max sampling weights tokens by $p_\theta(v \mid S_t, I_t)^{1/T}$ on the retained support ($T = 1$). Position ordering uses a separate randomness stream independent of unrevealed token noise. Conditional scoring uses these weights against full-distribution log probabilities. Finite-precision pseudorandom arithmetic approximates these ideal laws.

The two targets also have an exact decomposition. Let $\beta_0(s) = \mathbb{E}[b_0^0 \mid s]$. Subtracting the downstream numerator from the per-token numerator gives

$$(H+1)A_{\mathrm{p}}^{\pi}(s,a) = HA_{\mathrm{d}}^{\pi}(s,a) + \ell_a - \beta_0(s), \tag{13}$$

$$(H+1)\widetilde{A}_{\mathrm{p}}^{\pi}(s,a) = H\widetilde{A}_{\mathrm{d}}^{\pi}(s,a) + \widetilde{\ell}_a. \tag{14}$$

Equations 13–14 are population identities. Their finite-K counterparts hold samplewise when both targets use the same paired rollouts: $(H+1)\widehat{A}_{\mathrm{p}} = H\widehat{A}_{\mathrm{d}} + \ell_a - \widehat{\beta}_0$, where $\widehat{\beta}_0 = K^{-1}\sum_k b_{V,k}^{(1)}$. With a reference realization shared across candidates, centering gives $(H+1)\widetilde{\widehat{A}}_{\mathrm{p}} = H\widetilde{\widehat{A}}_{\mathrm{d}} + \widetilde{\ell}_a$. These identities isolate the removed score; larger downstream readout gains remain empirical findings.

G.3 Coupling, shared references, and precision

Coupling affects precision through joint variation across branches.

Proposition G.3 (Covariance of paired local labels)**.** *Conditional on a fixed state and candidate set, suppose complete replicate vectors are independent and identically distributed. For $D_k = U_k^a - U_k^0$, with U the normalized scored branch reward,*

$$\mathrm{Var}(\widehat{A}_K) = \frac{\mathrm{Var}(U^a) + \mathrm{Var}(U^0) - 2\,\mathrm{Cov}(U^a, U^0)}{K}. \tag{15}$$

If Σ_D is the covariance of the full candidate vector $\mathbf{D}_k$, then

$$\mathrm{Cov}(P_s\widehat{\mathbf{A}}_K) = K^{-1}P_s\Sigma_D P_s. \tag{16}$$

With a common reference U_k^0, $P_s\mathbf{D}_k = P_s\mathbf{U}_k^a$ and $D_{i,k} - D_{j,k} = U_{i,k}^a - U_{j,k}^a$ exactly.

Proof. The covariance of an independent-replicate mean is its one-replicate covariance divided by K. Expand $\mathrm{Var}(U^a - U^0)$ and use $P_s = P_s^\top$. The shared-reference cancellation follows from $P_s\mathbf{1} = 0$. □

Pairing reduces scalar variance relative to independent branches exactly when their covariance is positive. After shared-reference cancellation, forced-branch covariance determines centered precision. For a fixed target and replicate law, label standard error scales as $K^{-1/2}$, giving $\mathrm{SE}_{64}/\mathrm{SE}_{24} = \sqrt{24/64} \approx 0.6124$. This $K^{-1/2}$ relation characterizes fixed-population label standard error. A/B bootstrap widths additionally reflect their different document populations.

G.4 Conditional-expectation scoring

Removing token noise locally also changes its covariance with later scores.

Proposition G.4 (Single-step and sequential variance identities)**.** *Under Equation 11, $\mathbb{E}[b_t] = \mathbb{E}[Z_t]$ and*

$$\mathrm{Var}(Z_t) = \mathrm{Var}(b_t) + \mathbb{E}[\mathrm{Var}(Z_t \mid \mathcal{G}_t)]. \tag{17}$$

For a finite trajectory write $L = \sum_t Z_t$, $B = \sum_t b_t$, and $M = \sum_t (Z_t - b_t)$. If the pre-token information sets are nested and contain previous realized scores, then

$$\mathrm{Var}(L) - \mathrm{Var}(B) = \sum_t \mathbb{E}[\mathrm{Var}(Z_t \mid \mathcal{G}_t)] + 2\,\mathrm{Cov}(B, M). \tag{18}$$

Proof. Equation 17 is the variance decomposition for conditional expectation. The residuals $Z_t - b_t$ have zero conditional means and are orthogonal, so their variances add. Expanding $L = B + M$ proves Equation 18. □

Whole-path precision depends on the covariance term in Equation 18; Appendix H gives a concrete example and the paired-contrast identity. In the finite-state validation, conditional scoring reduces empirical estimator variance by 4.13–9.88 times (median 6.15), with estimated advantages agreeing with exact dynamic programming at the Monte Carlo error scale (Appendix F.1).

G.5 Population predictive accessibility

Let $Y = \widetilde{A}_r^{\pi}(s,a)$, $\mathbf{x}$ be centered output-and-trajectory controls, and $\mathbf{h}$ be centered hidden features. Define squared-error risk $\mathcal{R}(f) = \mathbb{E}[(Y-f)^2]$. The row-wise Bayes comparison characterizes unrestricted conditional-mean prediction. The fitted evaluation uses within-state-centered readouts.

Richer readout classes can reduce optimal squared-error risk.

Proposition G.5 (Conditional-mean and readout-class gains)**.** *For $m_x = \mathbb{E}[Y \mid \mathbf{x}]$ and $m_{xh} = \mathbb{E}[Y \mid \mathbf{x}, \mathbf{h}]$,*

$$\mathcal{R}(m_x) - \mathcal{R}(m_{xh}) = \mathbb{E}[(m_{xh} - m_x)^2] \geq 0. \tag{19}$$

For nonempty nested readout classes $\mathcal{F}_x \subseteq \mathcal{F}_{xh}$, each admitting a finite-risk predictor and using the same centering and population, their readout-accessible predictive gain is

$$\Gamma_{\mathcal{F}} = \inf_{f \in \mathcal{F}_x} \mathcal{R}(f) - \inf_{g \in \mathcal{F}_{xh}} \mathcal{R}(g) \geq 0. \tag{20}$$

Proof. Expand $Y - m_x = (Y - m_{xh}) + (m_{xh} - m_x)$. The cross term has zero expectation conditional on $(\mathbf{x}, \mathbf{h})$, proving the first identity. Set inclusion establishes the second inequality directly at the infimum level. □

These are squared-loss comparisons. The Bayes result concerns conditional means, whereas a restricted readout measures accessibility within its function class. These quantities are squared-loss risk differences. Shannon information is a separate measure.

G.6 Linear residual characterization and fitted ridge

Only hidden variation left after control projection can improve the optimal linear fit.

Theorem G.6 (Additional linear prediction is a residual correlation)**.** *Assume Y and every feature coordinate lie in L^2. Let Π_x project in L^2 onto the finite span of the controls and an intercept. Set $e = Y - \Pi_x Y$, $\mathbf{h}_\perp = \mathbf{h} - \Pi_x \mathbf{h}$ (componentwise), $\Sigma_\perp = \mathbb{E}[\mathbf{h}_\perp \mathbf{h}_\perp^\top]$, and $c = \mathbb{E}[\mathbf{h}_\perp e]$. Then*

$$\Gamma_{\text{lin}} = c^\top \Sigma_\perp^\dagger c, \qquad \Gamma_{\text{lin}} > 0 \iff c \neq 0, \tag{21}$$

where $\dagger$ is the Moore–Penrose inverse.

Proof. After control projection, the extra linear fit minimizes $\mathbb{E}[e^2] - 2w^\top c + w^\top \Sigma_\perp w$. For $v \in \ker \Sigma_\perp$, $v^\top \mathbf{h}_\perp = 0$ almost surely, so $v^\top c = 0$. Hence c lies in the range of $\Sigma_\perp$. Minimizing in its positive-eigenvalue coordinates gives reduction $\sum_{\lambda_j > 0} c_j^2/\lambda_j$, yielding both claims. □

A simple example separates linear accessibility from unrestricted information. Let control values be $(1,-1)$ in one state and $(2,-2)$ in another, all four candidates equally likely, and let $h = x^3 = Y$. Features and targets are already state-centered. The optimal affine control fit is $(17/5)x$ with risk $18/5$, while adding h gives zero risk. Yet h is a deterministic function of x, so $I(Y; h \mid x) = 0$.

Fitted readouts. Let $\mathcal{T}$ contain training data, validation choices, and representation selection; let $f_{\mathcal{T}}$ and $g_{\mathcal{T}}$ be the resulting control and hidden-augmented predictors, including the metric's prediction centering. For the fitted comparisons below, $\mathcal{R}$ and the class infima are conditional on $\mathcal{T}$, with that dependence suppressed in notation. Assume each fitted predictor belongs to its corresponding class. Their population gain is

$$\Gamma_{\text{fit}}(\mathcal{T}) = \mathcal{R}(f_{\mathcal{T}}) - \mathcal{R}(g_{\mathcal{T}}) = \Gamma_{\mathcal{F}} + e_x(\mathcal{T}) - e_{xh}(\mathcal{T}), \tag{22}$$

where e_x, e_{xh} are excess risks above the corresponding class infima, all on the same population conditional on $\mathcal{T}$. Adding and subtracting the infima proves the identity. Fitted gains combine approximation, estimation, and regularization effects. Feature parameterization can change a regularized fit even when the unpenalized function class is unchanged (Appendix H). Our empirical claim is improvement over the evaluated fitted controls; the linear-class interpretation in Theorem G.6 applies at the stated optima.

G.7 Unbiased evaluation noise preserves the fitted risk difference

Both fitted predictors face the same evaluation-label error.

Theorem G.7 (Noise cancellation in the fitted squared-error contrast)**.** *Let $\mathcal{X}$ contain the complete current state, candidate set, evaluated candidate action, and input features, so Y is measurable given the fixed continuation procedure. Assume the centered label is $\widehat{Y}_K = Y + \eta_K$, where*

$$\mathbb{E}[\eta_K \mid \mathcal{X}, \mathcal{T}] = 0. \tag{23}$$

Suppose $f_\mathcal{T}, g_\mathcal{T}$ are measurable with respect to $(\mathcal{X}, \mathcal{T})$ and $Y, \eta_K, f_\mathcal{T}, g_\mathcal{T}$ are square-integrable. Then

$$\begin{aligned}&\mathbb{E}[(\widehat{Y}_K - f_\mathcal{T})^2 - (\widehat{Y}_K - g_\mathcal{T})^2 \mid \mathcal{T}] \\ &\quad = \mathbb{E}[(Y - f_\mathcal{T})^2 - (Y - g_\mathcal{T})^2 \mid \mathcal{T}] = \Gamma_{\mathrm{fit}}(\mathcal{T}).\end{aligned} \tag{24}$$

With $v_K = \mathbb{E}[\eta_K^2 \mid \mathcal{T}]$, also $\mathrm{Var}(\widehat{Y}_K \mid \mathcal{T}) = \mathrm{Var}(Y \mid \mathcal{T}) + v_K$.

Proof. Expanding the two squares cancels η_K^2 and leaves the latent squared-error difference plus $2\eta_K(g_\mathcal{T} - f_\mathcal{T})$. Its conditional expectation is zero by Equation 23. Since Y is determined by $\mathcal{X}$, that condition also implies $\mathbb{E}[Y\eta_K \mid \mathcal{T}] = 0$ and $\mathbb{E}[\eta_K \mid \mathcal{T}] = 0$, giving the variance identity. □

Correct branch laws and evaluation randomness conditionally independent of fitting and selection supply Equation 23. Candidate centering preserves conditional mean zero, including with heteroskedastic and correlated candidate errors.

For $0 < \mathrm{Var}(Y \mid \mathcal{T}) < \infty$, population normalized gains obey

$$\Delta R_K^2 = \frac{\Gamma_{\mathrm{fit}}(\mathcal{T})}{\mathrm{Var}(Y \mid \mathcal{T}) + v_K} = \frac{\mathrm{Var}(Y \mid \mathcal{T})}{\mathrm{Var}(Y \mid \mathcal{T}) + v_K} \Delta R_{\mathrm{latent}}^2. \tag{25}$$

For fixed predictors and positive latent variance, conditionally unbiased evaluation noise preserves the population gain's sign and attenuates its normalized magnitude. Changing training K can also change the fitted predictors.

On a finite test set containing complete, equally weighted candidate sets, centering gives zero empirical label mean. With a positive denominator and the sum of squared errors denoted SSE,

$$\widehat{\Delta R^2} = \frac{\mathrm{SSE}_x - \mathrm{SSE}_{xh}}{\sum_{s,a} \widehat{Y}_K(s,a)^2}. \tag{26}$$

Experiment A supplies independent replication evidence for the fitted gain interpreted by this theorem. Appendix C specifies the document-cluster inference and pointwise interval coverage.

G.8 Pairwise ordering

Condition on the fitted scorers and fix a population of within-state unordered candidate pairs with identical weights for both. Here $\mathcal{X}$ includes the evaluated pair and its current inputs. Let $\delta_{ij} = A_r(s, a_i) - A_r(s, a_j)$ and $d_f = f(s, a_i) - f(s, a_j)$. Assume $\Pr(\delta \neq 0) > 0$. On pairs with unequal latent values, define

$$L_f(\delta) = \mathbf{1}\{\delta d_f > 0\} + \tfrac{1}{2}\mathbf{1}\{d_f = 0\}, \qquad \kappa(f) = \mathbb{E}[L_f(\delta) \mid \delta \neq 0]. \tag{27}$$

Ordering depends on candidate differences within a state.

Proposition G.8 (Excess concordance compares local pair ordering)**.** $\Gamma_{\mathrm{rank}} = \kappa(g) - \kappa(f)$ *is the difference in tie-adjusted pairwise ordering accuracy. Replacing A by Q or by centered A, or adding a state-only constant to either scorer, leaves the metric unchanged. A state-only scorer receives* $1/2$.

Proof. Equation 27 gives the first claim. Lemma G.1 preserves all within-state value differences, and a state-only addition preserves prediction differences. A state-only scorer ties every pair. □

Γ_{rank} measures improvement relative to the evaluated control scorer. Absolute concordance is a separate quantity. Pairwise concordance and best-candidate accuracy are separate ranking metrics: for true values $(3, 2, 1)$, descending prediction orders (a_3, a_1, a_2) and (a_2, a_1, a_3) have concordances $1/3$ and $2/3$, while the latter still misses the best action. Path-LL ordering and task-correctness ordering concern distinct rewards.

G.9 FINITE-ROLLOUT RANKING, MARGINS, AND ELIGIBILITY

Condition on the fitted scorers and a fixed pair population. The value gap relative to rollout noise controls agreement between finite-rollout and latent ordering.

Theorem G.9 (Finite-K approximation on a fixed pair population)**.** *Assume $\delta \neq 0$ almost surely, both scorers use no evaluation rollout randomness, and $\widehat{\delta}_K = \delta + \xi_K$ satisfies*

$$\mathbb{E}[\xi_K \mid \mathcal{X}] = 0, \qquad \mathrm{Var}(\xi_K \mid \mathcal{X}) = \sigma_{ij}^2(\mathcal{X})/K < \infty. \tag{28}$$

For this theorem retain every pair and give half credit when either the prediction or noisy label difference is zero. Let κ_K denote that noisy-label concordance and $\Gamma_{\mathrm{rank},K} = \kappa_K(g) - \kappa_K(f)$. Then

$$|\kappa_K(f) - \kappa(f)| \leq \mathbb{E}\left[\frac{\sigma_{ij}^2}{\sigma_{ij}^2 + K\delta_{ij}^2}\right], \tag{29}$$

$$|\Gamma_{\mathrm{rank},K} - \Gamma_{\mathrm{rank}}| \leq 2\mathbb{E}\left[\frac{\sigma_{ij}^2}{\sigma_{ij}^2 + K\delta_{ij}^2}\right]. \tag{30}$$

Both discrepancies converge to zero as $K \to \infty$ on this fixed population and with these fixed scorers.

Proof. The score changes by at most one, and only when the noisy label loses the latent label's strict sign. Cantelli's inequality supplies the one-sided bound. For mean-zero W with variance v, Markov's inequality applied to $(W-t)^2$ bounds $\Pr(W \leq -a)$ by $(v+t^2)/(a+t)^2$ for $a > 0, t \geq 0$. Choosing $t = v/a$ gives $v/(v+a^2)$; apply the same argument to $-W$ for the opposite tail. Conditional on $\mathcal{X}$, set $a = |\delta|$ and $v = \sigma_{ij}^2/K$, then average. The triangle inequality gives the factor two for a difference of scorers. Dominated convergence gives the limits. □

Here σ_{ij}^2 is the variance of the paired candidate difference, including its covariance structure. The bound is strongest for gaps large relative to noise. Equation 30 supplies a sufficient condition for positive latent gain when the noisy-label population gain exceeds the bound; its terms must themselves be reliably bounded to use that condition empirically.

Larger K tightens the bound for a fixed population and fixed scorers, with stronger agreement when value gaps are large relative to rollout noise. Independent experiment B demonstrates positive gains on finite-rollout Path-LL labels in all six conditions, including its three prespecified downstream primary endpoints. A/B interval differences combine population and rollout-count changes. The bound characterizes latent-ranking approximation; applying it to the reported eligible-pair metric additionally requires the eligibility terms in Appendix H.2.

H SUPPLEMENTARY MATHEMATICAL DETAILS

H.1 SEQUENTIAL SCORING AND REGULARIZED PARAMETERIZATION

Sequential scoring. For example, first-token probabilities $(4/5, 1/5)$ followed respectively by a uniform four-token distribution or a deterministic token give the same realized two-step score $L = -\log 5$ on every path. The conditional-score sum has variance $4(\log 4)^2/25 > 0$. For a paired contrast $D^{\mathrm{mc}} = D^{\mathrm{cond}} + N$, the variance difference is $\mathrm{Var}(N) + 2\,\mathrm{Cov}(D^{\mathrm{cond}}, N)$, which also depends on coupling.

Regularized parameterization. Duplicated features illustrate the distinction: if $x = h = Y$ has mean zero and variance one, population ridge with coefficient-wise squared penalty λ and an unpenalized intercept has risk $(\lambda/(1+\lambda))^2$ on x alone, but $(\lambda/(2+\lambda))^2$ on the duplicated pair. The unpenalized function class is unchanged.

H.2 ELIGIBILITY AND IMPLEMENTED WEIGHTING

The reported Path-LL metric pools unordered pairs with $|\widehat{\delta}_K| > \tau$, $\tau = 10^{-9}$, and gives half credit for prediction differences of magnitude at most 10^{-12}. Using that same prediction-tie tolerance in both the latent and noisy scores preserves the theorem's bound. Filtering noisy label ties, however, changes the eligible population as well as the signs.

On the original unfiltered pair population, set

$$\begin{aligned} E &= \{|\delta| > \tau\}, & \widehat{E} &= \{|\widehat{\delta}_K| > \tau\}, \\ p &= \Pr(E), & r &= \Pr(E \triangle \widehat{E}), \\ q &= \Pr(E \cap \widehat{E} \cap \{\operatorname{sign}\delta \neq \operatorname{sign}\widehat{\delta}_K\}). \end{aligned} \tag{31}$$

The symbol $\triangle$ means membership in exactly one event. Let Γ_τ and $\widehat{\Gamma}_{\tau,K}$ be population excess concordances conditioned on E and $\widehat{E}$, respectively. If $p > r$, then

$$|\widehat{\Gamma}_{\tau,K} - \Gamma_\tau| \leq \min\left\{2, \frac{2r + 2q}{p - r}\right\}. \tag{32}$$

Proof. Write each excess as a numerator divided by its eligibility probability. The excess-score integrand has absolute value at most one. On the symmetric difference of eligibility events its change is at most one; on shared eligible pairs with reversed signs its change is at most two. Thus the numerator changes by at most $r + 2q$. The eligibility probability changes by at most r and is at least $p - r$. Since the latent numerator's magnitude is at most p, its denominator change costs at most $r/(p - r)$. Combining gives the stated fraction; two is the universal maximum difference of two excess concordances. □

For any $u > 0$, provided $\mathbb{E}[\sigma_{ij}^2] < \infty$,

$$r \leq \Pr\big(\big||\delta| - \tau\big| \leq u\big) + \frac{\mathbb{E}[\sigma_{ij}^2]}{Ku^2}. \tag{33}$$

Outside this threshold band, eligibility can change only if $|\xi_K| > u$; conditional Chebyshev bounds that event. The eligibility events handle true ties explicitly within the no-ties framework.

Pooled concordance gives each eligible pair equal weight. It differs from averaging per-state concordances when states have different eligible-pair counts. Both fitted scorers use the same eligible pairs. Resampling documents preserves their states and pair statistics together, as well as the original pair weighting and finite-rollout labels.