TESTING ATTRIBUTION CLAIMS IN DIFFUSION-LM REMASKING: A LOCALIZED EFFECT WITHOUT A DEMONSTRATED BENEFIT

Anonymous authors
Paper under double-blind review

ABSTRACT

Masked diffusion language models decode in parallel, and training-free overlays improve them by re-decoding committed tokens. The strongest, CoRe, jointly masks a pool of low-margin candidates and re-predicts them, so a token that confidently holds a neighbor's error in place is rarely perturbed. We ask whether widening the perturbation to each candidate's neighbors helps, and test the resulting method, Local-Context-Dropout (LCD), against the controls a headline comparison omits. It fails them: its $+2.44$ pp HumanEval gain over CoRe is indistinguishable from spending the same extra compute on more CoRe verify calls, no gate configuration dominates, and the frozen method loses 0.86 pp on held-out MBPP problems. What survives is an attribution result. Removing one masked neighbor recovers a median 0.08 of the full-mask effect while restoring it inside the mask removes 0.52, so neighbors act in combination, and adjacent positions carry more of the effect than non-adjacent ones from the same mask, with commit confidence balanced at mask time, across two backbones and three tasks. The association is positional, not proof that proximity is the operative variable, and has no detectable signature in program correctness. Code: https://anonymous.4open.science/r/LCD-iclr27-1EE0/.

1 INTRODUCTION

A masked diffusion language model (MDM) decodes a sequence in parallel: it starts from a fully masked sequence and, over a fixed number of denoising steps, commits tokens where its prediction is most confident (Austin et al., 2021a; Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024). What this loop admits and autoregressive sampling cannot is a family of inference-time interventions, because a committed token can be revised once later context contradicts it. MDMs now reach the quality of autoregressive models at comparable scale (Nie et al., 2025; Ye et al., 2025), and a line of work exploits that directly: CoRe (Zhai et al., 2026), ReMDM (Wang et al., 2025), T2M (Yao, 2026), STaRR (Sun et al., 2025), Self-Reflective Remasking (Huang et al., 2025) and Lookahead Unmasking (Lee et al., 2025) each gate a small revision overlay on top of the standard low-confidence transfer rule. They differ in the selection rule (margin with a verify pass, schedule, threshold, spatiotemporal structure, a critic pass, a simulated commit); only CoRe and Lookahead re-run the model within a step.

Such an overlay buys accuracy without retraining and is cheap. CoRe, the best method on the LLaDA-8B-Base configuration we study, adds one forward pass per verify step, about 6% over the base loop. It builds a verify state by masking its pool of low-margin committed candidates *jointly*, runs one forward pass, and remasks a candidate when that pass proposes a different token with sufficient confidence; it is explicit that this is a perturbation test for conditional dependence, finding "tokens tied to hard consistency constraints (e.g., variable bindings, closing brackets)" (Zhai et al., 2026). The question we ask is not whether perturbing the context is right, which CoRe settles, but whether *the candidate set is the right thing to perturb*.

The candidate set is defined by low margin, which makes it a biased sample of the positions that matter. Consider a committed token at i whose correct value is co-determined by a committed neighbor j that is itself wrong. The rule takes the m smallest margins, a relative ranking rather than

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

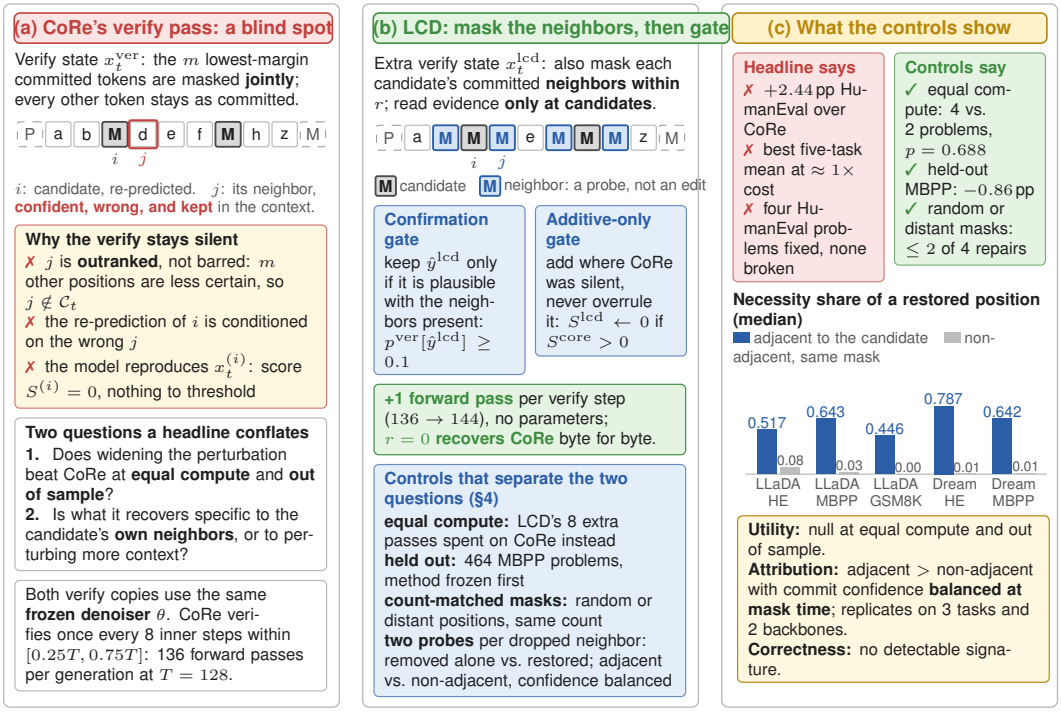

Figure 1: (a) CoRe's verify pass masks its low-margin candidates jointly, so a confident wrong neighbor j stays in context and the error survives. (b) LCD also masks each candidate's neighbors, gates the result, and costs one extra forward pass. (c) The controls give a null utility result and an adjacent-position advantage that replicates across tasks and backbones (HE: HumanEval).

an absolute threshold, so a confident j is excluded whenever m other positions are less certain, the common case. Then every verify state keeps j, the model re-predicts i against the wrong neighbor, and the verify proposes no change. Prior work has varied the selection score and the schedule, not which positions the perturbation touches. Two adjacent literatures bear on the question without answering it. That MDMs weight nearby context more is documented: Piskorz et al. (2026) show on LLaDA-8B-Base and Dream that in-context accuracy falls as relevant prompt material moves away from the masked answer, that the answer logit's gradient concentrates on nearby tokens and on masks, and that appended masks degrade comprehension; whether a remasking overlay's gain rests on any of this is not known. And pairing a removal probe with a restoration probe is standard in attribution, where rationale evaluation contrasts comprehensiveness with sufficiency and their divergence under feature interactions is characterised (DeYoung et al., 2020; Watson et al., 2021; Covert et al., 2021); we claim the application to an applied remasking intervention, not the principle (Appendix M).

Motivated by this, we test the simplest intervention that reaches the excluded positions: Local-Context-Dropout (LCD), a second verify pass that masks each candidate together with its committed neighbors, so the model re-predicts the candidate from a context in which a possibly-wrong neighbor is absent. LCD adds no parameters and one forward pass per verify step, and reduces to CoRe exactly at radius zero (Figure 1). It raises two questions that a headline comparison conflates: whether widening the perturbation recovers anything once CoRe is given the same compute and the configuration is frozen before it meets new problems, and whether whatever it recovers is specific to the candidate's own neighbors rather than to perturbing an equal number of arbitrary committed positions. We answer the first with equal-compute and held-out controls, and the second by instrumenting every applied intervention with two counterfactual probes and comparing adjacent against non-adjacent positions drawn from the same mask, with the confounds that comparison carries measured rather than assumed.

We validate both answers empirically. The utility answer is a null: LCD's $+2.44$ pp HumanEval gain over CoRe is indistinguishable from spending the same extra compute on more CoRe verify

calls ($p = 0.688$), and the frozen configuration scores -0.86 pp on 464 MBPP problems it had never seen. The attribution answer survives its controls: removing one masked neighbor recovers a median 0.08 of the full-mask effect while restoring it inside the mask removes 0.52, and restoring an adjacent position undoes far more of the effect than restoring a non-adjacent one from the same mask, a difference that holds with commit confidence balanced at mask time and replicates on MBPP, GSM8K and Dream-7B.

Key contributions.

- We introduce LCD, a verify pass that perturbs each candidate's neighborhood rather than the candidate set, so the positions a low-margin criterion necessarily excludes can be removed from the context; it costs one forward pass per verify step ($136 \to 144$ passes) and reduces to CoRe exactly at radius zero (§3).
- We report the controls that decide whether it is worth deploying, and they do not support it: a full 2×2 over its two gates in which no cell dominates (20 problems fixed against 15 broken for the reported pair, $p = 0.500$), an equal-compute comparison that leaves LCD and CoRe four problems to two apart ($p = 0.688$), and a held-out evaluation at -0.86 pp (§4.1).
- We separate two claims the design conflates and find they behave differently. A count-matched set of random or distant positions repairs one, two and one of LCD's four HumanEval problems, and none under the distant control; per position, neighbors are rarely sufficient alone (median 0.08) but often necessary in context (0.52), and adjacent positions carry more of the effect than non-adjacent ones from the same mask (39 events against 15, $p = 0.0015$), a result that survives balancing commit confidence at mask time (§4.2).
- We bound that finding with the confounds it carries and the generality it has. It is robust to within-event confidence on MBPP and GSM8K and not on HumanEval, holds within every coarse token class of nine cells but one, does not separate distance within the mask from confidence, replicates across three tasks, two backbones and two radii, and has no detectable signature in which programs become correct, against a test powered 0.88 and 0.98 for a strong association (§4.3).

Code, traces and the generator of every table are at `https://anonymous.4open.science/r/LCD-iclr27-1EE0/`.

2 Problem formulation

Masked diffusion decoding An MDM defines a forward process that replaces tokens of a clean sequence x_0 with a special [MASK] symbol and learns a reverse denoiser that predicts the clean token at each masked position (Austin et al., 2021a; Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024). At inference, sampling starts from a fully masked sequence and proceeds through a schedule of unmasking steps: at each step the model produces a predictive distribution at every position, and a retention rule commits the most confident masked positions, leaving the rest masked (Ghazvininejad et al., 2019; Gu et al., 2019). A partially decoded sequence at step t is $x_t \in (\mathcal{V} \cup \{[\text{MASK}]\})^L$, with $M_t = \{i : x_t^{(i)} = [\text{MASK}]\}$ the masked positions; the model produces $p_t = \text{softmax}(\theta(x_t))$; the transfer rule commits the most confident F_t predictions among masked positions; a revision overlay periodically returns up to $R_t \le F_t$ committed tokens to [MASK]. We write $\text{margin}^{(i)} = p_t^{(i)}[1] - p_t^{(i)}[2]$ for the top-two gap.

CoRe's verify pass At a verify step, CoRe selects the candidate set $\mathcal{C}_t$ as the up-to-m committed, in-block, non-prompt, non-protected positions of smallest margin, forms a verify state x_t^{ver} by masking all of $\mathcal{C}_t$ at once, runs one forward pass $p_t^{\text{ver}} = \text{softmax}(\theta(x_t^{\text{ver}}))$, and scores each candidate i by

$$S_t^{(i)} \;=\; -\text{PLL}_t^{(i)} \cdot \mathbf{1}\left[x_t^{\text{ver},(i)} \neq x_t^{(i)}\right] \cdot \mathbf{1}\left[p_t^{\text{ver},(i)}\left[x_t^{\text{ver},(i)}\right] \ge \tau_{\text{repl}}\right], \tag{1}$$

where $\text{PLL}_t^{(i)} = \log p_t^{\text{ver},(i)}[x_t^{(i)}]$ is the verify log-likelihood of the originally committed token and $\tau_{\text{repl}} = 0.3$. The top-R_t candidates by S_t are remasked. The decision at i is made under a context in which every position outside $\mathcal{C}_t$ is held at its committed value.

Remark 1 (What the candidate rule cannot perturb). Suppose position i is committed to a wrong token whose probability is held up by a committed neighbor j that is also wrong. CoRe's joint mask removes this interaction whenever $j \in \mathcal{C}_t$, so the case that survives is $j \notin \mathcal{C}_t$. Since $\mathcal{C}_t$ is the m smallest margins, this is a relative exclusion: a confident j is not barred from the set, it is outranked whenever m other positions are less certain. Then x_t^{ver} keeps j, the re-prediction at i is conditioned on the wrong j, and if the model reproduces $x_t^{(i)}$ the indicator in Eq. 1 is zero, $S_t^{(i)} = 0$, and CoRe keeps the error. No threshold on τ_{repl} recovers it, because the verify proposes no change to threshold. The remark says only what the rule cannot remove from the context; whether such errors are common, and whether a candidate's own neighbor or the whole neighborhood is what matters, are empirical questions (§4.2). It motivates the design and is not a finding.

The attribution target An *applied intervention* is a position the decoder actually remasked on LCD's evidence, together with the token $\hat{x}$ the local-dropout pass proposed there. For such an event we read the probability the model assigns to $\hat{x}$ at the candidate under three masks: CoRe's mask alone, $p(\text{none})$; LCD's full mask, $p(\text{all})$; and, for each dropped neighbor j, CoRe's mask plus j alone, $p(\text{only } j)$, and LCD's mask with j restored, $p(\text{all but } j)$. The event's *span* is $p(\text{all}) - p(\text{none})$, and on events with positive span the two per-neighbor statistics are the *sufficiency share* $[p(\text{only } j) - p(\text{none})]/\text{span}$ and the *necessity share* $[p(\text{all}) - p(\text{all but } j)]/\text{span}$. These are statements about token probabilities under counterfactual masks, not about which intervention made a program correct; §4.3 measures how far they reach.

3 Local-Context-Dropout

3.1 The local-dropout verify pass

LCD adds a second verify pass that drops the local committed context around each candidate. Let $\mathcal{N}_r(\mathcal{C}_t)$ be the positions within distance r of some candidate that are themselves eligible to be dropped (committed, in-block, non-prompt, non-protected). Form

$$\mathcal{C}_t^{\text{lcd}} \;=\; \mathcal{C}_t \cup \mathcal{N}_r(\mathcal{C}_t), \qquad x_t^{\text{lcd}} = x_t \text{ with [MASK] at } \mathcal{C}_t^{\text{lcd}}, \tag{2}$$

run $p_t^{\text{lcd}} = \text{softmax}(\theta(x_t^{\text{lcd}}))$, and read evidence only at the original candidate positions $\mathcal{C}_t$, using CoRe's rule Eq. 1 applied to p_t^{lcd}. Call this score $S_t^{\text{lcd},(i)}$, with verify argmax $x_t^{\text{lcd},(i)}$. Dropping j's value is intended to remove the reinforcement of Remark 1, so that the model may propose a different token and $S_t^{\text{lcd},(i)} > 0$ where CoRe's score was silent; whether dropping a particular neighbor is what makes such a proposal available is measured in §4.2, not assumed. At $r = 0$, $\mathcal{N}_r = \varnothing$ and the extra pass is identical to CoRe's. Figure 3 (Appendix F) shows the full pipeline.

3.2 Two gates

A naive combination takes the stronger evidence per position, $\tilde{S}_t^{(i)} = \max(S_t^{(i)}, S_t^{\text{lcd},(i)})$, and refills a remasked position from whichever pass won ($\text{LCD}_{\text{naive}}$). This wins on code but regresses reasoning and MBPP (§4.1), and two gates close the two regressions.

Confirmation An LCD proposal can be confident purely because a neighbor was removed: under the full context the proposed token may carry almost no mass. We keep the LCD evidence at i only if its proposed token is also plausible under CoRe's own verify distribution,

$$S_t^{\text{lcd},(i)} \leftarrow \begin{cases} S_t^{\text{lcd},(i)}, & p_t^{\text{ver},(i)}\left[x_t^{\text{lcd},(i)}\right] \geq \tau_{\text{conf}}, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

with $\tau_{\text{conf}} = 0.1$, which filters context-loss artifacts while retaining proposals that survive having their neighbors present.

Additive-only LCD should add discoveries where CoRe was silent, not overrule CoRe where it already decided, so we zero the LCD evidence wherever CoRe's own score fired,

$$S_t^{\text{lcd},(i)} \leftarrow 0 \quad \text{if} \quad S_t^{(i)} > 0. \tag{4}$$

Where CoRe wanted to keep the token, LCD may still propose a remask; where CoRe wanted to change it, CoRe's replacement stands.

Combination After the gates, the combined score and refill source are

$$\tilde{S}_t^{(i)} = \max\big(S_t^{(i)},\, S_t^{\mathrm{lcd},(i)}\big), \qquad \mathrm{refill}(i) = \begin{cases} x_t^{\mathrm{lcd},(i)}, & S_t^{\mathrm{lcd},(i)} > S_t^{(i)}, \\ x_t^{\mathrm{ver},(i)}, & \text{otherwise}, \end{cases} \tag{5}$$

and the top-R_t positions of $\tilde{S}_t$ are remasked, exactly as in CoRe. With both gates on, the LCD contribution is confined to confirmed, CoRe-silent positions.

Remark 2 (Strict superset). At $r = 0$, $x_t^{\mathrm{lcd}} = x_t^{\mathrm{ver}}$, hence $S_t^{\mathrm{lcd}} = S_t$ and every refill source is CoRe's: LCD reduces to CoRe byte for byte, which we use as a sanity check throughout (Appendix E).

3.3 Controls and probes

Three instruments separate the questions the headline conflates. *Count-matched masking controls* keep the number of extra masked positions per row identical to LCD's and change only which positions they are: uniformly drawn eligible committed positions, or eligible positions farther than the radius from every candidate (`random`, `nonlocal`). *Two probes per dropped neighbor* record, for every applied intervention, the four probabilities of §2, at the cost of four forward passes per intervention and without touching the decoding trajectory, so every probe run reproduces its reference score exactly. *Far probes* draw, alongside each candidate's adjacent neighbors, two positions from the same extra mask that are not adjacent to it, either uniformly or, in the *balanced* design, as the unused position whose commit confidence is closest to its adjacent partner's, so the two probe classes are compared at matched confidence by construction rather than by discarding unmatched probes. A radius-2 variant reads the probes at offsets one and two inside one mask. LCD itself costs one forward pass per verify step and no parameters: under CoRe's schedule (verify every 8 inner steps inside $[0.25T, 0.75T]$) at $T = 128$ that is 8 extra forwards, taking 136 passes to 144, a 5.9% increase that §4.1 spends on CoRe instead.

4 Experiments

A headline gain from a remasking overlay can come from more compute, from tuning on the benchmark it is reported on, or from perturbing more context, and none of these is the mechanism the design describes. Our experiments ask whether LCD's gain survives the controls that separate these, and what the per-position probes attribute the effect to. We organize them around three research questions:

- **(RQ1)** Does LCD improve on CoRe once compute is matched and on problems never used to tune it?
- **(RQ2)** Is the effect attributable to the candidate's own neighbors, beyond what an equal number of arbitrary masked positions, commit confidence, or token class would produce?
- **(RQ3)** Does the attribution finding generalise across tasks, backbones and radius, and does it reach program correctness?

Setup We use LLaDA-8B-Base (Nie et al., 2025) with CoRe's decoding and revision configuration unchanged (greedy, 128 steps, 512 generated tokens, verify every 8 inner steps within $[0.25T, 0.75T]$, candidate cap m=32). LCD adds three knobs, radius $r = 1$, $\tau_{\mathrm{conf}} = 0.1$ and additive-only on; r was not tuned, and $r = 2$ reproduces every one of the 164 HumanEval outcomes (Appendix N.2). Benchmarks are the CoRe suite: GSM8K (4-shot) (Cobbe et al., 2021), MATH (4-shot, `minerva_math`) (Hendrycks et al., 2021), BBH (3-shot) (Suzgun et al., 2023), HumanEval (0-shot) (Chen et al., 2021) and MBPP (3-shot) (Austin et al., 2021b), scored on CoRe's headline metric and filter per task. Prompts are pinned by content hash across methods, after GSM8K's few-shot demonstrations were found to differ between our runs and the external baselines on all but 6 of 1319 documents (Appendix C). All runs are single seed (42) on A100-80GB GPUs; greedy decoding makes CoRe and LCD reproducible from the seed alone. Every hyperparameter is in Appendix A. In every table, green marks a positive change against the reference row or a test that rejects at 0.05 (an interval excluding zero), red a negative change or a null, and bold the best value in a column. We compare against **vanilla** low-confidence decoding and **CoRe** run unmodified from the official repository; **ReMDM** (Wang et al., 2025), **T2M** (Yao, 2026), **STaRR** (Sun et al., 2025) and **Lookahead Unmasking** (Lee et al., 2025), reimplemented as training-free decoders on the same

Table 1: Seven methods on LLaDA-8B-Base at seed 42, full benchmarks; cost is forward passes per generation relative to vanilla.

	Reasoning			Code			
Method	GSM8K (4)	MATH (4)	BBH (3)	HumanEval (0)	MBPP (3)	Mean	Cost
Vanilla (`low_conf.`)	0.5178	0.1680	0.4617	0.1220	0.1640	0.2867	1.00×
ReMDM-conf (Wang et al., 2025)	0.4996	0.1522	0.4551	**0.1707**	0.2100	0.2975	1.00×
T2M (Yao, 2026)	0.5118	0.1662	0.4698	0.1463	0.2380	0.3064	1.00×
STaRR (Sun et al., 2025)	0.4898	0.1634	0.4608	0.1220	**0.2620**	0.2996	1.00×
CoRe (Zhai et al., 2026)	**0.5262**	0.1722	**0.4796**	0.1402	0.2540	0.3144	1.06×
Lcd (ours)	0.5254	**0.1728**	0.4795	0.1646	0.2520	**0.3189**	1.13×
Lookahead (Lee et al., 2025)	0.5451	0.2118	0.4890	0.1585	0.2560	0.3321	3.00×
Δ Lcd vs. CoRe	−0.08	+0.06	−0.02	+2.44	−0.20	+0.44	

Table 2: The full 2×2 over the two gates on LLaDA-8B-Base; both gates off is the naive union.

	absolute			Δ vs. CoRe (pp)		
Configuration	GSM8K	HumanEval	MBPP	GSM8K	HumanEval	MBPP
CoRe (neither pass)	0.5262	0.1402	0.2540	–	–	–
τ_{conf} off, additive off	0.5171	0.1585	0.2460	−0.91	+1.83	−0.80
τ_{conf} **on**, additive off	0.5246	0.1646	0.2460	−0.15	+2.44	−0.80
τ_{conf} off, additive **on**	0.5201	0.1585	0.2580	−0.61	+1.83	+0.40
τ_{conf} **on**, additive **on** (Lcd)	0.5254	0.1646	0.2520	−0.08	+2.44	−0.20

frozen backbone (reconstructions, not author-certified reproductions, Appendix B; two sample and are not deterministic; Self-Reflective Remasking is a retrained model and is excluded); and $\textbf{Lcd}_{\textbf{naive}}$, the two single-gate corners and the radius-zero check that must reproduce CoRe (Appendix E).

4.1 RQ1: Utility at matched compute and out of sample

We first answer RQ1 by comparing Lcd with CoRe directly, then by giving CoRe the same compute, then by freezing Lcd and scoring it on problems it had never seen. Table 1 reports the comparison. Lcd improves HumanEval by +2.44 pp over CoRe and differs from it by at most 0.20 pp on GSM8K, MATH, BBH and MBPP, three of those four differences being negative, so the +0.44 pp five-task mean is carried by one benchmark and the four-task mean excluding HumanEval is −0.06 pp. Among methods at about 1× compute Lcd has the highest mean and is the only one above CoRe, with Lookahead exceeding everything at 3× the passes; we do not lean on that ranking, since the external columns are reconstructions, two required filling in under-specified details, and one needed a substantive correction after review (Appendix B). What the intervals license is in Appendix J.2.

The two gates, as a factorial Table 2 gives the full 2×2 over the gates at full scale, since the MBPP signal flips sign between a 150-example subset and the full 500 (Appendix R); the absolute scores of every configuration are in Appendix H. The naive union is a wash: +1.83 pp on HumanEval against −0.91 on GSM8K and −0.80 on MBPP, a four-task mean of +0.06. Confirmation alone moves GSM8K to −0.15 and HumanEval to +2.44 but leaves MBPP at −0.80, which is the diagnosis that the harmful MBPP swaps carry full-context mass and are confidently wrong rather than context-loss artifacts. Additive-only sits where the naive union did on HumanEval (+1.83), so the additive rule contributes nothing on the task carrying the gain, and reaches +0.40 on MBPP against the reported pair's −0.20; GSM8K separates them, additive-only at −0.61 and confirmation bringing it to −0.08. Pooled over the three tasks, adding confirmation fixes twenty problems and breaks fifteen ($p = 0.500$). The gates are neither symmetric nor redundant, no cell dominates, and a reader preferring the single-gate configuration would give up little (Appendix I). The additive rule is additive per position, not per budget: the remask set is a top-k over the merged score, so a permitted Lcd discovery can displace CoRe's own pick elsewhere; we measure that rate and add a budget-preserving variant.

Equal compute Lcd uses 5.9% more computation than CoRe and 12.5% more than the 128-pass baselines. Table 3 spends that budget on CoRe instead, as more verify calls and as more base

Table 3: Giving CoRe the same extra compute; "fwd" is forward passes per generation.

		absolute		Δ vs. CoRe (pp)	
Configuration	fwd	HumanEval	MBPP	HumanEval	MBPP
CoRe (8 verify calls)	136	0.1402	0.2540	–	–
CoRe, 16 verify calls	144	0.1524	0.2560	+1.22	+0.20
CoRe, 144 base steps	152	0.1402	–	+0.00	–
Lcd	144	0.1646	0.2520	+2.44	−0.20
Lcd, budget-preserving	144	0.1524	–	+1.22	–

decoding steps. This control does not go our way: spending Lcd's extra eight passes on more CoRe verify calls recovers +1.22 pp of the +2.44, and head to head at 144 passes the two differ by four problems to two ($p = 0.688$), so at equal compute the difference is not distinguishable from zero here. They repair disjoint problem sets, so they are not the same intervention, but that does not rescue an attribution claim; more base decoding steps do nothing; and forbidding Lcd to displace CoRe's pick costs two of the four fixes, so half the headline gain comes from the global competition the additive rule does not prevent (Appendix Q).

Held out Every number above comes from benchmarks also used to diagnose the regressions and choose the gates, and stricter re-grading does not fix this since HumanEval+ re-grades the same 164 problems. We therefore froze the configuration and evaluated it once on the 374 train and 90 validation MBPP problems, 464 in total, disjoint from the 500-problem test split and from the three demonstrations, scored by the identical execution-based pass@1. It does not improve on them: 0.2866 against CoRe's 0.2953, a difference of −0.86 pp with three problems fixed and seven broken ($p = 0.344$, $[-2.16, +0.43]$). The run is a valid test, not a broken one, since CoRe improves on vanilla by +10.13 pp on the same problems. The set is MBPP-family and Lcd was already at −0.20 on MBPP test, so this does not contradict the HumanEval measurement; it rules out the reading that the configuration generalises as a code-generation improvement.

These results answer RQ1: on HumanEval, in this configuration, Lcd fixes four problems and breaks none (Appendix V), with no evidence that this carries to other problems or beats CoRe at equal compute.

Takeaway 1

At equal compute Lcd and CoRe are indistinguishable (four problems to two, $p = 0.688$), and the frozen configuration loses 0.86 pp on 464 held-out MBPP problems. The +2.44 pp HumanEval gain is a null utility result, so Lcd is an attribution instrument rather than an improvement.

4.2 RQ2: What the perturbation can be attributed to

We next answer RQ2 in two steps, at the level of whole runs and then at the level of single interventions.

Count-matched masking Lcd masks the union of the neighborhoods of all candidates, so a candidate sits in a context from which both its own neighbors and positions near other candidates have been removed, and "local" and "perturbed more" cannot be separated from the headline alone. Table 14 (Appendix K) holds the number of extra masked positions fixed and changes only which they are (a first version perturbed about 39% more context than Lcd; the corrected decoder matches the target at 40.14 against 40.29 masks per verify call). Lcd repairs four HumanEval problems; the random draws repair one, two and one, and the distant control repairs none, scoring exactly CoRe's 0.1402 despite masking forty extra positions per call. Lcd never loses a problem to any control, though no head-to-head reaches significance: each rests on two to four discordant problems at exact p between 0.125 and 0.500, and pooling the three draws is not eight independent observations, since the same three problems (60, 82, 85) recur, but three distinct problems at $p = 0.250$.

Table 4: Adjacent versus non-adjacent positions from the same mask, under uniform and confidence-balanced far sampling.

	HumanEval uniform		HumanEval balanced		MBPP uniform		MBPP balanced		GSM8K balanced	
	value	p	value	p	value	p	value	p	value	p
necessity share, adjacent vs. non-adjacent										
within-event difference (sign test)	$+0.233$	0.0015 ✓	$+0.390$	0.0038 ✓	$+0.482$	$< 10^{-4}$ ✓	$+0.461$	$< 10^{-4}$ ✓	$+0.367$	$< 10^{-4}$ ✓
within-event permutation	$+0.273$	$< 10^{-4}$ ✓	$+0.148$	0.0010 ✓	$+0.305$	$< 10^{-4}$ ✓	$+0.289$	$< 10^{-4}$ ✓	$+0.264$	$< 10^{-4}$ ✓
cluster bootstrap 95% CI	$[+0.053, +0.500]$ ✓		$[+0.025, +0.433]$ ✓		$[+0.211, +0.392]$ ✓		$[+0.210, +0.353]$ ✓		$[+0.189, +0.345]$ ✓	
sufficiency, within-event difference	$+0.023$	0.1337 ✗	$+0.010$	0.3497 ✗	$+0.190$	$< 10^{-4}$ ✓	$+0.200$	$< 10^{-4}$ ✓	$+0.192$	$< 10^{-4}$ ✓
events / problems / probes	60 / 41 / 209		60 / 41 / 178		326 / 201 / 1153		326 / 201 / 1002		692 / 520 / 1998	
median commit confidence (adj. / non-adj.)	0.727 / 0.969		0.727 / 0.727		0.727 / 0.953		0.727 / 0.738		0.664 / 0.664	

Two probes The run-level controls cannot say what a given intervention did, while the design's story is per position: a candidate's error is held in place by a particular neighbor, and removing it frees the correct token. The probes of §2 test that on every applied intervention, and they disagree; the disagreement is the finding. A neighbor removed alone recovers a median 0.08 of the effect that removing the whole neighborhood produces, while restoring that same neighbor inside the full mask removes a median 0.52 of it, so the neighbor acts in combination, and on 17.6% of applied interventions one is necessary at our 0.5 threshold without being sufficient at it (Appendix U). An earlier version drew the opposite conclusion from the first probe alone; we withdraw it. The result establishes that neighbors matter in combination, not that *neighbors* are what matters, since any jointly masked position might show the same profile.

Adjacent against non-adjacent Table 4 compares, within one run, each candidate's immediate neighbors with two positions from the same extra mask that are not adjacent to it, on events carrying both classes (60 on HumanEval, 326 on MBPP, 692 on GSM8K). The sign test pairs within event and is two-sided; the permutation test is one-sided and shuffles adjacency labels among an event's own probes, controlling within-event but not between-problem dependence; the bootstrap resamples problems and controls the complementary level, and neither alone treats the full nesting (Appendix N). On HumanEval, restoring an adjacent position removes a median 0.517 of the effect against 0.081 for a non-adjacent one, the adjacent position being larger on 39 events against 15 ($p = 0.0015$; permutation $+0.273$, $p < 10^{-4}$; bootstrap $[+0.053, +0.500]$); on MBPP the same test runs $+0.643$ against $+0.027$ over 326 events ($p < 10^{-4}$; bootstrap $[+0.211, +0.392]$). Isolated removal moves the same way, reaching significance on MBPP ($+0.259$ against $+0.014$, $p < 10^{-4}$) though not on HumanEval ($p = 0.134$), so we do not read the two probes as separate channels.

Confidence Adjacency travels with a second property: positions next to a candidate are themselves less settled, with median commit confidence 0.727 for the adjacent probes against 0.969 on HumanEval and 0.953 on MBPP for the non-adjacent ones, so the comparison is partly a confidence one. Matching pairs after the fact separates the two only partly, no advantage on HumanEval (41 pairs, $p = 1.00$) and a clear one on MBPP (268 pairs, $+0.183$, $p < 10^{-4}$), the HumanEval null having a measurable cause in the unusually confident adjacent probes matching retains (Appendix N). The balanced design of §3 builds the balance into the sampling instead; both runs reproduce their reference scores exactly, the match is close but not exact (mean absolute confidence gap 0.024 on HumanEval, 0.021 on MBPP), and it changes probe count as well as confidence, which a count-matched subanalysis bounds (Appendix N). The advantage holds under it on both tasks and on every event (Table 4: HumanEval permutation $+0.148$, $p = 0.0010$, bootstrap $[+0.025, +0.433]$; MBPP $+0.289$, $p < 10^{-4}$, $[+0.210, +0.353]$), and on GSM8K, a non-code task, on 692 events across 520 problems with confidence 0.664 against 0.664 (necessity $+0.446$ against $+0.002$, 472 against 189, $p < 10^{-4}$; bootstrap $[+0.189, +0.345]$). Within event, it holds whichever probe is the more confident on MBPP and GSM8K and not on HumanEval, where it is carried by the events in which the adjacent probe is the less confident (Appendix N).

Token class and distance within the mask Balancing confidence does not make the two classes exchangeable: they still differ in token identity, syntactic role, and whatever else co-varies with adjacency to an intervention. Classing tokens coarsely (whitespace, punctuation, word), the adjacent median leads within every class of every cell but one, whitespace on GSM8K tying, and the like-class paired test is significant on five of nine cells (Appendix N). A radius-2 probe extends the advantage

to offset 2 on MBPP, but distance *within* the mask is not separable from confidence: offset 1 over offset 2 holds only where offset 1 is the more confident probe (Appendix N.2).

These results answer RQ2: the effect is attributable to the candidate set's neighborhood rather than to any equal-count perturbation, and, per position, to offset from the candidate beyond commit confidence and coarse token class where the sample has power; whether proximity itself is the operative variable, no probe drawn from this mask can settle.

Takeaway 2

The effect belongs to the candidate set's neighborhood, not to perturbing more context: count-matched random or distant masks repair at most two of LCD's four problems, and within one mask an adjacent position carries more of an intervention's effect than a non-adjacent one (39 events against 15, $p = 0.0015$), with commit confidence balanced at mask time.

4.3 RQ3: GENERALITY, AND WHETHER IT REACHES CORRECTNESS

We answer RQ3 by repeating the comparison on a second backbone and a wider mask, and by testing the attribution statistics against outcomes. On Dream-7B (Appendix N.1), with the same configuration and both runs reproducing their reference scores, the advantage is larger than on LLaDA-8B-Base: restoring an adjacent position removes a median 0.787 of the effect against 0.009 on HumanEval (59 against 16, $p < 10^{-4}$) and 0.642 against 0.011 on MBPP (133 against 52), it survives balancing (HumanEval 60 against 15; MBPP 137 against 47, closest pairs 113 against 65, $p = 0.0004$), and isolated removal is significant under both designs. The two-probe divergence itself is not specific to one backbone, task or radius: the same battery gives a positive within-event contrast in all six cells spanning LLaDA-8B-Base and Dream-7B, HumanEval, MBPP and GSM8K, radius one and two, the sign test rejecting and the problem-level interval excluding zero in each, smallest on GSM8K ($+0.019$, $[+0.008, +0.037]$) and Dream MBPP ($+0.049$, $[+0.004, +0.127]$) (Appendix O).

It does not reach correctness. Every event carries a document id, so we can ask whether problems whose outcome LCD changes differ in attribution statistics from those it leaves alone. They do not: on HumanEval the four repaired problems have a lower median necessity share than the 37 unchanged ones ($+0.17$ against $+0.56$, $p = 0.32$), on MBPP the six changed problems have a lower within-event contrast than the 195 unchanged (-0.07 against $+0.15$, $p = 0.10$), both differences running opposite to a causal reading, against a test powered 0.88 and 0.98 for a strong association and 0.24 and 0.39 for a moderate one (Appendix P).

These results answer RQ3: the finding replicates across three tasks, two backbones and two radii at the level of token probabilities, and what these data exclude is a strong link to correctness, not every link.

Takeaway 3

The positional association replicates on three tasks, two backbones and two radii at the level of token probabilities, yet has no detectable signature in which programs become correct: it is a finding about what the decoder's evidence depends on, not a mechanism for a gain.

5 CONCLUSION

CoRe's verify pass perturbs its candidate set, so it rarely removes a confident position, and LCD was built to reach the positions it cannot. What survives the controls is attribution, not utility: adjacent positions account for more of an intervention's effect than non-adjacent ones from the same mask, on three tasks and two backbones, robust to within-event confidence on MBPP and GSM8K but not on HumanEval, while at equal compute LCD and CoRe differ by four problems to two and the frozen configuration scores -0.86 pp held out. It has no detectable signature in program correctness; closing that gap needs an intervention that changes correctness, not a further probe of the same distributions, and what transfers from this paper is the checks (Appendix K).

Reproducibility Statement

Algorithm 1 is fully specified and Appendix F gives the exact tensor operations for the neighbor mask, the confirmation gather, and the additive mask. The reference implementation is two files (`generate_core2.py`, `eval_core2.py`), drop-ins for the official CoRe repository that need no changes to CoRe's source, selected by the remasking string `core2` and controlled by environment variables (Appendix A). All decoding is greedy, so the numbers are deterministic and seed-invariant, with the exception of the two external baselines that sample (Appendix B). Three additions in this revision are what we would most want a reader to check. The prompt-pinning utility and its reference file make the few-shot prompts identical across methods and independent of the GPU count (Appendix C); every table of paired per-problem outcomes is emitted by a single script from the samples files, on the filter that defines each task's headline metric, so no paired number is transcribed by hand (Appendix J.1); and the held-out task definition (§4.1) is a task configuration file whose problem ids can be checked against the evaluated split by inspection. The radius-zero equivalence to CoRe (Appendix E) is a mechanical check anyone can run. All scripts are released as-is at https://anonymous.4open.science/r/LCD-iclr27-1EE0/.

Ethics Statement

LCD is an inference-time decoding method that introduces no new data, no new training, and no new model parameters, does not alter the underlying model's training distribution, and cannot produce content the base model could not produce without it. We do not anticipate it introducing biases or harms beyond those already present in the base MDM.

Use of Large Language Models

During the preparation of this paper, we used a large language model to draft and revise text and to improve its grammar, word choice, and conciseness. We also used it to help identify relevant literature. The authors independently read all suggested papers and verified each citation against its original source before inclusion. We used Claude Code to assist with designing and running the experiments and analyzing the results. The authors reviewed the code and verified all reported numerical results against the saved outputs. All claims are the authors' own, and we take full responsibility for the content of this paper.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021a.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021b.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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. *arXiv preprint arXiv:2110.14168*, 2021.

Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4443–4458, 2020.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 2016.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-Predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6111–6120, 2019.

Jiatao Gu, Changhan Wang, and Jake Zhao. Levenshtein transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Zemin Huang, Yuhang Wang, Zhiyang Chen, and Guo-Jun Qi. Don't settle too early: Self-reflective remasking for diffusion language models. *arXiv preprint arXiv:2509.23653*, 2025.

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

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2024.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 56354–56392, 2025. arXiv:2502.09992.

Julianna Piskorz, Cristina Pinneri, Alvaro Correia, Motasem Alfarra, Risheek Garrepalli, and Christos Louizos. Masks can be distracting: On context comprehension in diffusion language models. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. arXiv:2511.21338.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184, 2024.

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

Xinhao Sun, Huaijin Zhao, Maoliang Li, Zihao Zheng, Jiayu Chen, Yun Liang, and Xiang Chen. STaRR: Spatial-temporal token-dynamics-aware responsive remasking for diffusion language models. *arXiv preprint arXiv:2601.04205*, 2025.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023. arXiv:2210.09261.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 163488–163545, 2025. arXiv:2503.00307.

David S. Watson, Limor Gultchin, Ankur Taly, and Luciano Floridi. Local explanations via necessity and sufficiency: Unifying theory and practice. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1382–1392, 2021.

Lin Yao. Remask, don't replace: Token-to-mask refinement in diffusion language models. *arXiv preprint arXiv:2604.18738*, 2026.

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

Kevin Zhai, Sabbir Mollah, Zhenyi Wang, and Mubarak Shah. CORE: Context-robust remasking for diffusion language models. *arXiv preprint arXiv:2602.04096*, 2026.

A HYPERPARAMETERS

LCD adds three knobs on top of CoRe's decoding configuration, and this revision adds four switches that select the controls rather than change the reported method. Everything else is CoRe's default. Table 5 lists all of them with the values used throughout the main text. We did not tune r, τ_{conf} was chosen once as described in §4.1, and no value was tuned on the held-out set of §4.1.

Table 5: All hyperparameters, grouped as CoRe decoding, CoRe revision, LCD's additions and the control switches.

Parameter	Value	Description
`steps`	128	inner denoising steps
`gen_length`	512	generated tokens
`block_length`	512	block size (single block)
`temperature`	0	greedy decoding
`cfg_scale`	0	no classifier-free guidance
`REVISE_EVERY`	8	verify cadence in inner steps
`CANDIDATE_M` (m)	32	candidate cap per sample per verify call
remask budget R_t	1	positions remasked per call
verify window	$[0.25T, 0.75T]$	steps in which revision is active
τ_{repl}	0.3	replacement-confidence threshold (CoRe)
k_{protect}	16	trailing in-block positions never remasked
`LCD_RADIUS` (r)	1	neighbor-drop radius (0 recovers CoRe)
`LCD_CONFIRM_TAU` (τ_{conf})	0.1	full-context confirmation threshold
`LCD_ADDITIVE`	on	forbid LCD from overriding CoRe at a position
`LCD_MASK_MODE`	`neighbor`	count-matched controls: `random`, `nonlocal` (§4.2)
`LCD_CORE_PRIORITY`	off	spend the remask budget on CoRe's picks first (§4.1)
`LCD_TRACE_DIR`	unset	dump per-event mechanism traces (Appendix U)
`PROMPT_REF`	set	pin few-shot prompts to the reference (Appendix C)

B REIMPLEMENTATION OF BASELINE METHODS

The four external baselines in Table 1 are not available for LLaDA-8B-Base in a common harness, so we reimplement each as a training-free decoder on the frozen backbone, following its published algorithm as closely as a fixed-step decoder allows. We no longer call these reimplementations faithful. Two of them required reconstruction of under-specified details (STaRR's spatial weights and confidence definition), one required a substantive correction after review (ReMDM's stored confidence, below), and one algorithm we do not implement at all in its published LLaDA-8B-Base form for the σ-draw variant. We document each method, its hyperparameters, the adaptations a fixed-step decoder forced, and a methodological pitfall we hit and corrected, because it materially changed the numbers. Readers should treat the external columns as our best reconstruction under a common harness rather than as reproductions certified by the original authors.

Common scaffolding All four share CoRe's decoding configuration (Table 5, top block) and reuse the same model forward and block schedule. ReMDM, T2M, and STaRR are *remasking overlays*: they return some committed tokens to [MASK] and let the standard transfer rule re-decode them; they add no forward passes ($1.00\times$). Lookahead replaces the commit *selection* with a simulate-and-score step and adds k forward passes ($3.00\times$ for k=2).

ReMDM (Wang et al., 2025) The submitted version had two defects here, both corrected. First, we described ReMDM-conf as "the one the paper applies to LLaDA-8B-Base". It is not: ReMDM's LLaDA-8B-Base experiments (their Appendix D.4) replace the σ-draw sampler with a modified ReMDM-loop, which generates most of the sequence with the LLaDA-8B-Base sampler, then enters a loop phase that remasks a fixed number of tokens *with the smallest decoding confidence* and unmasks the same number each step, then fills the remainder. Second, our ReMDM-conf computed $\psi^{(\ell)}$ from the current step's logits, whereas their Algorithm 2 *stores* ψ when a token is decoded ("Store confidence scores $\psi_t^{(\ell)} = (z_s^{(\ell)})^\top x_\theta(z_t)^{(\ell)}$ for all newly decoded $z_s^{(\ell)}$") and reuses it later; CoRe (Zhai et al., 2026) makes the same reading, criticising ReMDM for using "stale confidence scores from past decoding steps". We now report both methods. *ReMDM-conf* returns each committed position to [MASK] with probability $\sigma_t^{(\ell)} = \mathrm{softmax}_{\ell \in \text{committed}}(-\psi^{(\ell)}) \cdot \eta_{\text{rescale}} \sigma_t^{\max}$, with ψ the stored decoding confidence, $\sigma_t^{\max} = \min(1, (1-\alpha_s)/\alpha_t)$ on the loglinear schedule, and $\eta_{\text{rescale}} = 0.9$ combined with the confidence schedule, which is the combination their ablation selects. *ReMDM-loop* follows Appendix D.4: the standard sampler runs for the first 87.5% of the step budget (their 28 of 32 tokens), then each remaining step remasks the single committed token of smallest stored decoding confidence and credits it back to the transfer budget. ReMDM-loop is deterministic under greedy prediction, since it is a top-k rule rather than a Bernoulli draw. Neither adds forward passes.

T2M (Yao, 2026) We implement the LowProb detector, the paper's default: a committed token is returned to [MASK] when its own probability is below $\tau_{\text{lp}} = 0.7$, subject to a per-position budget $C_{\max} = 3$ within a block. The freed token is re-decoded by the normal transfer rule ("remask, don't replace"). No extra forward passes.

STaRR (Sun et al., 2025) We adopt STaRR's per-position dynamic threshold $\tau_t^{(i)} = p\,\tau_{\text{base}} + (1-p)\,\tau_{\text{neigh}}$ as the remask criterion, with the temporal-variance term $\tau_{\text{base}} = m \cdot \mathrm{var}_t^{(i)}$ over a length-W_t=3 confidence history and the spatial-deviance term over a $\pm W_n$=3 neighborhood with exponentially-decaying, renormalized weights, $\tau_{\text{fixed}} = 0.9$, $m = 3$. The paper's general spatial-weight form and confidence definition are under-specified; we use $w_d \propto 2^{-d}$ and max-softmax confidence and flag these as reconstructions. We apply the threshold as a remask criterion inside the fixed-step transfer; STaRR's orthogonal variable-commit *speedup* is not relevant to a matched-step quality comparison and is not implemented. No extra forward passes.

Lookahead Unmasking (Lee et al., 2025) At each step we form a candidate pool of the top-N masked positions by confidence, generate k=2 candidate commit-sets of the step's budget size (one greedy, the rest sampled from the pool), tentatively commit each (greedy token values at T=0), run the model on each tentative state, and keep the set with the lowest average next-state entropy. Lookahead is commit-forward only (it never remasks) and costs k extra forward passes per step. We use the schedule's per-step budget as the commit-set size so the block completes deterministically.

A pitfall we corrected: the end-dump artifact Our first implementation of the three remasking overlays *deferred* every re-decode without returning the freed slot to the step schedule, so aggressively-remasked tokens accumulated and were filled in a single shot at the final step from a fully-committed (hence maximally-informative) bidirectional context. This last-step "end-dump" is an artifact of a fixed-step loop, not part of any of these methods, and it inflated all three baselines dramatically (for example T2M reached 0.6058 on GSM8K, $+8$ pp over CoRe, which is implausible for a confidence-threshold remask). We caught it because the margins were too large to be credible, traced it to the end-dump (a control with vanilla routed through the same harness reproduced the known vanilla score, isolating the remask path), and fixed it by crediting each remask back into the step budget so freed tokens re-decode *within* the schedule, exactly as CoRe does. After the fix the end-dump fires on a negligible fraction of generations and the same T2M configuration scores 0.5034 on GSM8K, in line with the literature (this is the value before the prompt repair of Appendix C, which was applied later and moves it to 0.5118; the two corrections are independent). All numbers in this paper are post-fix; we report the pitfall because a fixed-step reimplementation of any deferred-remask method is liable to it, and because the inflated and corrected numbers differ by enough to flip the entire ranking.

Fidelity checks on the two sampling baselines Two consequences of the corrections above are worth reporting separately. First, the ψ correction and the LLaDA-8B-Base-specific loop procedure change ReMDM's numbers, and Table 6 quantifies by how much on the three discriminative tasks. We did not re-run MATH and BBH for these variants, so Table 1 keeps the σ-draw reconstruction in its ReMDM row and we report the corrected variants as a fidelity check rather than as a replacement column; readers weighing the multi-method ranking should discount the ReMDM row accordingly. Second, greedy token prediction does not make either sampling baseline deterministic: ReMDM-conf draws its masks from a Bernoulli and Lookahead samples its candidate commit-sets, so their single-seed numbers carry algorithmic uncertainty that CoRe and Lcd do not. Table 7 gives three seeds each on HumanEval as a measure of that spread. ReMDM-loop is exempt, being a deterministic top-k rule.

Table 6: Effect of the two ReMDM corrections; rows are the submitted reconstruction, the corrected one, and their Appendix D.4 procedure.

ReMDM variant	GSM8K	HumanEval	MBPP
as submitted (ψ from the current step)	0.4822	0.1707	0.2100
corrected (ψ stored at decode, η_{rescale}=0.9)	0.4996	0.1646	–
ReMDM-loop, the paper's LLaDA-8B-Base procedure	–	0.1280	–

Table 7: HumanEval pass@1 at three seeds for the two baselines that sample.

Method	seed 42	seed 1337	seed 7	spread (pp)
ReMDM-conf	0.1707	0.1768	0.1768	+0.61
Lookahead	0.1585	0.1585	0.1585	+0.00

Self-Reflective Remasking, excluded Self-Reflective Remasking (Huang et al., 2025) attaches a separately-trained per-token confidence stream to the backbone (an 8.9B retrained dual-stream model). It is not a training-free decoder and cannot be reproduced on a frozen LLaDA-8B-Base without retraining, so we exclude it rather than substitute a proxy that would not represent the method.

C Prompt Matching Across Methods

The submitted version asserted that all methods saw byte-identical prompts. That was true for four of the five tasks and false for GSM8K, and since the comparison is the paper's main claim we describe the defect, its cause, and the repair in full.

What was wrong For each of ReMDM, T2M, STaRR and Lookahead, only 6 of the 1319 GSM8K prompts matched the corresponding CoRe and Lcd prompts. The evaluation question and target matched in all 1319 cases; the four-shot demonstrations differed. The GSM8K column of the submitted table therefore mixed decoder differences with prompting differences. The other four tasks were unaffected, and the direct CoRe-versus-Lcd comparison was unaffected, because those two runs shared prompts exactly.

Cause lm-evaluation-harness builds few-shot contexts once per rank, iterating over that rank's shard of the documents and drawing demonstrations from a single task-level generator whose state advances with every document. The demonstrations a document receives are therefore a function of its position within its rank's shard, and so of the number of ranks. Our CoRe and Lcd runs used four GPUs and the external-baseline sweep used three. Replaying the harness's sampler offline confirms this exactly: at four ranks it reproduces the CoRe and Lcd prompts on all 1319 documents, and at three ranks it reproduces each baseline's prompts on all 1319. The six coincidences are the documents whose draw happens to agree between the two shardings. GSM8K is the only affected task of the five because it samples demonstrations from the training split, whereas MBPP, MATH and BBH use fixed demonstrations declared in their task configurations (we verified that replays at one and at four ranks are identical for every one of their subtasks) and HumanEval is zero-shot.

Repair The repair moves the four external GSM8K numbers by between -2.05 and $+1.74$ pp (Table 8), which is larger than several of the per-task differences this paper reports, and it changes the ordering among the external methods: STaRR and T2M swap, and Lookahead's five-task mean falls from 0.3360 to 0.3321. The CoRe-to-Lcd comparison is unaffected, since those two runs already shared prompts exactly.

Table 8: GSM8K for each external method with prompts pinned to the CoRe reference, against the submitted value.

Method	GSM8K (prompt-matched)	GSM8K (submitted, mismatched)	change (pp)
ReMDM-conf	0.4996	0.4822	+1.74
T2M	0.5118	0.5034	+0.83
STaRR	0.4898	0.5102	−2.05
Lookahead	0.5451	0.5648	−1.97

We pin prompts to a reference file keyed by a content hash of the evaluation document, so the prompt a document receives no longer depends on the GPU count. The reference is built by replaying the harness's own sampler, and we verified it reproduces the CoRe and Lcd prompts on all 1319 GSM8K documents before using it. All four external baselines were then re-run on GSM8K under that reference, and we checked prompt-hash equality against CoRe for every document after the re-run rather than assuming it. Table 1 reports the re-run numbers. The mechanism is not specific to this paper: any multi-GPU lm-evaluation-harness comparison whose runs used different process counts will differ in its sampled few-shot demonstrations, silently, and we release the pinning utility for that reason. Figure 2 shows one pinned prompt in full.

GSM8K, pinned four-shot prompt

```
Question: Jen and Tyler are gymnasts practicing flips. Jen is practicing the
  triple-flip while Tyler is practicing the double-flip. Jen did sixteen triple-flips
  during practice. Tyler flipped in the air half the number of times Jen did. How many
  double-flips did Tyler do?
Answer: Jen did 16 triple-flips, so she did 16 * 3 = <<16*3=48>>48 flips.
Tyler did half the number of flips, so he did 48 / 2 = <<48/2=24>>24 flips.
A double flip has two flips, so Tyler did 24 / 2 = <<24/2=12>>12 double-flips.
#### 12

Question: Four people in a law firm are planning a party. Mary will buy a platter of
  pasta for $20 and a loaf of bread for $2. Elle and Andrea will split the cost for
  buying 4 cans of soda which cost $1.50 each, and chicken wings for $10. Joe will buy
  a cake that costs $5. How much more will Mary spend than the rest of the firm put
  together?
Answer: Mary will spend $20 + $2 = $<<20+2=22>>22.
Elle and Andrea will spend $1.5 x 4 = $<<1.5*4=6>>6 for the soda.
Elle and Andrea will spend $6 + $10 = $<<6+10=16>>16 for the soda and chicken wings.
Elle, Andrea, and Joe together will spend $16 + $5 = $<<16+5=21>>21.
So, Mary will spend $22 - $21 = $<<22-21=1>>1 more than all of them combined.
#### 1

Question: A charcoal grill burns fifteen coals to ash every twenty minutes of
  grilling. The grill ran for long enough to burn three bags of coals. Each bag of
  coal contains 60 coals. How long did the grill run?
Answer: The grill burned 3 * 60 = <<3*60=180>>180 coals.
It takes 20 minutes to burn 15 coals, so the grill ran for 180 / 15 * 20 =
  <<180/15*20=240>>240 minutes.
#### 240

Question: A bear is preparing to hibernate for the winter and needs to gain 1000
  pounds. At the end of summer, the bear feasts on berries and small woodland animals.
  During autumn, it devours acorns and salmon. It gained a fifth of the weight it
  needed from berries during summer, and during autumn, it gained twice that amount
  from acorns. Salmon made up half of the remaining weight it had needed to gain. How
  many pounds did it gain eating small animals?
Answer: The bear gained 1 / 5 * 1000 = <<1/5*1000=200>>200 pounds from berries.
It gained 2 * 200 = <<2*200=400>>400 pounds from acorns.
It still needed 1000 - 200 - 400 = <<1000-200-400=400>>400 pounds.
Thus, it gained 400 / 2 = <<400/2=200>>200 pounds from salmon.
Therefore, the bear gained 400 - 200 = <<400-200=200>>200 pounds from small animals.
#### 200

Question: Janet's ducks lay 16 eggs per day. She eats three for breakfast every
  morning and bakes muffins for her friends every day with four. She sells the
  remainder at the farmers' market daily for $2 per fresh duck egg. How much in
  dollars does she make every day at the farmers' market?
Answer:
```

Figure 2: The pinned GSM8K prompt for one document (content hash `88a2e5e2`); every method receives this exact string, so the demonstrations no longer depend on the GPU count.

D SECOND-BACKBONE PORT (DREAM-7B)

The Dream-7B results (Table 12) use the identical harness, decoding configuration, and LCD hyperparameters as the LLaDA-8B-Base runs; only two backbone-specific facts change, both mechanical.

Logit shift Dream predicts the token at position i from the logits at position $i-1$ (a shift-by-one), whereas LLaDA-8B-Base reads the logits at position i directly. Applying LLaDA-8B-Base's loop to Dream without accounting for this produces incoherent output even for the no-revision baseline (in our check, GSM8K generations collapsed to whitespace). We add a one-line shift, $\texttt{logits} \leftarrow \text{concat}(\texttt{logits}[:,:1],\ \texttt{logits}[:,:-1])$, at every forward pass (base, verify, and local-dropout), enabled by the flag `DREAM_SHIFT=1`; with it, the no-revision baseline recovers coherent, correct generations. Dream's mask id (151666) is passed through the existing `mask_id` argument. We use the max-softmax transfer rule (Dream's `maskgit_plus` algorithm), matching CORE's LLaDA-8B-Base setup, so both methods share one Dream-appropriate base decoder.

Code-fence extraction Dream formats HumanEval completions as a fenced code block followed by prose; the standard harness scored the whole blob (which does not compile) and reported near-zero pass@1 for *both* methods. Because the emitted code itself is correct, we truncate the completion at the first code fence (flag `STRIP_CODE_FENCE=1`, applied only to the Dream HumanEval cells) so the model's actual code is what gets executed. This is an extraction fix shared identically by LCD and CoRe; it changes the absolute numbers but not their comparison. No other task or backbone uses it.

E The Strict-Superset Property

We use $\text{LCD}_{r=0}$ as a sanity-check ablation: it must reproduce CoRe exactly. The reason is mechanical. At $r = 0$ the neighbor set $\mathcal{N}_0(\mathcal{C}_t)$ is empty, so the local-dropout state of Eq. 2 is $x_t^{\text{lcd}} = x_t^{\text{ver}}$, the identical tensor CoRe already constructs. The two forward passes then return identical logits, so $p_t^{\text{lcd}} = p_t^{\text{ver}}$, hence $S_t^{\text{lcd}} = S_t$ and $x_t^{\text{lcd},(i)} = x_t^{\text{ver},(i)}$ at every i. The confirmation test compares $p_t^{\text{ver}}[x_t^{\text{lcd}}]$ against τ_{conf}; since $x_t^{\text{lcd}} = x_t^{\text{ver}}$ is the verify argmax it always passes, but the additive rule then zeros S_t^{lcd} wherever $S_t > 0$, and on the complement $S_t^{\text{lcd}} = S_t = 0$, so $\max(S_t, S_t^{\text{lcd}}) = S_t$ and no refill source differs from CoRe's. The remask set, the refilled tokens, and the transfer are therefore CoRe's, position by position. In practice we verify this empirically on a small subset: $\text{LCD}_{r=0}$ and CoRe produce identical generations. Because the extra pass at $r = 0$ is wasted compute, $\text{LCD}_{r=0}$ is used only as a correctness check, never as a deployed configuration.

It is worth being exact about what this does and does not show, because the submitted version leaned on it too hard. It shows that the family of configurations LCD defines contains CoRe as the $r = 0$ member, so the comparison at $r = 1$ is a comparison within one implementation rather than across two codebases. It does not show that the deployed configuration preserves CoRe's behaviour at $r = 1$, and it cannot: at $r = 1$ the merged score competes globally for a remask budget of R_t positions, so a gated LCD discovery can displace a position CoRe would have remasked (§4.1). Radius-zero equivalence is a statement about the $r = 0$ member of the family and carries no guarantee about any other member. The budget-preserving variant of §4.1 is the configuration for which a preservation statement can be made, and we report its cost in the gain rather than assuming it is free.

F Implementation Details

The reference implementation, the decoding traces behind every attribution table, and the scripts that regenerate each table are at `https://anonymous.4open.science/r/LCD-iclr27-1EE0/`. Figure 3 gives the full pipeline.

Algorithm 1 LCD: a single verify step (additive + confirmation gates).

Require: x_t, θ, step t, radius r, candidate cap m, remask cap R, thresholds $\tau_{\text{repl}}, \tau_{\text{conf}}$.

$\mathcal{C}_t \leftarrow \text{TopK}_m(-\text{margin} \mid \text{committed, in-block, non-prompt, non-protected})$
$x_t^{\text{ver}} \leftarrow x_t$ with [MASK] at $\mathcal{C}_t$; $p_t^{\text{ver}} \leftarrow \text{softmax}(\theta(x_t^{\text{ver}}))$ ▷ CoRe pass
$S_t \leftarrow \text{CoReScore}(p_t^{\text{ver}}, x_t, \tau_{\text{repl}})$ ▷ Eq. 1
if $r > 0$ **then**
  $\mathcal{C}_t^{\text{lcd}} \leftarrow \mathcal{C}_t \cup \mathcal{N}_r(\mathcal{C}_t)$; $x_t^{\text{lcd}} \leftarrow x_t$ with [MASK] at $\mathcal{C}_t^{\text{lcd}}$
  $p_t^{\text{lcd}} \leftarrow \text{softmax}(\theta(x_t^{\text{lcd}}))$ ▷ local-dropout pass
  $S_t^{\text{lcd}} \leftarrow \text{CoReScore}(p_t^{\text{lcd}}, x_t, \tau_{\text{repl}})$ restricted to $\mathcal{C}_t$
  $S_t^{\text{lcd}}[i] \leftarrow 0$ where $p_t^{\text{ver},(i)}[x_t^{\text{lcd},(i)}] < \tau_{\text{conf}}$ ▷ confirmation
  $S_t^{\text{lcd}}[i] \leftarrow 0$ where $S_t^{(i)} > 0$ ▷ additive-only
  $\tilde{S}_t \leftarrow \max(S_t, S_t^{\text{lcd}})$; refill from the winning pass ▷ Eq. 5
else
  $\tilde{S}_t \leftarrow S_t$
end if
remask top-R positions of $\tilde{S}_t$; caller applies base transfer next

The reference implementation lives entirely inside the `core2` branch of `generate_core2.py` and reuses CoRe's verify routine for both passes. We give the exact operations.

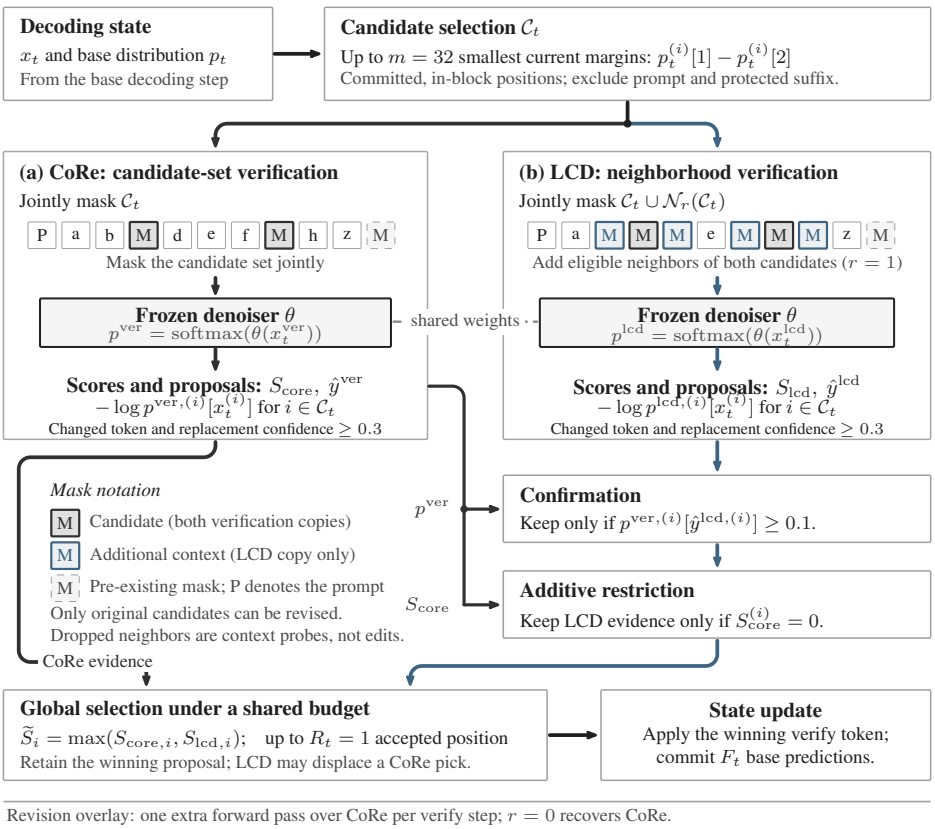

Figure 3: The full LCD pipeline: gray masks are CoRe's candidates, blue the neighbors masked only in the LCD copy; both passes score the candidates, and the gates act before the shared top-R_t selection.

Neighbor mask Let `cand` be the boolean $[B, L]$ mask of candidates ($\mathcal{C}_t$). The radius-r neighbor mask is built by shifting `cand` left and right by each $d \in \{1, \ldots, r\}$ and unioning, then intersecting with the eligibility mask (committed, in-block, non-prompt, non-protected), and finally unioning back the candidates themselves:

neighbor mask, tensor ops

```
neigh = zeros_like(cand)
for d in 1..r:
    neigh[:, :-d] |= cand[:, d:]     # candidate's left neighbor
    neigh[:, d:]  |= cand[:, :-d]    # candidate's right neighbor
verify_mask_lcd = cand | (neigh & eligible)
```

Eligibility is the same predicate CoRe uses to form candidates, so neighbors that are masked, out-of-block, prompt, or protected are never dropped.

Reading evidence at candidates only After the local-dropout forward pass, CoRe's score (Eq. 1) is computed from p_t^{lcd} and then masked to the original candidate set `cand`; positions outside `cand` are set to $-\infty$ so they can never be selected. The dropped neighbors are thus used only to change the context, never themselves remasked by the LCD signal (they remain ordinary committed tokens, eligible for CoRe's usual treatment on later calls).

Confirmation gather The confirmation test (Eq. 3) is a single gather: take the LCD argmax token at each candidate, look up its probability under the CoRe-pass distribution p_t^{ver}, and zero the LCD score where that probability is below τ_{conf}. Both tensors are already resident from the two passes.

Additive mask and combination The additive rule (Eq. 4) zeros the LCD score wherever CoRe's score is finite-positive. The combined score is an elementwise maximum, and the refill token and its probability are selected per position from the winning pass via `where(S_lcd > S_core, lcd_*, core_*)`. The downstream remask-and-refill code is CoRe's, unmodified.

Cost accounting The only added model compute is the one local-dropout forward pass, gated behind $r > 0$ and behind the verify-step condition, so it fires exactly on CoRe's verify steps. Everything else (mask construction, gather, maximum, where) is elementwise on $[B, L]$ or $[B, L, |\mathcal{V}|]$ tensors already in memory.

G Headline, Gate-Ablation and EvalPlus Figures

Figure 4 plots the five-task mean against forward passes. LCD has the highest mean among the near-1× points, above CoRe and the three external reconstructions, and only Lookahead exceeds it, at 3× the passes. Two caveats keep it out of the main text: the 1.06× and 1.13× points are not the same budget, which §4.1 corrects by spending LCD's extra passes on CoRe, and a five-task mean is a weak summary when the deltas are carried by one task (§J.2). Points are exact for CoRe and LCD, whose decoding is greedy; ReMDM and Lookahead sample (Appendix B). Figure 5 shows the path from the naive union to the reported configuration, three of the four factorial corners; the complete 2×2 is Table 2 and the absolute numbers are in Table 9. Figure 6 re-grades the same 164 HumanEval problems under EvalPlus HumanEval+, which adds about 80× more tests per problem; CoRe loses more (0.1402 to 0.1098) than LCD (0.1646 to 0.1402), so the gap widens from +2.44 to +3.05 pp with 5 fixes and 0 breaks ($p = 0.062$), which addresses whether the base suite's tests were too weak and not whether the advantage generalises (§4.1).

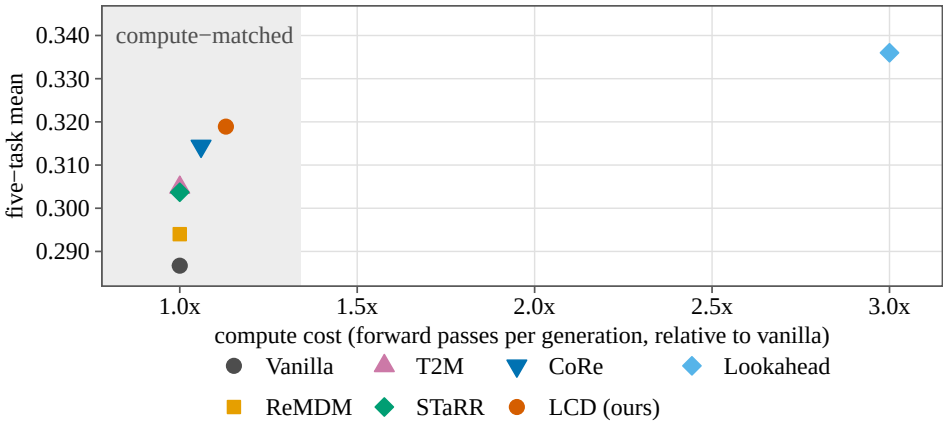

Figure 4: Five-task mean against forward passes per generation, relative to vanilla.

H Full Gate-Ablation Numbers

Figure 5 reports the three discriminative tasks at each ablation step. Table 9 here collects every measured cell, including the tasks completed only for a subset of the configurations. Cells marked "–" were not run at that step; the search was deliberately concentrated on the three tasks that move (the win plus the two regressions), and only the final configuration was run on all five.

A note on the naive configuration's four-task mean. Completing $\text{LCD}_{\text{naive}}$ on MATH gives 0.1734 (+0.12 vs CoRe); with GSM8K −0.91, HumanEval +1.83, and MBPP −0.80, the four-task mean delta is +0.06 pp, a wash. The final configuration's five-task mean delta is +0.44 pp.

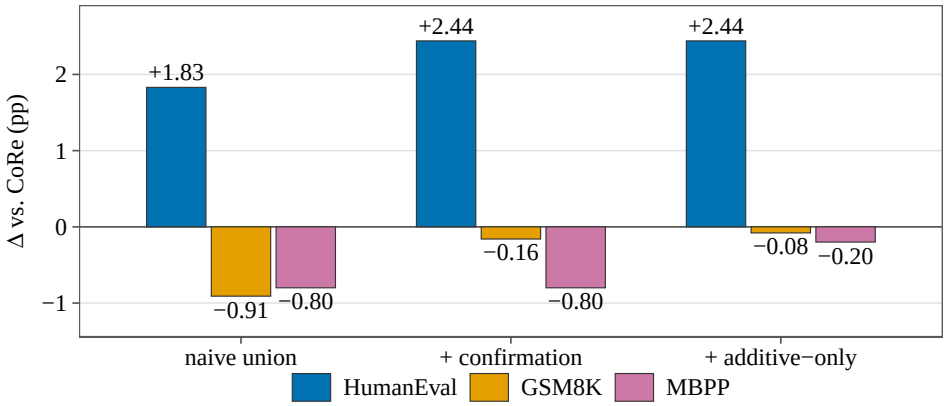

Figure 5: Change versus CoRe at each build step from the naive union to the reported configuration.

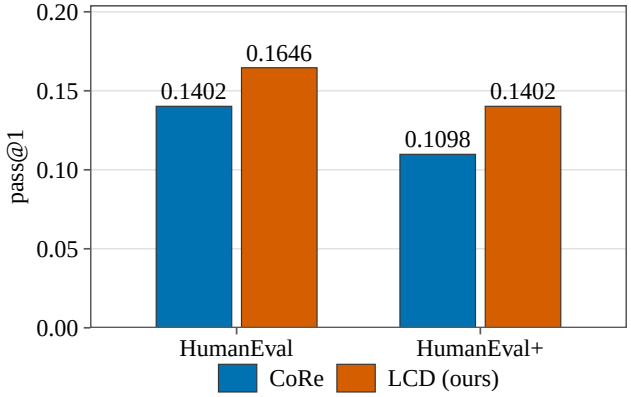

Figure 6: pass@1 under base HumanEval and HumanEval+ for CoRe and LCD.

I Why Each Gate Works, in Detail

Table 2 gives the four factorial corners and Figure 5 the path we took through them. This section gives the reason we believe each gate acts on the task it does, and checks that reason against the measured cells. We separate two questions the submitted version ran together: whether each gate moves the task it was diagnosed for, which the path already showed, and whether each gate is *necessary* given the other, which only the full factorial can answer.

I.1 Where the HumanEval gain comes from

LCD's only improvement outside the small-difference band is HumanEval. The account in Remark 1 is consistent with that, and consistency is not proof; Appendix U tests the account directly and does not support its per-position form, so what follows is confined to what the factorial itself shows. If the HumanEval gain came from LCD overriding CoRe's decisions, forbidding overrides would erase it, and it does not: HumanEval is 0.1646 with and without the additive rule, so the gain lives on positions CoRe's own score left silent. That is a fact about where in the score space the gain sits, and we no longer read it as a signature of a wrong-neighbor mechanism. An alternative we considered is that the gain is noise on 164 problems, and we cannot rule that out. The submitted version argued that the gain growing under EvalPlus's 80× extra tests is something noise would not do. That argument is wrong, and we withdraw it. HumanEval+ re-grades the same 164 problems, so it removes uncertainty about whether a completion is correct, not uncertainty about which problems were drawn; a spurious four-problem advantage on this particular sample can widen under stricter grading just as a real one can. What the EvalPlus result does establish is narrower and still worth reporting: the completions LCD produces on the problems where the two methods differ survive the added tests more often than

Table 9: Every measured cell across configurations, LLaDA-8B-Base seed 42; "–" marks corners not run.

Configuration	GSM8K	MATH	BBH	HumanEval	MBPP
CoRe	0.5262	0.1722	0.4796	0.1402	0.2540
$\text{LCD}_{\text{naive}}$ (r=1, no gates)	0.5171	0.1734	–	0.1585	0.2460
LCD + confirmation only	0.5246	–	–	0.1646	0.2460
LCD + additive only	0.5201	–	–	0.1585	0.2580
LCD + confirmation + additive (reported)	0.5254	0.1728	0.4795	0.1646	0.2520

CoRe's, so whatever the four-problem advantage is, it is not an artifact of the base suite's weak tests. Uncertainty over the problem sample is what the held-out evaluation in §4.1 is for.

I.2 Why the confirmation gate fixes reasoning but not MBPP

The two regressions of the naive union have different causes, and the confirmation gate is the tool that tells them apart. The gate keeps an LCD proposal only if the proposed token also has mass under the full-context distribution. On GSM8K the harmful proposals fail this test. They were confident only because a neighbor was dropped, so restoring the neighbor removes their mass and the gate filters them. The GSM8K regression closes, from -0.91 to -0.15. On MBPP the harmful proposals pass this test, because the model assigns them full-context mass: they are confidently wrong. A confidence test cannot separate a confident-correct edit from a confident-wrong one, so the gate leaves MBPP at -0.80. Raising the confidence threshold would not help either, since the MBPP failure is not a low-confidence problem, and it reproduces, through a different mechanism, the known observation that confidence filters do not catch MBPP's bad edits. The lever that works on MBPP has to be structural, which is what the additive rule provides.

I.3 Why additive-only is safe for the gain

Forbidding overrides could in principle discard good edits, and on HumanEval it does not: adding the additive rule to the confirmation gate leaves the score at 0.1646, and additive-only scores 0.1585, exactly the naive union's value. The additive rule's effect on HumanEval is not small, it is zero. That is what the position-space account predicts: the HumanEval gain lives on CoRe-silent positions, where the additive rule by definition does nothing, while the MBPP harm lived on CoRe-active positions, where the rule blocks the override, so one rule separates them because they never act on the same positions.

Completing the factorial on MBPP then reverses the conclusion we drew from that. Additive-only reaches 0.2580 on MBPP, the best MBPP figure of any configuration here and above CoRe's 0.2540, whereas adding the confirmation gate brings it down to 0.2520. In problem counts, adding confirmation to the additive rule wins one HumanEval problem and loses three MBPP problems, five against seven pooled ($p = 0.774$). So the position-space separation is real, but the pair is not better than the additive rule alone on this evidence, and the submitted version's framing of two gates each earning their place was wrong in a way that completing one cell of the design would have caught.

J Robustness of the HumanEval Result, in Detail

The main text compresses four checks into two paragraphs. Here we give the tables, and for each check what it does and does not establish. Across all four, the direction of the result is stable and its magnitude is not.

J.1 Greedy determinism, the paired test, and the metric each table uses

Greedy decoding ($T = 0$) is deterministic, and we confirmed that seeds $\{42, 1337, 7\}$ produce byte-identical outputs for both LCD and CoRe. Seed averaging therefore carries no information here: the across-seed variance is exactly zero, and a reported mean over seeds would be the single-seed number with a standard error of zero by construction. Determinism of this kind removes decoding

randomness only. It does not remove the sampling uncertainty over which problems a benchmark contains, which is the uncertainty that matters for an effect of a few problems, so we report the paired per-problem difference with both an exact McNemar test and a paired bootstrap interval (Table 10).

One correction to the submitted version. GSM8K is scored under two lm-evaluation-harness filters, `strict-match` and `flexible-extract`, and our headline GSM8K numbers (0.5262 for CoRE, 0.5254 for LCD) are `strict-match`. The paired split we previously reported for GSM8K, 6 fixes against 12 breaks, is the `flexible-extract` split and is inconsistent with that headline: 6 and 12 out of 1319 imply a -0.45 pp change, not the -0.08 pp the headline shows. Under `strict-match` the same runs give 3 fixes and 4 breaks at $p = 1.000$. Every paired number in this paper is now computed on the filter that defines the task's headline metric, and the regeneration script is released with the code.

In Table 10, "fix" is a problem LCD passes and CoRE fails, "break" the converse, p is the exact two-sided McNemar test on the discordant pairs, and the interval is a paired bootstrap 95% CI on the accuracy difference (20,000 resamples). The HumanEval interval should be read with its footnote: with four discordant pairs all in one direction, the bootstrap excludes zero while the exact test cannot, and the exact test is the defensible statement at this size.

Table 10: Paired per-problem outcomes of LCD against CoRE on each task's headline metric.

Task	n	fix	break	McNemar p	Δ (pp)	bootstrap 95% CI
HumanEval	164	4	0	0.125 ✗	+2.44	[+0.61, +4.88] ✓
MBPP	500	3	4	1.000 ✗	−0.20	[−1.20, +0.80] ✗
GSM8K	1319	3	4	1.000 ✗	−0.08	[−0.45, +0.30] ✗
MATH	5000	10	7	0.629 ✗	+0.06	[−0.10, +0.22] ✗
BBH	6511	64	65	1.000 ✗	−0.02	[−0.35, +0.32] ✗
HumanEval+	164	5	0	0.062 ✗	+3.05	[+0.61, +6.10] ✓
MBPP+	378	4	7	0.549 ✗	−0.79	[−2.65, +0.79] ✗

On HumanEval, LCD fixes 4 problems and breaks 0, so every problem CoRE solves LCD also solves. With a 4-vs-0 split, 0.125 is the smallest two-sided p the sample can produce (2×0.5^4), so the test sits at its floor for this sample size rather than above it, and a single 164-problem benchmark cannot establish significance for an effect of four problems. The paired bootstrap interval on the same data excludes zero, but that interval is anti-conservative with four discordant pairs, because every resample that draws any discordant pair draws it in the same direction; we therefore treat $p = 0.125$ as the statement of record and the interval as descriptive. On MBPP the split is 3 vs 4 and on GSM8K 3 vs 4, both ties by the test, and on MATH and BBH the discordant counts are near-balanced. We do not claim any of the four non-code tasks improves.

J.2 What the intervals do and do not license

The submitted version described LCD as having "no task regressing" and treated per-task differences within 0.20 pp on LLaDA-8B-Base and 0.60 pp on Dream as ties. That is not a claim the data support, for two reasons. Failing to detect a decline is not evidence of equivalence, and a tie band chosen after seeing the deltas is not a margin. Three of the five LLaDA-8B-Base deltas are in fact negative (GSM8K -0.08, BBH -0.02, MBPP -0.20), and the five-task mean excluding HumanEval is -0.06 pp, so the positive mean is carried entirely by one task.

To say something testable instead, we fix a noninferiority margin before reading the intervals and use the margin the benchmark's own resolution suggests: one problem on the smallest task we report, HumanEval, is 0.61 pp, and one problem on MBPP is 0.20 pp, so we take $\delta = 1$ pp as a margin that is coarse enough to be meaningful on every task and fine enough to exclude the drops these benchmarks are usually reported to resolve. A task is then noninferior at δ when the lower bound of its paired 95% interval is above $-\delta$. Under that rule four of the five LLaDA-8B-Base tasks pass (GSM8K $[-0.45, +0.30]$, MATH $[-0.10, +0.22]$, BBH $[-0.35, +0.32]$, HumanEval $[+0.61, +4.88]$) and **MBPP does not**: its interval is $[-1.20, +0.80]$, which does not exclude a 1 pp drop. MBPP+ likewise fails, at $[-2.65, +0.79]$.

We therefore state the empirical claim as follows. LCD *improves HumanEval in this configuration* by $+2.44$ pp (4 fixes, 0 breaks, $p = 0.125$; $+3.05$ pp under HumanEval+), *with small observed differences on the other benchmarks*, of which MBPP is the one whose interval still admits a 1 pp decline. Claims of a transferable safety property, and the word "regression-free" as a property of the method rather than of a particular run, are removed throughout.

J.3 Does the gain survive stricter grading

The base HumanEval suite runs few tests per problem, so a completion can pass by handling only the easy cases. EvalPlus adds about $80\times$ more tests, and if LCD's edits were cosmetic the stricter tests would erase the gain. Figure 6 shows they do not. Both methods lose accuracy under the harder tests, CoRe by 3.04 pp ($0.1402 \to 0.1098$) and LCD by 2.44 pp ($0.1646 \to 0.1402$). LCD loses less, so the gap grows from $+2.44$ to $+3.05$, and the paired profile stays 5 fixes and 0 breaks with the exact p tightening from 0.125 to 0.062.

This does not mean LCD is unaffected: some LCD completions that passed the base suite fail the added tests, only fewer of them than CoRe's. That ordering is consistent with the repairs changing program behavior rather than surface form, since a completion that passed only by luck would not survive more tests. We no longer attribute it to a wrong-neighbor mechanism, which Appendix U does not support. MBPP+ ties (-0.79, $p = 0.55$), which matches base MBPP and is expected, since the additive-only rule was built to leave CoRe's MBPP decisions alone.

J.4 Stochastic decoding

Greedy is the standard evaluation mode for this suite and is what the headline uses, but it is one operating point, and under sampling the picture can differ. We therefore run $T = 0.2$ with five seeds (Table 11, Figure 7).

Table 11: Stochastic decoding at $T = 0.2$ over five seeds; Δ is the paired difference in pp with its standard error.

Task	LCD	CoRe	Δ (pp)
HumanEval	0.1476 ± 0.0027	0.1415 ± 0.0027	$+0.61 \pm 0.00$
MBPP	0.2556 ± 0.0077	0.2580 ± 0.0071	-0.24 ± 0.28
GSM8K	0.5187 ± 0.0042	0.5186 ± 0.0032	$+0.02 \pm 0.07$

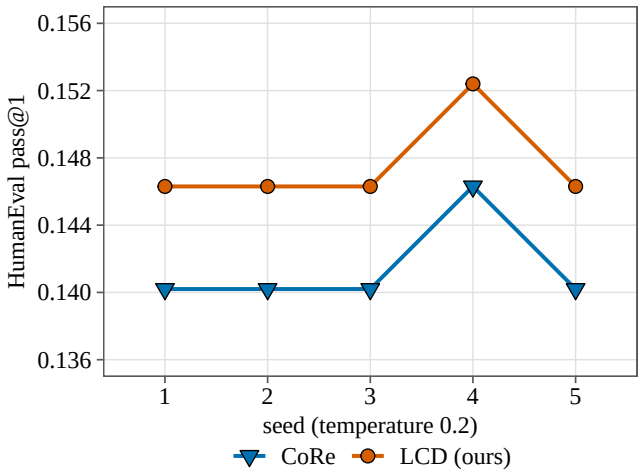

Figure 7: HumanEval pass@1 for CoRe and LCD at five sampling seeds, $T = 0.2$.

The delta has essentially zero standard error: LCD fixes exactly one net problem at each of the five seeds, so the $+0.61$ is stable rather than noisy, even though it is small. It is also much smaller than the greedy $+2.44$, which puts part of the greedy gain down to greedy decoding itself, since sampling

flattens the distribution and reduces the number of confidently-wrong local commitments that LCD is built to catch. We read this as bounding the headline rather than undercutting it, and the claim we defend is that the size of the gain is greedy-specific. The submitted version also claimed that at $T = 0.2$ LCD is never below CoRe on any of the three tasks, which Table 11 contradicts: MBPP is -0.24 ± 0.28 pp. The accurate statement is that HumanEval improves at every seed and the other two tasks show differences smaller than their seed-to-seed spread. We did not sweep temperature, and $+0.61$ could shrink further at a higher one for the same reason; we flag this rather than run the sweep, because the greedy result is the headline and the stochastic run is a supplement.

J.5 Second backbone: Dream-7B

The strongest test of generality available to us is a different model. Dream-7B is a masked diffusion LM with a different architecture and tokenizer. Appendix D documents the two mechanical changes the port needed. We then run LCD against CoRe unchanged (Table 12, Figure 8).

Table 12: LCD against CoRe on Dream-7B, seed 42, greedy, identical configuration; Δ in pp.

Method	GSM8K	MATH	BBH	HumanEval	MBPP	Mean
CoRe	0.4564	0.1792	**0.3990**	0.1768	0.3500	0.3123
LCD (ours)	**0.4579**	0.1790	0.3967	0.1768	**0.3560**	**0.3133**
Δ	+0.15	−0.02	−0.23	+0.00	+0.60	+0.10

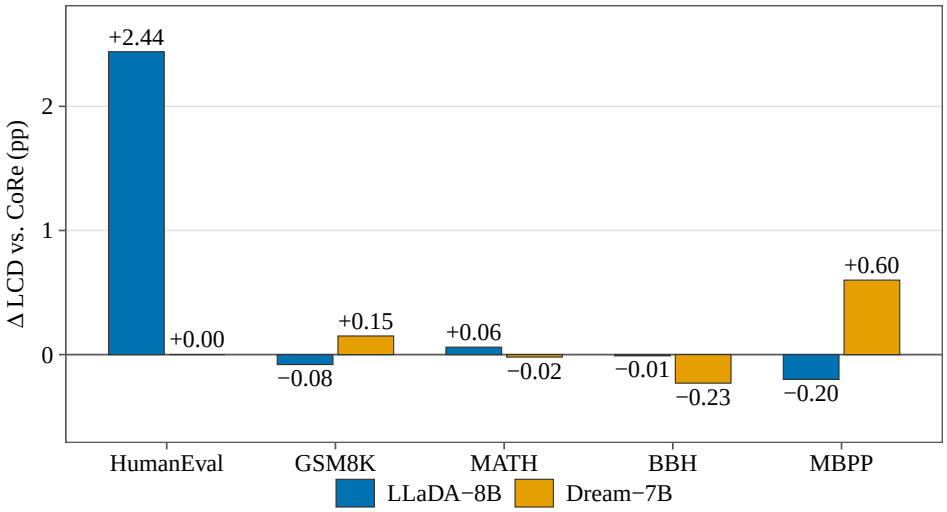

Figure 8: Change versus CoRe per task on LLaDA-8B-Base and Dream-7B.

Both halves of the table hold at once. On Dream every per-task delta is within ± 0.6 pp and the mean is slightly positive, though two tasks are negative (MATH -0.02, BBH -0.23, the latter 65 fixes against 80 breaks), so this is a set of small measured differences rather than an absence of regression. The HumanEval win does not transfer, at $+0.00$ on Dream against $+2.44$ on LLaDA-8B-Base. We previously explained this by saying the pattern LCD targets, a committed token held in place by a confident neighbor, is rarer under Dream's decoding. We withdraw that explanation: Appendix U does not support the per-position account on LLaDA-8B-Base either, so invoking it to explain a backbone difference would be inventing a mechanism to cover a gap. What we can say is narrower. LCD changes the verify context in one specific way, that change produces four repairs on LLaDA-8B-Base HumanEval and none on Dream, and we have no account of why the two backbones differ. The comparison is worth reporting precisely because it is unexplained.

We considered the alternative that the port is wrong and suppresses the mechanism, and rule it out on two checks. LCD's remasking does fire on Dream, since LCD and CoRe disagree on two to three GSM8K and MBPP problems, and the base decode is correct, since vanilla Dream through our

harness matches the model's own decoder on a spot check. The tie is therefore a real property of the backbone rather than a porting artifact, and the transferable claim is correspondingly narrow: on both backbones every per-task difference from CoRe is under 0.6 pp in absolute value, with no single task's interval excluding a small decline, and the measured gain is LLaDA-8B-Base-specific. We do not claim more than that, and in particular we do not claim a safety property that carries from one backbone to another.

K What Transfers From a Null Result

A negative result about one extension is worth little on its own. What we think transfers is a set of checks, each of which caught an error in this paper that we would otherwise have published, and each of which applies to any claim of the form "perturbing X rather than Y is what helps". We state them as the failure they catch, because that is how we encountered them.

A perturbation control must be verified, not asserted Our count-matched control was not count-matched. It sized the replacement set by $|\mathcal{N}_r(\mathcal{C}_t)|$ rather than by the number of positions the method actually adds, and because candidates cluster, the control perturbed about 39% more context than the method it was controlling for (§4.2). The defect is invisible in the aggregate scores and survived our own review of the code. What exposes it is cheap: have the decoder emit the realised count for every condition and compare the totals. We now do, and report the residual mismatch, 16 masks in $53{,}205$ where the distant pool runs dry.

Sufficiency and necessity are different probes, and the cheap one is misleading Removing one position and seeing little change tells you that position is not sufficient. It does not tell you the position is irrelevant, because it may act only in combination with others. Our first instrumentation measured only sufficiency and we drew the stronger conclusion from it; Appendix U reports both probes and they disagree. Any attribution claim over a set of simultaneously perturbed positions needs the leave-one-out probe as well, and the two together are still not a decomposition when interactions are present.

Log the intervention that was applied, not the one that was proposed Our first trace recorded every position the scoring rule favoured, before the revision budget chose among them, and carried no problem identifier. Roughly a quarter of the recorded events never happened, and none could be joined to an outcome. Recording after the budget, with the document id attached, costs nothing and is what makes an event-to-outcome claim checkable.

Tune and report on disjoint problems, and say which is which Our gates were selected against failures on the same benchmarks we then reported. §4.1 is the only evaluation here made after freezing, and it is the one that did not go our way. The asymmetry is the point: every number chosen on is exploratory, and a paper should mark them.

Spend the ablation on the baseline too A method that adds a forward pass should be compared against the baseline given that pass, not only against the baseline as published. Ours is indistinguishable from CoRe with more verify calls (§4.1), which is the result that decides whether the idea is worth anything, and it is one line of configuration to run.

Scripts drift, and a pipeline that scores identical output differently is a defect in the pipeline Two defects in this revision were not in the decoder or the analysis but in the plumbing around them, and both would have produced wrong numbers had they not been caught. The live copy of the Dream driver had lost the code-fence truncation that its released copy and Appendix D carry, so a probe script built from it scored 0.0061 on generations that are byte-identical to the 0.1768 reference. And the table writer keyed its output on task and sampling design but not backbone, so the Dream traces overwrote the LLaDA-8B-Base HumanEval table, the second time in this revision that one trace set silently replaced another's numbers (the first was MBPP over HumanEval). The remedies are mechanical: the release check now refuses any Dream HumanEval script without the flag, and every table file is stamped with the trace directory it was written from, which the writer checks before overwriting. What transfers is the rule that a reproduced score is a precondition for reporting anything derived from a run, and that a table's provenance has to be machine-checked, not remembered.

What we would try next The constraint behind the method still looks real: a verify pass restricted to its own candidate set rarely removes a confidently committed token, and that follows from the margin ranking rather than from any particular decoder. Our way of exploiting it does not pay, because the extra pass costs as much as verifying more often. A version that earned its place would reach those positions without a second pass, for instance by widening the existing verify mask to include some committed non-candidates, which is free in forward passes and can be tested against exactly the controls above.

Table 13: MBPP-Heldout: 464 problems (task ids 511–974) disjoint from the test split and demonstrations, method frozen.

Method	n	pass@1	fix	break	McNemar p	Δ vs. CoRe (pp)
Vanilla (`low_conf.`)	464	0.1940	–	–	–	−10.13
CoRe	464	0.2953	–	–	–	–
Lcd (frozen)	464	0.2866	3	7	0.344 ✗	−0.86 [−2.16, +0.43]

Table 14: Count-matched masking controls; "3/0" means Lcd passes three problems the control fails and none the reverse.

	HumanEval		MBPP		vs. Lcd on HumanEval	
Extra positions masked	pass@1	Δ	pass@1	Δ	Lcd fix/brk	exact p
none (CoRe)	0.1402	–	0.2540	–	4/0	0.125 ✗
candidate's neighbours (Lcd)	0.1646	+2.44	0.2520	−0.20	–	–
random eligible, draw 42	0.1463	+0.61	–	–	3/0	0.250 ✗
random eligible, draw 1337	0.1524	+1.22	–	–	2/0	0.500 ✗
random eligible, draw 7	0.1463	+0.61	–	–	3/0	0.250 ✗
distant eligible	0.1402	+0.00	–	–	4/0	0.125 ✗

L Replicating the Probes on a Second Task

The sufficiency and necessity probes were measured on HumanEval, on the applied interventions of one configuration, and a pattern found that way can be a property of those events rather than of the method. We therefore repeat the whole procedure unchanged on MBPP, a different task with different prompts and five hundred problems, and compare the two.

Table 15: The two probes on HumanEval and MBPP; shares are of the full-mask effect on positive-span events.

Quantity	HumanEval	MBPP
positive-span events with both probes	60	326
distinct problems	41	201
median sufficiency share	+0.08	+0.26
median necessity share	+0.52	+0.64
median within-event necessity minus sufficiency	+0.106	+0.123
sufficient alone, share of applied	18.7%	23.8%
necessary not sufficient, share of applied	17.6%	21.2%

The replication run reproduces the reference MBPP score exactly, 0.2520 on all 500 problems, so the probes do not perturb decoding on this task either. The split survives, on a sample five times larger. On MBPP the probes separate in the same direction and by a similar amount: a neighbor removed alone recovers a median 0.26 of the effect while restoring one inside the full mask removes a median 0.64, against 0.08 and 0.52 on HumanEval. The classification rates are closer still. Interventions in which a neighbor is necessary without being sufficient occur on 21.2% of applied interventions on MBPP against 17.6% on HumanEval, and interventions with a sufficient neighbor on 23.8% against 18.7%. Both rates are computed at the same threshold on 326 positive-span events across 201 MBPP problems and 60 across 41 HumanEval problems.

One claim we made from these numbers was wrong and we withdraw it. Comparing marginal medians, we said the gap between the probes was much narrower on MBPP. A difference of medians is not the median of differences, and the quantity the claim needs is the within-event gap: computed correctly it is 0.106 on HumanEval and 0.123 on MBPP, so marginally *larger* on MBPP. What differs between the tasks is the level of both probes, not the separation between them. We are likewise careful about the rates: 17.6% against 21.2% is consistent with a similar rate, which is all we claim, and two points do not establish invariance. Jointness is higher on MBPP as well (0.29 against 0.10), still well below necessity.

What this establishes and what it does not. It establishes that the divergence between the two probes is not an artifact of the HumanEval events it was found on: it reproduces on a different task, with different prompts, at a similar rate, under the same threshold. It does not establish that the neighbor's contribution is what made any program correct, since the probes measure the probability assigned to a proposed token and not the outcome of the program. The two-probe divergence has been measured on one backbone; the adjacency comparison of §4.2 has also been run on a second, below.

M PRIORITY, STATED IN FULL

The related-work paragraph of §1 compresses two concessions that the submitted version of this paper got wrong. Both are stated here at length, because a compressed concession is easy to read as a smaller one than it is.

The perturbation idea is CoRe's, not ours CoRe is explicit that its verify pass is a perturbation test for conditional dependence, describing its score as a "load-bearing test that identifies tokens tied to hard consistency constraints (e.g., variable bindings, closing brackets)" (Zhai et al., 2026). The premise that a wrongly committed token distorts other predictions, and that masking removes a misleading token from the context, is therefore CoRe's and T2M's (Yao, 2026), and the submitted version claimed it as a contribution. What is ours is the perturbed set: CoRe masks the candidate set, whose membership is decided by low margin, while LCD masks each candidate's neighborhood, which deliberately includes high-margin positions the candidate rule excludes by construction. The distinction exists because the selection criterion and the suspected source of the error are in tension: a neighbor that confidently holds a wrong token in place is, for that reason, usually ranked out of the set. Alongside it we contribute the gating construction and the instrumentation. The local neighborhood is loosely analogous to dropping co-adapted units (Gal & Ghahramani, 2016), an analogy that survives at the level of the set (§4.2) but not of an individual unit and its neighbor.

Sufficiency and necessity are established attribution concepts Pairing a removal probe with a restoration probe is standard. Rationale evaluation contrasts comprehensiveness with sufficiency in exactly this shape (DeYoung et al., 2020), and the divergence between the two under feature interactions is characterised in the explanation literature (Watson et al., 2021; Covert et al., 2021). We do not claim the probes, the concepts, or the observation that they can disagree. What §4.2 contributes is the measurement inside a margin-ranked verify pass of a diffusion decoder, the finding that the disagreement is organised by position relative to the candidate, and the confidence-balanced probe of §4.2 that bounds how far that organisation can be read as spatial. A reader who regards the decoding setting as the only new ingredient is reading us correctly.

N THE ADJACENCY/CONFIDENCE CONFOUND, IN DETAIL

§4.2 reports that the adjacent-position advantage survives confidence matching on MBPP and not on HumanEval. This appendix gives the sampling procedure that entangles the two variables, the stratifications behind that claim, what the inference does and does not assume, and what the design still cannot control.

Where the confound comes from The probe set is built from $\mathrm{extra} = x_t^{\mathrm{lcd}} \setminus x_t^{\mathrm{ver}}$, the positions LCD masks beyond CoRe's. For a candidate at p the adjacent probes are the members of extra within r of p; the non-adjacent probes are $k = 2$ positions drawn from extra at distance greater than r from p. Both classes are therefore neighbors of *some* candidate, which is what makes the comparison

meaningful, but only the first class is adjacent to an intervention that was actually applied. Applied interventions sit in locally uncertain regions, and commit confidence is spatially autocorrelated, so a position next to one is systematically less settled than an extra-masked position elsewhere in the block. The median absolute offset is 1 for the adjacent probes against 81.5 for the non-adjacent ones on HumanEval, and the confidence gap follows from that geometry rather than from any choice we made in the analysis.

What the tests assume Probes are nested inside events, events inside problems, and the two tests we substitute for the pooled correlation are exact for different parts of that structure, not for all of it. The permutation test conditions on each event's own composition, shuffling only which of that event's probes are labelled adjacent, and is exact under exchangeability within event; it is also *one-sided*, reporting $\Pr[\text{statistic(permuted)} \geq \text{statistic(observed)}]$, since the hypothesis under test throughout this paper is the directional one, adjacent exceeds non-adjacent, not merely that the two differ. A two-sided count is available and is within a factor of two of the one-sided figure in every cell we report. What within-event exchangeability does *not* license is treating different events from the same problem as independent draws: two events on one HumanEval solution can share whatever made that solution's mask distinctive, and permuting labels separately within each event does not resample across that shared structure. The bootstrap resamples *problems* with replacement and so reflects between-problem variation in its interval. It does not, however, repair the other two tests. The paired sign test and the permutation test both treat events as independent, and reporting a problem-level interval beside them does not make their p-values exact under clustering; it supplies a separate estimate of uncertainty at the problem level, which is the one that generalises across problems. So the three quantities in each cell of Table 4 answer different questions and carry different assumptions: the sign test and permutation test are event-level and assume event independence, the bootstrap interval is problem-level and does not. A reader who wants a single test that is valid under the full nesting will not find one here; the closest thing is the bootstrap interval, and it is an interval rather than a test. The intervals in this version are also corrected ones. An earlier implementation merged every draw of a resampled problem under that problem's original event keys before computing the within-event differences, and because the within-event mean is invariant to duplicating an event's own records, a problem drawn three times entered a replicate once. That discards the multiplicity the bootstrap depends on and understates its variance; a reviewer caught it. The corrected intervals are wider at both ends in every cell, by up to 0.09 at the lower end, and all four still exclude zero. The resampling now sorts events and problems before drawing, so a reported interval is a function of the data and the seed alone and not of the order in which trace files were read; without this, byte-identical traces from two runs gave limits differing in the third decimal. We report the pooled Spearman coefficients as descriptive summaries and do not use their p-values; on the events carrying both classes they are $+0.223$ for adjacency against -0.163 for confidence on HumanEval, and $+0.293$ against -0.057 on MBPP.

The confidence match is close, not exact The balanced design draws each non-adjacent probe as the closest available match in the far pool at that step, which need not be an exact match: the pool is a finite set of already-masked positions, not a continuum. On the pairs it actually constructs, the mean absolute confidence gap is 0.024 on HumanEval and 0.021 on MBPP (medians 0.008 on both), against 0.338 and 0.270 on the same index-paired probes under uniform sampling (medians 0.297 and 0.229). The balanced design closes most of the gap and leaves a residual of about two hundredths, which is small relative to the confidence scale but not nothing, and we report it rather than the looser "achieved" framing an earlier draft used.

Balance in aggregate is not balance within event A per-event gap that averages 0.02 still has a sign, and an advantage that appears only when the adjacent probe is the less confident of the pair is a confidence effect. So for the balanced design we split events by which probe is the more confident and also pair each event's closest-confidence probes under a tolerance (Table 16). On MBPP the advantage is present in both strata (89 against 35 where the adjacent probe is the more confident, 106 against 32 where it is the less) and on closest pairs within 0.05 (200 against 82, $p < 10^{-4}$). On HumanEval it is not: it is carried by the events where the adjacent probe is the less confident (23 against 5, $p = 0.0009$) and absent where it is the more confident (12 against 9, $p = 0.66$), and closest pairs give 33 against 19 ($p = 0.070$) and 31 against 19 ($p = 0.12$). The balanced HumanEval result in Table 4, 38 against 16, is therefore not separable from residual within-event confidence differences;

the MBPP result is, and so are GSM8K's (171 against 78 and 209 against 93 in the two strata; closest pairs 374 against 223, $p < 10^{-4}$; 692 events, pinned few-shot prompts as in Appendix C) and Dream's MBPP result (closest pairs 113 against 65, $p = 0.0004$; the adjacent-more-confident stratum 41 against 26, $p = 0.086$); Dream's HumanEval result holds in both strata (25 against 7 and 33 against 7) with closest pairs at $p = 0.032$ and 0.064. This is the same asymmetry every other strict test has shown, and we state it as a limit on the HumanEval evidence rather than explain it away.

Table 16: Within-event confidence checks: per-event confidence gap, sign tests by which class is more confident, and closest-confidence pairs.

			mean $\lvert\Delta c\rvert$	by which is more confident		closest-confidence pair	
cell	comparison	events	per event	A more	A less	$\lvert\Delta c\rvert \le 0.10$	$\lvert\Delta c\rvert \le 0.05$
LLaDA HE, r=1 balanced	adjacent vs far	60	0.019	12/9 (p=0.6636)✗	23/5 (p=0.0009)✓	33/19 (p=0.0704)✗	31/19 (p=0.1189)✗
LLaDA MBPP, r=1 balanced	adjacent vs far	326	0.020	89/35 ($p<10^{-4}$)✓	106/32 ($p<10^{-4}$)✓	205/90 ($p<10^{-4}$)✓	200/82 ($p<10^{-4}$)✓
LLaDA GSM8K, r=1 balanced	adjacent vs far	692	0.021	171/78 ($p<10^{-4}$)✓	209/93 ($p<10^{-4}$)✓	391/239 ($p<10^{-4}$)✓	374/223 ($p<10^{-4}$)✓
Dream HE, r=1 balanced	adjacent vs far	76	0.025	25/7 (p=0.0021)✓	33/7 ($p<10^{-4}$)✓	45/26 (p=0.0319)✓	41/25 (p=0.0640)✗
Dream MBPP, r=1 balanced	adjacent vs far	193	0.020	41/26 (p=0.0864)✗	77/20 ($p<10^{-4}$)✓	115/67 (p=0.0005)✓	113/65 (p=0.0004)✓
LLaDA HE, r=2	offset 1 vs offset 2	50	0.163	8/11 (p=0.6476)✗	13/15 (p=0.8506)✗	16/18 (p=0.8642)✗	13/17 (p=0.5847)✗
LLaDA MBPP, r=2	offset 1 vs offset 2	337	0.222	105/56 (p=0.0001)✓	87/80 (p=0.6426)✗	102/89 (p=0.3853)✗	80/65 (p=0.2449)✗
LLaDA HE, r=2 balanced	offset 2 vs far	59	0.071	17/14 (p=0.7201)✗	16/7 (p=0.0931)✗	36/22 (p=0.0869)✗	35/20 (p=0.0581)✗
LLaDA MBPP, r=2 balanced	offset 2 vs far	407	0.098	108/73 (p=0.0113)✓	119/90 (p=0.0525)✗	228/156 (p=0.0003)✓	226/151 (p=0.0001)✓
LLaDA HE, r=2 balanced	near (1 or 2) vs far	68	0.016	12/8 (p=0.5034)✗	26/8 (p=0.0029)✓	42/23 (p=0.0248)✓	42/21 (p=0.0111)✓
LLaDA MBPP, r=2 balanced	near (1 or 2) vs far	421	0.015	122/49 ($p<10^{-4}$)✓	153/52 ($p<10^{-4}$)✓	257/154 ($p<10^{-4}$)✓	256/154 ($p<10^{-4}$)✓

Balancing confidence also changes probe count The two designs differ in a second respect. `LCD_PROBE_FAR=2` draws exactly two non-adjacent probes per event under uniform sampling regardless of how many adjacent probes that event has, while the balanced design draws one non-adjacent probe per adjacent probe. Events with a single adjacent neighbor are 1-versus-2 under uniform sampling and 1-versus-1 under balanced sampling, so a difference between the two designs on such an event is not attributable to confidence alone. Only events with two adjacent neighbors give 2-versus-2 under both designs; restricting to that subset (indexed by the shared decoding trajectory, since far-probe sampling does not change which tokens are remasked) holds probe count fixed while confidence balance still varies between the two runs. On this count-matched subset, HumanEval gives 29 events with a within-event median of $+0.225$ under uniform sampling (20 against 8, $p = 0.036$) and $+0.101$ under balanced sampling (19 against 8, $p = 0.052$); MBPP gives 175 events with $+0.354$ under uniform (142 against 33, $p < 10^{-4}$) and $+0.324$ under balanced (142 against 30, $p < 10^{-4}$). What this subset establishes is a shared positive direction under both designs once probe count is held fixed, not equivalence: on HumanEval the median difference falls from $+0.225$ to $+0.101$ and the balanced p sits at 0.052, so the two designs are not interchangeable there, and on 29 events the subset cannot say how much of that drop is confidence balance and how much is noise. It is evidence against probe count being the whole of what separates uniform from balanced sampling in the main comparison, and no more than that; a design that varies confidence balance continuously while holding count fixed throughout, which we did not run, is what would settle it.

The stratifications Table 17 gives all four cells. Under uniform sampling on HumanEval the adjacent median runs $+0.460$, $+0.101$, $+0.039$ from the least- to the most-confident tercile against $+0.020$, $+0.000$, $+0.031$ for the non-adjacent probes, so the gap closes as confidence rises and the two classes are indistinguishable at the top. That is what decides how much weight the post-hoc matched comparison can bear there: matching preferentially retains high-confidence adjacent probes, which on HumanEval is precisely where no effect exists. Nothing else behaves that way. Uniform MBPP runs $+0.193$, $+0.145$, $+0.304$ against $+0.000$ throughout; confidence-balanced HumanEval runs $+0.358$, $+0.162$, $+0.026$ against $+0.028$, $+0.069$, $+0.004$; confidence-balanced MBPP runs $+0.179$, $+0.138$, $+0.374$ against $+0.005$, $+0.000$, $+0.000$. In each of those three the adjacent median leads in every tercile. The near-equal n per tercile in the balanced cells is the balance working: both classes are drawn into the same confidence range by construction.

Token class The named confound we can test is token identity, coarsely. Classing each probe's token as whitespace, punctuation, or word (Table 18) shows that composition differs between the probe classes on MBPP and on Dream, where adjacent probes are more often whitespace on the former and more often words on the latter, but barely on LLaDA-8B-Base HumanEval. Within every class of every cell, twenty-seven comparisons, the adjacent median exceeds the non-adjacent one, with one exception: whitespace tokens on GSM8K tie at 0.000, and their like-class pairs run 22 against 33 ($p = 0.18$), while punctuation and words there lead clearly (30 against 9; 224 against 105).

Table 17: Necessity share by commit-confidence tercile in the four LLaDA-8B-Base task/design cells.

HumanEval, uniform

Probe	adjacent	distant	within-event	exact p
sufficiency	+0.081	+0.022	+0.023	0.1337 ✗
necessity	+0.517	+0.081	+0.233	0.0015 ✓
median necessity share, by commit-confidence tercile (209 probes)				
low ($\leq$ 0.66)	+0.460	+0.020	$n = 42$ vs. 28	
mid (0.66–0.98)	+0.101	+0.000	$n = 29$ vs. 43	
high ($>$ 0.98)	+0.039	+0.031	$n = 18$ vs. 49	

HumanEval, confidence-balanced

Probe	adjacent	distant	within-event	exact p
sufficiency	+0.081	+0.005	+0.010	0.3497 ✗
necessity	+0.517	+0.074	+0.390	0.0038 ✓
median necessity share, by commit-confidence tercile (178 probes)				
low ($\leq$ 0.51)	+0.358	+0.028	$n = 28$ vs. 32	
mid (0.51–0.92)	+0.162	+0.069	$n = 32$ vs. 27	
high ($>$ 0.92)	+0.026	+0.004	$n = 29$ vs. 30	

MBPP, uniform

Probe	adjacent	distant	within-event	exact p
sufficiency	+0.259	+0.014	+0.190	0.0000 ✓
necessity	+0.643	+0.027	+0.482	0.0000 ✓
median necessity share, by commit-confidence tercile (1153 probes)				
low ($\leq$ 0.69)	+0.193	+0.000	$n = 232$ vs. 153	
mid (0.69–0.98)	+0.145	+0.000	$n = 162$ vs. 227	
high ($>$ 0.98)	+0.304	+0.000	$n = 107$ vs. 272	

MBPP, confidence-balanced

Probe	adjacent	distant	within-event	exact p
sufficiency	+0.259	+0.018	+0.200	0.0000 ✓
necessity	+0.643	+0.010	+0.461	0.0000 ✓
median necessity share, by commit-confidence tercile (1002 probes)				
low ($\leq$ 0.56)	+0.179	+0.005	$n = 170$ vs. 166	
mid (0.56–0.92)	+0.138	+0.000	$n = 167$ vs. 169	
high ($>$ 0.92)	+0.374	+0.000	$n = 164$ vs. 166	

Table 18: The adjacency comparison within coarse token class: composition, within-class medians, and the like-class paired test.

	composition, adj. / non-adj.			median necessity, adj. / non-adj.			like-class pairs within event			
cell	space	punct	word	space	punct	word	pairs	median	+/−	p
LLaDA HE uniform	0.31/0.33	0.21/0.25	0.47/0.42	+0.23/+0.00	+0.21/+0.04	+0.05/+0.00	34	+0.000	16/13	0.711 ✗
LLaDA HE balanced	0.31/0.31	0.21/0.27	0.47/0.42	+0.23/+0.00	+0.21/+0.10	+0.05/+0.00	36	+0.037	20/13	0.296 ✗
LLaDA MBPP uniform	0.27/0.14	0.23/0.22	0.49/0.65	+0.20/+0.00	+0.33/+0.00	+0.15/+0.00	246	+0.125	161/81	$< 10^{-4}$ ✓
LLaDA MBPP balanced	0.27/0.18	0.23/0.23	0.49/0.59	+0.20/+0.00	+0.33/+0.00	+0.15/+0.00	208	+0.154	134/66	$< 10^{-4}$ ✓
LLaDA GSM8K balanced	0.21/0.22	0.19/0.19	0.61/0.59	+0.00/+0.00	+0.29/+0.00	+0.20/+0.00	455	+0.111	276/147	$< 10^{-4}$ ✓
Dream HE uniform	0.16/0.28	0.16/0.27	0.67/0.45	+0.22/+0.00	+0.04/+0.00	+0.06/+0.00	56	+0.066	34/18	0.036 ✓
Dream HE balanced	0.16/0.20	0.16/0.22	0.67/0.57	+0.22/+0.00	+0.04/+0.00	+0.06/+0.00	45	+0.000	21/19	0.875 ✗
Dream MBPP uniform	0.15/0.14	0.22/0.26	0.63/0.60	+0.21/+0.00	+0.06/+0.00	+0.35/+0.00	144	+0.219	85/51	0.004 ✓
Dream MBPP balanced	0.15/0.17	0.22/0.27	0.63/0.56	+0.21/+0.02	+0.06/+0.01	+0.35/+0.00	134	+0.040	74/53	0.076 ✗

The strictest reading pairs each event's adjacent and non-adjacent probes only when they share a class. It is significant on LLaDA-8B-Base MBPP under both designs (246 and 208 pairs, $p < 10^{-4}$), on GSM8K (455 pairs, 276 against 147, $p < 10^{-4}$) and on Dream under uniform sampling on both tasks (56 pairs, $p = 0.036$; 144 pairs, $p = 0.0045$), and null on LLaDA-8B-Base HumanEval under both

designs (34 and 36 pairs, $p = 0.71$ and 0.30), on Dream HumanEval balanced (45 pairs, 21 against 19, $p = 0.87$) and on Dream MBPP balanced (134 pairs, 74 against 53, median $+0.040$, $p = 0.076$). The nulls are nulls, and one of them is well powered. Against LLaDA-8B-Base MBPP's like-class effect, a sign proportion of 0.67, the four small cells have power 0.34, 0.45, 0.55 and 0.66 at their non-tied counts and observed proportions of 0.55, 0.61, 0.53 and 0.65, consistent with that effect and with none; the Dream MBPP balanced cell has power 0.96 at its 127 non-tied pairs and an observed proportion of 0.58, so under the strictest condition, like class and balanced confidence together, the advantage is significant on LLaDA-8B-Base MBPP and GSM8K and not on Dream MBPP. The like-class test does not touch syntactic role, which a three-way class cannot represent.

An untested hypothesis for the shape of the effect The verify pass ranks candidates by margin, so among the positions a wider mask reaches, a candidate's own neighbors are the least settled and the likeliest to interact with it. This predicts a different profile under a different selection rule, which we did not run, and it is why we do not read the result as evidence about diffusion decoding in general.

What is still uncontrolled Holding commit confidence fixed does not make the two probe classes exchangeable, and we do not claim it isolates distance. A position adjacent to an applied intervention differs from a distant extra-masked position in token identity, in syntactic role, in how often it is whitespace or a delimiter, and in whatever else co-varies with sitting beside a position the decoder just revised. Confidence is the one such property we can measure and balance here; others we cannot, and a design that balanced them would have to match on token-level covariates at mask time rather than on confidence alone. The supported statement is that offset from the candidate carries information about necessity beyond what commit confidence carries, on the task where we can test it at adequate power.

Reproducing these numbers `rev_locality.py` emits every figure in this appendix from the trace files: `confound()` for the stratifications, correlations, permutation test and bootstrap, and `emit_both()` for Table 4. Each task writes its own table file, keyed by task name, after an earlier version wrote a fixed filename and let one task's traces silently overwrite the other's numbers.

N.1 The adjacency comparison on a second backbone

Every adjacency number in the main text comes from LLaDA-8B-Base. We repeated the comparison, under both sampling designs, on Dream-7B, the second backbone of Appendix D, on HumanEval and on MBPP, with the same LCD configuration and the same probe instrumentation. Both runs reproduce the July reference score exactly, 0.1768 on $164/164$, with byte-identical generations. A first pair of runs omitted the code-fence truncation of Appendix D and scored 0.0061 on those same generations; their traces are identical to the corrected pair's, since far-probe sampling does not touch the decoding trajectory, and Appendix K records the defect. Dream applies LCD on 100 events, 76 of which carry both probe classes, across 58 problems.

Table 19: The adjacency comparison on Dream-7B, HumanEval and MBPP; same layout and tests as Table 4.

	Dream-7B HumanEval uniform		Dream-7B HumanEval balanced		Dream-7B MBPP uniform		Dream-7B MBPP balanced	
	value	p	value	p	value	p	value	p
necessity share, adjacent vs. non-adjacent								
within-event difference (sign test)	$+0.789$	$< 10^{-4}$ ✓	$+0.778$	$< 10^{-4}$ ✓	$+0.513$	$< 10^{-4}$ ✓	$+0.477$	$< 10^{-4}$ ✓
within-event permutation	$+0.460$	$< 10^{-4}$ ✓	$+0.471$	$< 10^{-4}$ ✓	$+0.342$	$< 10^{-4}$ ✓	$+0.287$	$< 10^{-4}$ ✓
cluster bootstrap 95% CI	$[+0.343, +0.526]$ ✓		$[+0.323, +0.526]$ ✓		$[+0.258, +0.421]$ ✓		$[+0.157, +0.352]$ ✓	
sufficiency, within-event difference	$+0.258$	$< 10^{-4}$ ✓	$+0.385$	$< 10^{-4}$ ✓	$+0.177$	$< 10^{-4}$ ✓	$+0.163$	$< 10^{-4}$ ✓
events / problems / probes	76 / 58 / 274		76 / 58 / 244		193 / 138 / 682		193 / 138 / 592	
median commit confidence (adj. / non-adj.)	0.541 / 0.922		0.541 / 0.557		0.693 / 0.961		0.693 / 0.688	

The advantage is larger on Dream than on LLaDA-8B-Base and survives everything it survived there (Table 19). Restoring an adjacent position removes a median 0.787 of the effect against 0.009 for a non-adjacent one; paired within event the adjacent probe is larger on 59 events against 16 ($p < 10^{-4}$), the permutation statistic is $+0.460$ ($p < 10^{-4}$) and the problem-level interval $[+0.343, +0.526]$. The confidence confound is present in the same form (0.541 against 0.922), and balancing it at mask

time changes nothing that matters: 60 against 15, permutation $+0.471$, interval $[+0.323, +0.526]$, the adjacent median leading in every confidence tercile ($+0.019$, $+0.051$, $+0.320$ against $+0.000$ throughout). Isolated removal is significant under both designs ($p < 10^{-4}$), as on MBPP and unlike LLaDA-8B-Base HumanEval. On the count-matched 2-versus-2 subset (46 events) the two designs give $+0.460$ (36 against 9) and $+0.465$ (38 against 7), both $p \leq 10^{-4}$. Post-hoc matching, which produced the null on LLaDA-8B-Base HumanEval, is positive here under both designs (55 pairs, $+0.313$, $p = 0.0008$; 76 pairs, $+0.081$, $p = 0.025$).

MBPP on Dream is the powered replication, and it reproduces the reference score exactly (0.3560 on $500/500$ under both designs). Dream applies LCD on 283 events, 193 carrying both probe classes, across 138 problems. Restoring an adjacent position removes a median 0.642 of the effect against 0.011 for a non-adjacent one (133 against 52, $p < 10^{-4}$); balancing confidence at mask time achieves 0.693 against 0.688 and leaves the result in place (137 against 47; permutation $+0.287$, $p < 10^{-4}$; bootstrap $[+0.157, +0.352]$; the adjacent median leads in every tercile). It passes the within-event checks of Table 16 in the form LLaDA-8B-Base MBPP passes them and LLaDA-8B-Base HumanEval does not: closest-confidence pairs within 0.05 give 113 against 65 ($p = 0.0004$), the stratum where the adjacent probe is the less confident gives 77 against 20, and the stratum where it is the more confident is positive but short of significance (41 against 26, $p = 0.086$). Isolated removal is significant under both designs, and the count-matched 2-versus-2 subset (103 events) gives 81 against 21 and 80 against 22.

What this adds and does not add. The adjacency result is now replicated at power on two backbones on MBPP, robust to within-event confidence on both, and on neither backbone at power on HumanEval. The two-probe divergence and the coarse token-class check are reported for Dream in Appendix N, where the strictest like-class test under balanced sampling is null on Dream MBPP at good power. It does not change the reading of §4.2: adjacency and commit confidence are entangled on Dream exactly as on LLaDA-8B-Base, and syntactic role remains uncontrolled. It is one configuration throughout.

N.2 Distance within the mask: radius 2

Everything above compares positions inside the masked neighborhood with positions outside it, which cannot say whether distance matters *within* the neighborhood. A wider mask can. At $r = 2$ LCD masks offsets ± 1 and ± 2 around each candidate, so the necessity probe can be read at two distances inside one event and compared with far positions ($|o| > 2$) drawn from the same mask. Two readings are distinguishable: a monotone decline with offset, which is the dose-response pattern an effect of proximity would produce, and a step at the neighborhood boundary with offsets 1 and 2 alike, which is what membership in an applied intervention's mask would produce on its own. We ran $r = 2$ on LLaDA-8B-Base on both tasks under both far-sampling designs, with four far probes per event so near and far counts match. On the utility side the change is inert: $r = 2$ reproduces every one of the 164 HumanEval outcomes of $r = 1$ (the same 27 passes, no discordant problem), so the four repairs of §4 are not radius-specific. Table 20 gives the probes.

The near probes are identical across the two far-sampling designs, since only the far draw differs, so the 1-versus-2 column of Table 20 repeats.

Table 20: Necessity by offset inside a radius-2 mask, with commit confidence and within-event sign tests at each offset.

	median necessity			median commit conf.			within-event, sign test		
cell	$\|o\|=1$	$\|o\|=2$	far	$\|o\|=1$	$\|o\|=2$	far	1 vs 2	2 vs far	1 vs far
HumanEval uniform	$+0.154$	$+0.046$	$+0.000$	0.756	0.875	0.961	-0.035 (21/28, p=0.3916) ✗	$+0.058$ (35/23, p=0.1480) ✗	$+0.108$ (33/25, p=0.3581) ✗
HumanEval balanced	$+0.154$	$+0.046$	$+0.000$	0.756	0.875	0.895	-0.035 (21/28, p=0.3916) ✗	$+0.083$ (35/23, p=0.1480) ✗	$+0.114$ (38/19, p=0.0163) ✓
MBPP uniform	$+0.154$	$+0.019$	$+0.000$	0.734	0.781	0.961	$+0.070$ (194/139, p=0.0030) ✓	$+0.087$ (239/164, p=0.0002) ✓	$+0.220$ (233/115, $p<10^{-4}$) ✓
MBPP balanced	$+0.154$	$+0.019$	$+0.006$	0.734	0.781	0.797	$+0.070$ (194/139, p=0.0030) ✓	$+0.066$ (234/168, p=0.0012) ✓	$+0.214$ (234/116, $p<10^{-4}$) ✓

MBPP is the task with power, and its aggregate reading is a gradient: on 452 events with a positive span, median necessity is $+0.154$ at $|o| = 1$, $+0.019$ at $|o| = 2$ and 0.000 far, and all three within-event comparisons are significant in that order (offset 1 over 2 on 194 events against 139, $p = 0.0030$; 2 over far on 239 against 164, $p = 0.0002$; 1 over far on 233 against 115, $p < 10^{-4}$). An earlier draft read the first of these as a comparison at matched confidence because the two offsets have the same mean confidence across events, 0.699 and 0.700. That was wrong, and a reviewer caught it:

equal means across events are not a match within them. The per-event gap between the two offsets' mean confidences averages 0.222 in absolute value, and the necessity difference follows it (Spearman $+0.23$ across events). Split by which offset is the more confident (Table 16), offset 1 beats offset 2 where offset 1 is the more confident (105 against 56, $p = 0.0001$) and not where it is the less (87 against 80, $p = 0.64$); on closest-confidence pairs it is null within 0.10 (102 against 89) and within 0.05 (80 against 65), and reaches 58 against 37 ($p = 0.040$) only within 0.02, on pairs whose median confidence is 0.98. Distance within the mask is therefore not separable from confidence in these data, and the gradient reading is withdrawn.

What the wider mask does establish is that membership extends to offset 2. Offset-2 positions are more necessary than far positions drawn to match them in confidence (balanced far sampling, far median 0.797 against 0.781): 234 against 168 ($p = 0.0012$) overall, in both confidence strata (108 against 73, $p = 0.011$; 119 against 90, $p = 0.053$) and on closest pairs within 0.05 (226 against 151, $p = 0.0001$). Near positions taken together, offsets 1 and 2, exceed far in both strata and on closest pairs (256 against 154, $p < 10^{-4}$), with a per-event confidence gap averaging 0.015. The near probes are identical across the two far-sampling designs on $466/466$ events, and the utility side is inert: $r = 2$ reproduces every one of the 164 HumanEval outcomes of $r = 1$ and scores 0.2460 against 0.2520 on MBPP (491 of 500 identical, six against three, $p = 0.51$).

HumanEval does not reach any of this. Its marginal medians decline with offset ($+0.154$, $+0.046$, 0.000) as does confidence (0.756, 0.875, 0.961); offset 1 versus 2 is null within event (21 against 28); offset 2 versus far is marginal on closest pairs (35 against 20, $p = 0.058$); near versus far survives them (42 against 21, $p = 0.011$).

What this adds. On the task with power, positions the mask reaches at distance two are more necessary than equally confident positions it reaches farther away, so the effect is a property of the neighborhood rather than of the immediate neighbors only. It is not evidence that proximity operates over and above confidence: the one comparison that could show that, offset 1 against offset 2, does not survive holding confidence fixed. Token identity and syntactic role are uncontrolled at both offsets, and it is one backbone and one task at the level of significance.

O Generality of the Attribution Finding: a Second Backbone and a Second Radius

The divergence between the two probes was first reported on one backbone at one radius. The decoder writes every per-neighbour field (`p_only_this`, `p_all_but_this`) into the Dream-7B traces and into the radius-2 traces as well, so the battery of §4.2 can be recomputed on them without decoding anything new. Table 21 reports all six cells under identical definitions: sufficiency is $p(\text{only } j) - p(\text{none})$, necessity is $p(\text{all}) - p(\text{all but } j)$, each divided by the event's own full-mask span, on events whose span is positive. Inference on the paired within-event contrast uses the same two complementary procedures as elsewhere in the paper, an event-level sign test that assumes events are independent and a problem-level cluster bootstrap that does not; neither is exact under the full nesting, and we read them together.

The qualitative finding replicates in every cell. The median within-event necessity-minus-sufficiency contrast is positive in all six, the sign test rejects in all six, and the problem-level interval excludes zero in all six. That answers the question the result was open to: removal alone understates the neighbourhood on Dream-7B as well as on LLaDA-8B-Base, on GSM8K as well as on code, and at radius 2 as well as radius 1.

Three differences are worth stating rather than smoothing over. The effect is smallest on GSM8K ($+0.019$, interval $[+0.008, +0.037]$ over 645 problems) and on Dream MBPP ($+0.049$, interval reaching down to $+0.004$); on those two cells the direction is established but the size is not much above zero, and we would not describe either as established on its own. The share of events where a single neighbour is sufficient by itself is much higher on Dream HumanEval (43.0%) than on LLaDA-8B-Base HumanEval (18.7%), so the backbone changes how often the neighbourhood acts through one position even though the direction of the divergence is unchanged. And the radius-2 cell has the largest contrast of the five ($+0.189$), which is consistent with a wider mask leaving more room between the two probes, not with any claim about distance; §N.2 explains why distance within the mask is not separable from confidence in these data.

One limitation of the reuse: the Dream and radius-2 traces were collected to answer other questions, so the events they contain were selected by those runs' decoding trajectories rather than sampled for this comparison. The battery is a recomputation on data already collected, not an independent replication with its own sampling design.

Table 21: The two-probe divergence in six cells spanning two backbones, three tasks and two radii.

Quantity	LLaDA-8B-Base, HumanEval, $r{=}1$	LLaDA-8B-Base, MBPP, $r{=}1$	Dream, HumanEval, $r{=}1$	Dream, MBPP, $r{=}1$	LLaDA-8B-Base, HumanEval, $r{=}2$	LLaDA-8B-Base, GSM8K, $r{=}1$
positive-span events with both probes	60	326	95	273	84	928
distinct problems	41	201	70	185	54	645
median sufficiency share	+0.08	+0.26	+0.22	+0.14	+0.16	+0.16
median necessity share	+0.52	+0.64	+0.70	+0.42	+0.54	+0.31
median within-event necessity − sufficiency	+0.106	+0.123	+0.088	+0.049	+0.189	+0.019
event-level sign test	40/20, 0.013 ✓	212/108, $< 10^{-4}$ ✓	61/31, 0.002 ✓	156/109, 0.005 ✓	57/25, 0.001 ✓	513/400, 0.000 ✓
problem-level bootstrap 95% CI	[+0.016, +0.277] ✓	[+0.068, +0.172] ✓	[+0.018, +0.251] ✓	[+0.004, +0.127] ✓	[+0.022, +0.287] ✓	[+0.008, +0.037] ✓
sufficient alone, share of applied	18.7%	23.8%	43.0%	28.3%	27.6%	32.5%
necessary not sufficient, share of applied	17.6%	21.2%	21.0%	22.3%	29.9%	15.5%

P Does the Attribution Statistic Track Program Correctness?

Everything the probes measure is a probability the decoder assigns to a proposed token under three masks. Whether that has anything to do with a program becoming correct is a separate question, and because every event carries a `doc_id` and both decode runs record per-problem `pass@1`, it is one we can measure rather than defer.

For each problem carrying at least one positive-span event we summarise its attribution statistics by the median over its events, and split problems by whether LCD changed the problem's outcome relative to CoRe. Table 22 reports the comparison. It is a null on both tasks and on both statistics. On HumanEval the four repaired problems have a *lower* median necessity share than the thirty-seven unchanged ones ($+0.17$ against $+0.56$, $p = 0.32$); on MBPP the six changed problems have a lower within-event contrast than the 195 unchanged ones (-0.07 against $+0.15$, $p = 0.10$). Both differences run opposite to the direction a causal reading would predict, and neither is significant.

The null is only as informative as its power, so we report that too, simulating a two-sided Mann–Whitney at the observed group sizes. Against a strong association, a probability of superiority of 0.90, the design has power 0.88 on HumanEval and 0.98 on MBPP; against a moderate one, 0.70, it has 0.24 and 0.39. What these data exclude is therefore a strong problem-level association between the attribution statistics and whether a program's outcome changes. They do not exclude a moderate one, and with four and six outcome-changing problems no design built on these runs could.

We draw the conservative conclusion. The positional association the paper reports is a statement about token probabilities under counterfactual masks, it replicates across backbones, tasks and radii at that level, and it has no detectable signature in which programs become correct. That is consistent with the paper's utility result, which found no reliable benefit, and it is the reason the contribution is framed as an attribution finding rather than a mechanism for the gain.

Table 22: Attribution statistics of problems whose outcome LCD changed against those it left unchanged, with simulated power.

Quantity	HumanEval	MBPP
problems with attribution events	41	201
outcome changed by LCD (of which repaired)	4 (4)	6 (2)
outcome unchanged	37	195
median necessity share, changed vs unchanged	+0.17 vs +0.56	+0.67 vs +0.66
Mann–Whitney p	0.32 ✗	0.83 ✗
median within-event necessity − sufficiency	+0.14 vs +0.18	−0.07 vs +0.15
Mann–Whitney p	0.75 ✗	0.10 ✗
power at probability of superiority 0.70	0.24	0.39
power at 0.80	0.55	0.77
power at 0.90	0.88	0.98

Q EQUAL-COMPUTE CONTROLS, PER PROBLEM

The main text reports that CoRe with sixteen verify calls recovers half of LCD's HumanEval gain at an identical budget. The per-problem view shows the two configurations are not interchangeable even though their aggregate difference is not significant. LCD repairs HumanEval problems 60, 66, 82 and 85; CoRe with sixteen verify calls repairs 4 and 71. The two sets are disjoint, so the extra budget is being spent on different errors rather than on the same ones by two routes. Head to head the split is four against two at exact $p = 0.688$.

The same view explains the budget-preserving result. The budget-preserving variant repairs 66 and 85 and loses 60 and 82, which are exactly the two problems whose repairs required LCD to take the remask slot at a position CoRe's own score had claimed. So the displacement the reviewer-identified top-k competition permits is not an incidental side effect of the implementation: it is responsible for half of the measured gain. A reader who wants the guarantee that LCD never removes one of CoRe's interventions should use the budget-preserving variant and expect $+1.22$ pp rather than $+2.44$ pp on this benchmark.

R A METHODOLOGICAL CAUTION: SUBSET SIGNALS CAN FLIP

Early in the search we screened configurations on small subsets to save compute, then re-ran the survivors at full scale. On MBPP this nearly misled us. A 150-example MBPP subset reported $\text{LCD}_{\text{naive}}$ *above* CoRe by $+0.67$ pp; the full 500-example MBPP reported the same configuration *below* CoRe by -0.80 pp, so the sign flips. Similarly, a 100-example GSM8K subset reported -2.00 pp for $\text{LCD}_{\text{naive}}$ while the full 1319 reported -0.91. We therefore ran the entire gate ablation (Figure 5) at full scale on the discriminative tasks, and we recommend the same for any revision-method comparison whose per-task deltas are at the one-percentage-point scale: at $n = 150$, a 0.7 pp difference is far inside the binomial noise band, and a subset screen can select the wrong configuration with high probability. The cost is real (a full MBPP run is roughly an order of magnitude longer than a 150-example screen), but the alternative is choosing a configuration on noise.

S MECHANISM STATISTICS, EXTENDED

Table 23 gives per-call remask and LCD-win rates for the naive and gated configurations, and Figure 9 plots the win rates.

Table 23: Per-call remask and LCD-win rates on LLaDA-8B-Base, naive against gated.

	rmsk/call		lcd-win/call	
Task	naive	LCD	naive	LCD
GSM8K	0.94	0.91	3.45	0.56
MATH	0.93	0.90	3.79	0.54
BBH	0.60	0.52	1.18	0.21
MBPP	0.87	–	2.35	–

The gates are selective, not blanket The drop in LCD-win rate from naive to gated (for example $3.45 \to 0.56$ on GSM8K, $3.79 \to 0.54$ on MATH, $1.18 \to 0.21$ on BBH) is about six-fold. If the gates simply turned LCD off, the win rate would go to zero and the headline would equal CoRe; instead it lands near half a position per call, which is the residual of confirmed, CoRe-silent discoveries that drive the HumanEval gain while doing no harm on reasoning.

The base remask rate barely moves Across tasks the per-call remask count under LCD (0.52 to 0.91) tracks CoRe's closely; LCD is not remasking more tokens, it is occasionally remasking a different, better-chosen token. This is consistent with the headline pattern of one task improving and the rest holding: the total intervention budget is essentially CoRe's, redirected.

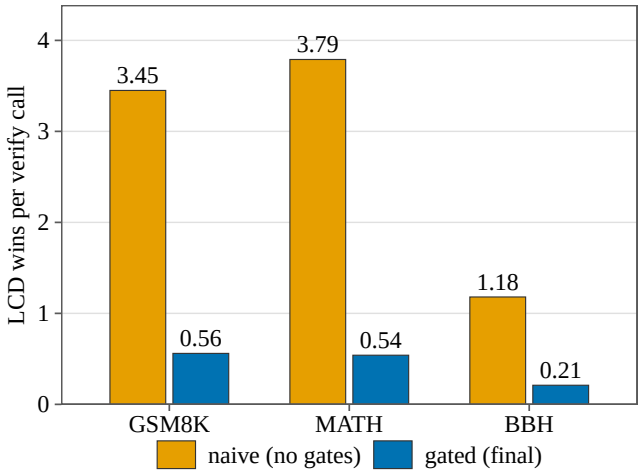

Figure 9: LCD-win rate per verify call, without gates and with both gates.

How often LCD displaces CoRe's own pick The additive rule stops LCD substituting a token at a position CoRe decided, but the remask set is still a top-k over the merged score under a budget of $R_t = 1$, so LCD can take the slot at a different position. We count this directly: at each verify call we record the positions CoRe's own score would have selected under the same budget, and how many of them the merged selection did not keep. Table 24 reports the rate. This is the quantity that decides whether "additive" is a property of the positions or of the budget, and it is the reason the budget-preserving variant exists.

Table 24: Displacement of CoRe's intended remasks by the merged top-k.

Quantity	value
pending: instrumented runs in progress	

BBH is the lowest-activity task BBH shows the lowest LCD-win rate (0.21 per call gated), and its delta is correspondingly the smallest in magnitude (-0.01 pp). Its outputs are short and templated, which offers fewer eligible committed neighbors to drop; we report the association without the co-adaptation reading we previously gave it, since Appendix U does not support that reading.

T Compute and Wall-Clock

All runs use 4× A100-80GB with `accelerate launch --num_processes 4`, bf16. Per-task wall-clock at the full configuration is dominated by sequence count and length: BBH (6,511 scored items) at roughly 10 hours and MATH (5,000) at roughly 9 hours are the long poles; GSM8K (1,319) is about 2 hours; MBPP (500) under an hour; HumanEval (164) about 12 minutes. The LCD forward-pass overhead (one extra pass on each of the 8 verify steps per generation, against 128 base steps plus 8 CoRe verify passes) is approximately 6% in forward passes and is within the per-generation wall-clock noise of CoRe. We did not observe a measurable throughput difference between $\text{LCD}_{r=0}$ and CoRe, consistent with the radius-zero pass being identical work.

U Mechanism Evidence on the Corrected Generations

This appendix has been rebuilt twice, and the second rebuild is a retraction. The submitted version gave a hypothetical pattern and declined causal attribution. Our first revision replaced it with instrumentation, and that instrumentation did not measure what we said it did. We describe the defect, because it is the kind that is easy to repeat, then the corrected measurement.

What the first instrumentation got wrong It recorded, at every position where the gated LCD evidence beat CoRe's, the proposed token's probability under LCD's mask and under CoRe's mask, and reported that the proposal was more probable with "neighbors dropped" than "restored" in 98.5% of 408 events. Three things were wrong with using that as evidence for a local mechanism. First, CoRe's mask restores *every* additionally masked position at once, not the particular neighbor in the story, so the comparison measures sensitivity to the whole neighborhood union and cannot attribute anything to one neighbor. Second, 113 of the 408 events, 27.7%, had no dropped immediate neighbor at all, because the adjacent positions were already candidates or were ineligible, and 112 of those 113 showed the same positive lift. A statistic that behaves almost identically on events where the claimed cause is absent is not evidence for the claimed cause. Third, the events were recorded before the revision budget was spent, so they included proposals the decoder never applied, and they carried no problem identifier, so they could not be joined to the four repaired programs; the first revision printed program diffs next to the aggregate and left the join to the reader, which is not the same thing.

The corrected measurement Three changes, all in the decoder rather than the analysis. Events are recorded after the top-k budget selection, so only applied proposals appear. Each event carries the `doc_id` of its problem, so events join to per-problem outcomes. And for every dropped immediate neighbor j of an applied position i, the decoder runs a dedicated forward pass masking CoRe's candidate set plus j alone, which yields $p(\text{only } j)$ and isolates that neighbor's contribution. The quantity the local account needs is then $p(\text{only } j) - p(\text{none})$, not $p(\text{all}) - p(\text{none})$, and Table 25 reports both so the difference between them is visible. Events with no dropped immediate neighbor are counted and reported separately rather than pooled into the claim.

The single number that bears most directly on a per-neighbor claim is the share of the whole-neighborhood effect that the best single neighbor accounts for,

$$\text{share} \;=\; \frac{\max_j p(\text{only } j) - p(\text{none})}{p(\text{all}) - p(\text{none})}, \tag{6}$$

defined only on events where the denominator is positive, that is where the full mask raises the proposal's probability at all; events where it lowers the probability are counted separately rather than divided through. A sufficiency share near one means dropping one neighbor reproduces almost everything dropping the whole neighborhood does. A share at or below zero means that neighbor alone is not what makes the proposal available, which is *not* the same as the effect coming from elsewhere: the neighbor may still be required in combination with other masked positions, and the necessity probe below is what distinguishes the two. We report medians and the fraction of events at either end, because the distribution matters more than its centre.

Table 25: Mechanism checks on applied LCD interventions under the three masks.

Check on applied LCD-win events	value
applied events recorded	91
with ≥ 1 dropped immediate neighbour	63 (69.2%)
with none, excluded from the claim	28 (30.8%)
on the events with a dropped immediate neighbour:	
extra positions masked per verify call, median	45
of which this candidate's own neighbours, median	3.3%
CoRe's verify proposed no change at the position	87.3%
dropped-neighbour commit confidence, median	0.699
isolated lift $p(\text{only } j) - p(\text{none})$, median	+0.009
share of isolated tests with positive lift	59.1%
whole-neighbourhood lift $p(\text{all}) - p(\text{none})$, median	+0.268
share of that effect the single best neighbour explains, median	0.08
events where one neighbour explains $\geq 50\%$	28.3%
events where it explains none of it	31.7%

Sufficiency, necessity, and what each can establish A previous revision measured only whether removing one neighbor alone reproduces the effect, found that it usually does not, and concluded that the effect came from elsewhere in the mask. That inference was wrong, and we withdraw it.

Removing one position tests whether it is *sufficient*; it says nothing about whether the position matters in combination with others, and two positions can jointly produce an effect neither produces alone. The decoder therefore now runs a second probe per dropped neighbor j: the full LCD mask with j *restored*, which tests necessity, together with a third that masks both immediate neighbors and nothing else. Writing $\text{span} = p(\text{all}) - p(\text{none})$, we report $[\max_j p(\text{only } j) - p(\text{none})]/\text{span}$ for sufficiency, $[p(\text{all}) - \min_j p(\text{all but } j)]/\text{span}$ for necessity, and $[p(\text{both}) - p(\text{none})]/\text{span}$ for jointness.

Table 26: Sufficiency, necessity and jointness shares on applied LCD interventions.

Quantity	all applied events	repaired problems
events	60	7
median sufficiency share	$+0.08$	$+0.00$
median necessity share	$+0.52$	$+0.21$
median jointness share	$+0.10$	-0.04
neighbour sufficient alone	28%	0%
necessary but not sufficient	27%	43%

What the corrected measurement shows Table 25 gives the descriptive picture: the verify state masks a median of 45 positions per call, of which a given candidate's own immediate neighbors are 3.3%, and 30.8% of applied interventions drop no immediate neighbor at all. CORE's verify had proposed no change at the position in 87.3% of cases, and the dropped neighbors are committed at a median confidence of 0.699.

Table 26 is the part that bears on attribution, and the two probes disagree. Sufficiency is small, a median of 0.08 of the span, which is what we previously reported. Necessity is not: restoring a single neighbor inside the full mask removes a median of 0.52 of the effect. To classify events we call a neighbor *sufficient* when its sufficiency share reaches 0.5 and *necessary* when its necessity share does, a threshold we fix once and apply to both probes; it is arbitrary in the way any cut on a continuous quantity is, and the medians above do not depend on it. Of the 60 events that have a dropped neighbor, both probes, and a positive span, 17 have a neighbor that is sufficient alone, 16 have one that is necessary without being sufficient, and 27 have neither. As fractions of all 91 applied interventions those are 18.7%, 17.6% and 29.7%; the remaining 34% either drop no immediate neighbor or have a span that is not positive. We quote both denominators because the first is the subset the ratio is defined on and the second is the population a reader will have in mind.

So the per-position account is not refuted; it is unsupported in its sufficiency form and partially supported in its necessity form, and on roughly one applied intervention in six the implicated neighbor matters only in combination with other masked positions. The narrow claim we can state is that single-neighbor removal usually does not reproduce the full-neighborhood effect. Jointness adds little, a median of 0.10 over 29 events where both immediate neighbors exist, so the adjacent pair is not together doing what each fails to do alone, and whatever the interaction involves reaches beyond that pair.

The four repairs specifically Table 27 gives both probes on every applied intervention on the four problems that are the whole gain, and the two tell different stories. On sufficiency the account fails outright: no neighbor exceeds 0.17 of the span, six of eleven tests are negative, and the median is zero. HumanEval 60 is the sharpest case and was our previous revision's leading example. A committed `return` is re-proposed as `sum`, reaching probability 1.000 under the whole-neighborhood mask against 0.106 with nothing dropped. Dropping the implicated neighbor alone, an indentation token committed at 0.99, gives 0.095, slightly below baseline, and restoring it inside the full mask changes nothing either, so on that event the repair is not attributable to that neighbor at all. On necessity the picture differs: three of the eleven tests have a neighbor whose restoration removes most of the effect, including $+1.27$ and $+0.91$, so on those events the neighbor is doing something that requires the rest of the mask to be present. We therefore report the four repairs as a mixture rather than as a refutation, and we no longer claim the effect came from elsewhere.

What this can and cannot establish With the isolation in place, two of the reviewer-relevant sub-claims are measurable and are measured: that the neighbor LCD drops was itself committed at high confidence, and that CORE's verify declined to change the candidate under the full context.

Problem	step	p(none)	p(only j)	p(all but j)	p(all)	suff.	nec.
HumanEval 60	56	0.106	0.095	1.000	1.000	−0.01	+0.00
HumanEval 66	48	0.291	0.268	0.326	0.328	−0.63	+0.05
HumanEval 66	48	0.291	0.293	0.320	0.328	+0.05	+0.21
HumanEval 82	48	0.107	0.092	0.484	0.385	−0.05	−0.36
HumanEval 82	48	0.107	0.117	0.033	0.385	+0.04	+1.27
HumanEval 82	56	0.223	0.293	0.703	0.645	+0.17	−0.14
HumanEval 82	56	0.223	0.244	0.660	0.645	+0.05	−0.04
HumanEval 82	88	0.406	0.375	0.824	0.883	−0.07	+0.12
HumanEval 82	88	0.406	0.404	0.891	0.883	−0.00	−0.02
HumanEval 85	48	0.348	0.320	0.377	0.680	−0.08	+0.91
HumanEval 85	72	0.408	0.408	0.438	0.680	+0.00	+0.89

Table 27: Both probes on every applied intervention on the four repaired problems.

The third, whether removing that particular neighbor is what made the proposal available, is what p(only j) addresses, and we report it as a distribution rather than as a single headline number. The fourth, that the dropped neighbor was *wrong*, remains outside what these benchmarks can show, since neither HumanEval nor MBPP carries token-level ground truth; for that we report only the executable consequence, the diff between CoRe's and LCD's final programs on the problems LCD repairs, now joined to the events by `doc_id` rather than printed alongside them. We make no claim of per-token causal attribution beyond the interventions the trace performs.

V The Four Repaired HumanEval Problems

Table 27 gives the probe values on the four problems LCD repairs; this appendix shows the programs. Each box holds the HumanEval prompt (function header and docstring) followed by CoRe's completion and LCD's, from the seed-42 greedy runs of Table 1, with the token that differs underlined in red and green. All four repairs are single-token: a `return` that became an accumulation, a variable name that became consistent, a comparison operator, and a parity constant. The caveat of Appendix P applies to every one of them: the probes attribute the change in the proposed token's probability to the masked neighborhood, not the change in program correctness.

HumanEval/60, sum_to_n

```
def sum_to_n(n: int):
    """sum_to_n is a function that sums numbers from 1 to n.
    >>> sum_to_n(30)
    465
    >>> sum_to_n(100)
    5050
    >>> sum_to_n(5)
    15
    >>> sum_to_n(10)
    55
    >>> sum_to_n(1)
    1
    """

# CoRe completion: fails the tests
    sum =  0
    for i in range(1, n +1):
        return  i
    return sum

# LCD completion: passes
    sum =  0
    for i in range(1, n +1):
        sum += i
    return sum
```

Figure 10: HumanEval/60: CoRe returns inside the loop; LCD accumulates.

HumanEval/66, digitSum

```
def digitSum(s):
    """Task
    Write a function that takes a string as input and returns the sum of the upper
      characters only'
    ASCII codes.

    Examples:
        digitSum("") => 0
        digitSum("abAB") => 131
        digitSum("abcCd") => 67
        digitSum("helloE") => 69
        digitSum("woArBld") => 131
        digitSum("aAaaaXa") => 153
    """

# CoRe completion: fails the tests
    total = 0

    for letter in s:
        if letter.isupper():
            sum += ord(letter)

    return sum

# LCD completion: passes
    sum = 0

    for letter in s:
        if letter.isupper():
            sum += ord(letter)

    return sum
```

Figure 11: HumanEval/66: CoRe initialises `total` but sums into `sum`; LCD names one variable.

HumanEval/82, prime_length

```
def prime_length(string):
    """Write a function that takes a string and returns True if the string
    length is a prime number or False otherwise
    Examples
    prime_length('Hello') == True
    prime_length('abcdcba') == True
    prime_length('kittens') == True
    prime_length('orange') == False
    """

# CoRe completion: fails the tests
    length = len(string)
    if length == 2:
        return False
    if length in range(2,4):
        return True
    for i in range(2,length):
        if length % i == 0:
            return False
    return True

# LCD completion: passes
    length = len(string)
    if length ≤ 2:
        return False
    if length in range(2,4):
        return True
    for i in range(2, length):
        if length % i == 0:
            return False
    return True
```

Figure 12: HumanEval/82: CoRe rejects only length two; LCD rejects lengths below two.

HumanEval/85, add

```
def add(lst):
    """Given a non-empty list of integers lst. add the even elements that are at odd
      indices..


    Examples:
        add([4, 2, 6, 7]) ==> 2
    """

# CoRe completion: fails the tests
    result = 0
    for i in range(len(lst)):
        if i % 2 == 0 and lst[i] % 2 == 0:
            result += lst[i]
    return result

# LCD completion: passes
    result = 0
    for i in range(len(lst)):
        if i % 2 == 1 and lst[i] % 2 == 0:
            result += lst[i]
    return result
```

Figure 13: HumanEval/85: CoRe tests even indices; LCD tests odd indices as the task asks.