LOGICAL EQUIVARIANCE IN DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Rewriting a problem without changing its answer can change whether a language model solves it, even when average accuracy barely moves. We study this as logical equivariance: predictions should transform consistently with answer-preserving input changes. In four diffusion language models, paired evaluations on Sudoku, code, and mathematics reveal many problems solved in one presentation but not another. We build a Sudoku benchmark whose forced targets transport exactly under rotations and reflections, and prove that exact grid equivariance leaves row-major relative-score attention only two or three offset classes, so stability must be learned. In post-training, Reasoning-State Masking (RSM) improves solving but alone reduces stability; presentation randomization (PR) removes this cost, and its gain in normalized reliability is 8.3 points larger with RSM. Matched controls reproduce this interaction at two scales and trace its stability gain to training on partial states at RSM's density. The combined recipe raises LLaDA-8B's all-view correctness from 47.7% to 86.1% alongside higher accuracy, and improves worst-view maze success by 7.3–18.8 points.

1 INTRODUCTION

The same problem can be written in many ways. Variables can take different names, and statements or conditions can appear in a different order, without changing what is being asked. A model that can solve a problem should therefore solve it however it is written. Current models do not always meet this expectation (Lu et al., 2022; Liu et al., 2023). In Figure 1, LLaDA-8B (Nie et al., 2025a) answers a Sudoku puzzle, a code completion, and a math question correctly in one form but fails after an equivalent rewrite. The form of a correct answer may change with the input, but its correctness should not. Behavioral testing and code-generation robustness studies have highlighted the importance of this requirement (Ribeiro et al., 2020; Wang et al., 2022).

Average accuracy can hide how often these successes are lost. In our MATH-500 evaluation, consistent variable renaming raises LLaDA-8B's accuracy from 28.5% to 30.2%, even though correctness changes on 17.9% of instances. Renaming breaks 8.1% of all instances, more than a quarter of those originally solved, while repairing another 9.8%. These changes largely cancel in the average, but not for the user whose problem is no longer solved.

Preserving success across equivalent inputs is a question of generalization: can a learned solving ability transfer when notation changes while the defining relations remain fixed? This also provides a practical reason to seek internal representations that capture these relations (Cohen & Welling, 2016; Bronstein et al., 2021). We study this observable property as *logical equivariance* (LE), where predictions transform consistently with specified answer-preserving input changes. Tracking each problem across its views distinguishes favorable-view success from consistent correctness in every presentation, building on consistency evaluation (Elazar et al., 2021; Cho et al., 2025).

Diffusion language models make this learning question concrete (Austin et al., 2021; Ou et al., 2024; Sahoo et al., 2024; Shi et al., 2024; Nie et al., 2025b; Ye et al., 2025). Their masked-denoising objective trains on partially observed inputs, so the masking distribution determines which conditional tasks the model practices. Flexible decoding can choose which position to fill next (Kim et al., 2025a; Liu et al., 2024; Peng et al., 2025; Kim et al., 2025b; Ni et al., 2026); presentation changes instead alter which problem object occupies each position. We investigate how the selection of partial inputs and exposure to equivalent views affect reliable solving.

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.

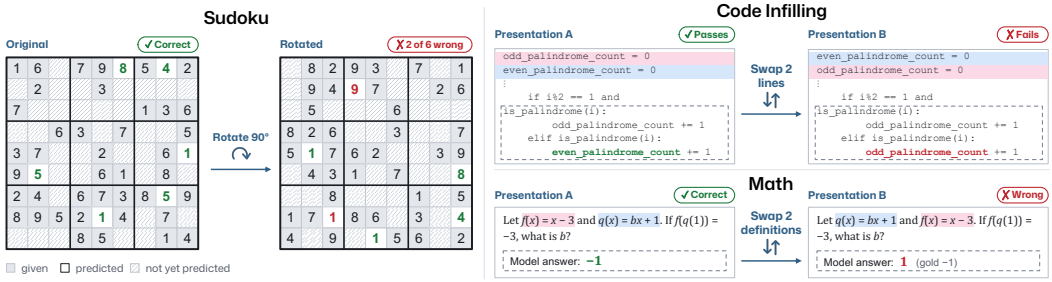

Figure 1: **A correct answer does not survive an equivalent change of presentation.** Rotating a Sudoku board, swapping two independent lines of code, or reordering two definitions in a math problem leaves the task and its answer unchanged, yet LLaDA-8B solves one presentation and fails the other. Accuracy on a single presentation cannot reveal this: whether a problem counts as solved depends on how it happens to be written.

Building on Sudoku reasoning benchmarks (Seely et al., 2025), our SUDOKU-AC-ORBIT benchmark supplies an exact test: its targets are logically forced and transport under all eight rotations and reflections. Together with paired code-infilling and mathematics evaluations, it reveals presentation-dependent correctness in four large dLLMs. On Sudoku-AC32, Dream-7B, iLLaDA-8B, and LLaDA-MoE each solve about 99% of states in at least one view, but only 12.9–23.8% in every view.

We also characterize the positional information compatible with this symmetry. A row-major relative-score attention head (Vaswani et al., 2017) is exactly D_4-equivariant under a position-only action only if its offset-indexed score functions collapse to two classes on even-sized grids and three on odd-sized grids (Theorem 4.1). On a 9×9 grid, 161 offsets become three classes. Standard row-major RoPE does not impose these ties (Su et al., 2024); we therefore investigate presentation stability through post-training while keeping the backbone unchanged.

We combine *Reasoning-State Masking* (RSM), which supplies partial inputs encountered during a valid reasoning process and supervises every unresolved target rather than a solver-ordered output sequence (Shah et al., 2024), with *presentation randomization* (PR), which presents those inputs under equivalent views. In the initial 50M recipe study, RSM improves solving but lowers normalized orbit reliability; adding PR retains the solving gain and improves worst-view success. The combination leads that study in mean success, worst-view success, sECR, and token-marginal agreement. The recipe also improves all-view correctness on LLaDA-8B and yields substantial gains over random-mask+PR on maze completion.

Our contributions are:

- **Presentation-level evaluation.** Exact transported targets and paired evaluations expose changes in correctness that average accuracy conceals across four diffusion models and three task families.
- **Serialization rigidity.** We characterize the offset distinctions compatible with exact grid symmetry in relative-score attention, including its bilinear and RoPE instances.
- **Complementary post-training choices.** RSM and PR produce a positive normalized-reliability interaction in Sudoku and useful recipe-level gains across additional Sudoku symmetries, an 8B model, and maze completion. Controlled follow-up experiments distinguish this interaction from endpoint superiority.

2 RELATED WORK

Consistency and structured evaluation. Ribeiro et al. (2020) introduced behavioral testing as an alternative to aggregate accuracy metrics, while Elazar et al. (2021) evaluated consistency across meaning-preserving inputs, tracking both single-instance success and all-instance correctness. Similarly, Sclar et al. (2024) analyzed prompt-format sensitivity, and Wang et al. (2022) assessed code generation robustness under semantics-preserving edits. We extend this evaluation perspective to

diffusion reasoning models under exact target transportation. While Sudoku variants modify core task constraints (Seely et al., 2025), our within-orbit comparisons hold them strictly invariant.

Selecting conditional training tasks. Masked diffusion learns conditional token predictors (Sahoo et al., 2024; Nie et al., 2025a; Ye et al., 2025). Kim et al. (2025a) analyze masking-induced prediction tasks and adaptive decoding, while Shih et al. (2022) select and reweight conditionals for any-order autoregressive models. Shah et al. (2024) supervise solver-ordered output sequences; AGDO uses attention-derived dependencies for denoising and post-training (Deng et al., 2026). RSM instead uses a reasoning process to select partial input states and retains dense completion of unresolved targets. We study how this state distribution interacts with equivalent-view exposure.

Equivariance and augmentation. Parameter sharing and compatible positional encodings can enforce equivariance (Ravanbakhsh et al., 2017; Romero & Cordonnier, 2021). Our theorem identifies the additional restrictions imposed by row-major offset sharing. The supplementary loss decomposition follows the logarithmic-opinion-pool identity of Heskes (1997).

3 Logical Equivariance

3.1 Equivalent problems and transported outputs

Let a finite group G act bijectively on inputs $\mathcal{X}$ and outputs $\mathcal{Y}$, preserving the task's correctness relation. Include an invalid-output symbol $\perp$ with $g\perp = \perp$. A deterministic decoder D is logically equivariant when $D(gx) = gD(x)$. This property is distinct from correctness: an equivariant model may be consistently wrong. For a stochastic decoder with law $K_\theta(\cdot \mid x)$, define

$$P_g(\cdot \mid x) = g_{\#}^{-1} K_\theta(\cdot \mid gx), \qquad \text{LE on } x \iff P_g = P_h \text{ for every } g, h \in G, \tag{1}$$

where $g_{\#}$ denotes pushforward. Canonicalization removes the prescribed transformation of the answer. We evaluate one base problem together with its views, preserving the pairing between presentations. For code and mathematics transformations that are not a closed group, we use validated original–transformed pairs and the corresponding correctness evaluator.

3.2 An exact orbit with forced targets

An anchored-completion state $x_{\mathrm{AC}}(A, U, R)$ specifies visible value-assigned cells A, scored cells U, and other unresolved cells R. If $\mathcal{S}(A)$ is the set of valid Sudoku completions consistent with A, we retain states satisfying

$$\{s_U : s \in \mathcal{S}(A)\} = \{s_U^\star\}. \tag{2}$$

Thus all valid completions agree on the scored answer. Rotations and reflections preserve rows, columns, and boxes, so $\mathcal{S}(gA) = g\mathcal{S}(A)$ and the unique target assignment transports with g. Appendix A.1 proves this claim. We generate target sets from contiguous solver-trace windows: ACk denotes k scored cells, with earlier assignments exposed as anchors.

3.3 Correctness across presentations

For the recorded prediction in view g, let $a_g(x)$ indicate correctness after canonicalization. With $m = |G|$, define

$$\mathrm{Acc} = \mathbb{E}_x \frac{1}{m} \sum_g a_g(x), \quad \mathrm{ANY} = \mathbb{E}_x \max_g a_g(x), \quad \mathrm{ALL} = \mathbb{E}_x \min_g a_g(x). \tag{3}$$

For a fixed deterministic decoder, ANY and ALL measure favorable-view and all-view correctness. For two views, let b and r be the fractions of all instances broken and repaired by the transformation. Then

$$\mathrm{Acc}_g - \mathrm{Acc}_e = r - b, \qquad \mathrm{ANY} - \mathrm{ALL} = r + b. \tag{4}$$

Net accuracy change can therefore be small even when many instances change correctness. With sampled predictions, these formulas describe the recorded outcomes rather than equality of the output laws.

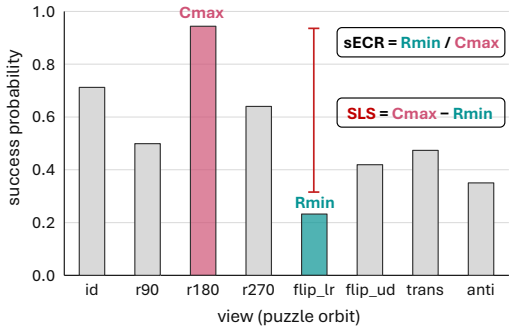

Figure 2: **Visualizing stochastic orbit metrics.** For each logical state, we estimate success under every D_4 view after canonicalizing predictions back to the original board. $C_{\max}$ captures best-view capacity, $R_{\min}$ captures worst-view reliability, and sECR measures the fraction of best-view capacity retained in the worst view.

Table 1: **Metrics used in this paper.** Recorded metrics use one prediction per view. Sampled metrics use K samples per view, and SLS includes finite-sample extrema bias.

Metric	Definition	
Recorded (Table 2)		
Acc	mean correctness over views	↑
ANY	correct in at least one view	
ALL	correct in every view	↑
cLE	100 ALL/ANY	↑
Sampled (Tables 3, 4)		
MeanSucc	mean success over views	↑
$\widehat{C}_{\max}$	best-view success	↑
$\widehat{R}_{\min}$	worst-view success	↑
SLS	$\widehat{C}_{\max} - \widehat{R}_{\min}$	↓
sECR	$\widehat{R}_{\min}/\widehat{C}_{\max}$	↑
$\widehat{L}_{\text{tok}}$	token-marginal view leakage	↓

For stochastic decoding, let $p_g(x)$ denote task-success probability. We estimate it from K samples per view and report

$$\begin{aligned} \text{MeanSucc} &= \mathbb{E}_x \frac{1}{m} \sum_g p_g(x), \qquad C_{\max} = \mathbb{E}_x \max_g p_g(x), \\ R_{\min} &= \mathbb{E}_x \min_g p_g(x), \qquad \text{sECR} = R_{\min}/C_{\max}. \end{aligned} \tag{5}$$

Hats denote finite-K estimates, and the ratio is defined when $C_{\max} > 0$. Figure 2 illustrates how these quantities are computed from per-view success rates. Extrema over sampled success rates are biased, so the Sudoku recipe study includes a canonical pseudo-orbit control (Appendix A.6). Correctness summaries are complemented by token-marginal view leakage $\widehat{L}_{\text{tok}}$. Its population counterpart averages KL divergence to the across-view mixture, and full-output leakage is zero exactly when Eq. 1 holds (Appendix A.2). Table 6 lists every metric used in the paper with its definition.

4 Method

RSM changes which conditional tasks receive supervision; PR changes how those tasks are presented. We retain the general state-selection objective and dense completion of unresolved targets. The structural result characterizes what the unchanged positional component can guarantee.

4.1 Reasoning-State Masking

Let $e = (c, y^\star)$ contain a problem specification c and a structured reference $y^\star$ indexed by $\mathcal{I}_e$. A task-specific state generator $\mathcal{O}$ supplies a valid reasoning trajectory $(s_0, \ldots, s_T)$. Each nonterminal state defines an input $x_t = \text{Enc}(c, s_t)$ and unresolved target indices $H_t \subseteq \mathcal{I}_e$. With example sampler ν and state sampler π_e, RSM optimizes

$$\mathcal{L}_{\text{RSM}}(\theta) = \mathbb{E}_{e\sim\nu,\, t\sim\pi_e} \frac{1}{|H_t|} \sum_{i\in H_t} -\log p_{\theta,i}(y_i^\star \mid x_t). \tag{6}$$

The reasoning process selects the next *input state*; every unresolved target remains supervised. For multi-token fields, the encoding and token-level normalization are part of the task specification.

Which conditional tasks are practiced? RSM changes the distribution of visible information, not the completion target or inference budget. Random partial observations of a correct reference may themselves be logically consistent; RSM selects states encountered by a specified reasoning process. Sound deductions can make consequences explicit without changing the feasible answers. Appendix E formalizes this distinction. The relevant learning question is which selected conditions improve behavior on held-out problems.

4.2 TASK INSTANTIATIONS AND EQUIVALENT VIEWS

Sudoku. Here c contains the clues and rules, $y^\star$ is a reference board, and s_t consists of visible assignments A_t. An offline one-wave oracle identifies frontier F_t and forms

$$F_t = \mathcal{O}_{1\text{wave}}(y^\star, A_t), \qquad A_{t+1} = A_t \cup F_t. \tag{7}$$

The implemented hybrid procedure supplies a fallback expansion when required; the retained state records define the training distribution. With $H_t = C \setminus A_t$, Eq. 6 becomes dense completion of the hidden cells,

$$\ell_{\text{RSM}}(\theta; x_t, y^\star) = \frac{1}{|H_t|} \sum_{i \in H_t} -\log p_{\theta,i}(y_i^\star \mid x_t). \tag{8}$$

This is dense completion, not frontier-only supervision. Oracle rules, fallback validation, and the sampling probabilities determine the realized training distribution.

Maze and other instantiations. For maze completion, walls and start/end markers remain visible. RSM uses cached intermediate assignments and supervises every still-hidden corridor target. The experiment's state construction is distinct from random masking at an arbitrary ratio (Section 5.5). Verified derivations or execution-derived values provide analogous constructions in mathematics and programming (Appendix E); these illustrations are not claims of measured training transfer to those domains.

Presentation randomization. PR samples $g \sim \text{Unif}(G)$ per training instance and transports the input and targets together. Write gi for the corresponding transformed target index. The joint risk is

$$\mathcal{L}_{\text{RSM+PR}}(\theta) = \mathbb{E}_{e,t,g} \frac{1}{|H_t|} \sum_{i \in H_t} -\log p_{\theta,gi}((gy^\star)_{gi} \mid gx_t). \tag{9}$$

RSM chooses the conditional tasks; PR supplies equivalent representations. We use $G = D_4$ for Sudoku and maze. The online RSM+PR implementation transforms cached states rather than rerunning the state generator in each orientation. Random-view sampling and grouped orbit averaging implement the same population objective under matched state weights (Chen et al., 2020). The state generator prepares supervision offline; post-training changes model parameters or adapters, leaving the backbone and inference procedure intact.

Training procedure. We cache validated nonterminal states (x_t, H_t) from the state generator. At each fine-tuning update, we sample an example $e \sim \nu$, a state $t \sim \pi_e$, and a presentation $g \sim \text{Unif}(G)$; transport the input, target indices, and labels together; and update the trainable parameters with Eq. 9. The architecture and inference procedure are unchanged.

4.3 CONDITIONAL-STATE STRUCTURE AND SUPERVISION ALLOCATION

PR averages cross-entropy over equivalent views; the standard opinion-pool decomposition separates its consensus and disagreement components (Appendix E.1). RSM selects the state–target pairs on which this objective is evaluated. For nested reveal states, target i remains supervised until reveal time $\tau(i)$, accumulating coefficient

$$\omega_i = \sum_{t < \tau(i)} \frac{\pi_e(t)}{|H_t|}. \tag{10}$$

Its loss also depends on the other information visible at each occurrence. Appendix C tests whether conditional-state structure matters beyond these marginal weights.

4.4 THE POSITIONAL INFORMATION COMPATIBLE WITH GRID SYMMETRY

Let $\Lambda_n = \{0, \dots, n-1\}^2$, $p(r, c) = nr + c$, and $d(u, v) = p(v) - p(u)$. Consider full, unmasked attention with finite scores

$$L_{uv}(Q, K) = \ell_{d(u,v)}(Q_u, K_v), \qquad \text{Att}_\ell(Q, K, V) = \text{softmax}_v(L)V. \tag{11}$$

The position-only action satisfies $(\Pi_g Q)_{gu} = Q_u$.

Theorem 4.1 (Serialization rigidity for relative-score attention)**.** *For $n \geq 2$, $\mathrm{Att}_\ell(\Pi_g Q, \Pi_g K, \Pi_g V) = \Pi_g \mathrm{Att}_\ell(Q, K, V)$ for every $g \in D_4$ and all Q, K, V if and only if*

$$\ell_{d(gu,gv)}(q,k) = \ell_{d(u,v)}(q,k) \quad \textit{for all } g,u,v,q,k. \tag{12}$$

The induced offset classes are $\{0\}$ and the nonzero offsets for even n; for odd n, they are $\{0\}$, nonzero even offsets, and odd offsets. In particular, if $\mathcal{T}_n$ is the space of scalar row-major offset kernels and $\mathcal{C}_n$ the space of D_4-invariant pair kernels, then

$$\dim(\mathcal{T}_n \cap \mathcal{C}_n) = \begin{cases} 2, & n \textit{ even}, \\ 3, & n \textit{ odd}. \end{cases} \tag{13}$$

The compatible positional vocabulary remains constant as the grid grows. For $n = 9$, 161 offsets reduce to three classes. The reduction follows from the transitive closure of two sharing rules: symmetry-related displacements share a score, and equal serialized offsets share a score. Equivariance of attention outputs identifies the attention matrices; the self-edge removes softmax's additive ambiguity. Appendix A.4 gives the proof and the general transport-graph formulation.

For bilinear-affine scores $\ell_d(q,k) = q^\top M_d k + b_d$, Eq. 12 ties M_d and b_d within each class. Row-major RoPE uses $M_d = R_d$ (Su et al., 2024); a head with $R_1 \neq R_9$ fails the 9×9 condition. The theorem specifies a component-level certificate under a position-only action. Learned representations and layer interactions can still yield equivariance in the full model. Our training intervention evaluates end-to-end behavior without imposing these positional ties.

5 Experiments

We first evaluate whether solving ability persists across presentations, then examine model scale and post-training. The central recipe study separates the effects of RSM and PR; additional symmetry groups, an 8B model, and maze completion test the recipe's reach.

5.1 Protocol

Data and tasks. The symbolic models use Kaggle-3M Sudoku, with 500 test puzzles, 2,000 validation puzzles, 80,000 puzzles reserved for post-training, and the remaining approximately 2.9 million for pretraining. k scores a forced k-cell target after revealing earlier solver assignments; the initial post-training study scores full completion of the original blanks. We also evaluate HumanEval-Infilling (Chen et al., 2021) on 4,318 original–transformed pairs and MATH-500 under variable renaming, premise permutation, and identity-operation insertion. Code edits are checked against the execution harness, and mathematical edits preserve the checked answer (Appendix D.9).

Models. The large-model evaluation covers LLaDA-8B (Nie et al., 2025a), Dream-7B (Ye et al., 2025), iLLaDA-8B (Nie et al., 2026), and LLaDA-MoE (Zhu et al., 2025). Controlled models are grid-token MDLMs with DiT backbones and SUBS parameterization, trained at 5M–100M parameters. Post-training leaves the backbone unchanged. Model and task-specific training budgets appear in Appendix D.

Evaluation. We canonicalize predictions before scoring. Table 2 summarizes one recorded prediction per view using Acc, ANY, and ALL. Symbolic sampling evaluations use $K = 16$ per view and report MeanSucc, $\widehat{C}_{\max}$, $\widehat{R}_{\min}$, $\widehat{\mathrm{sECR}}$, and token-marginal leakage (Table 6). For the initial Sudoku study, real-minus-pseudo contrasts accompany the raw extrema. Reported confidence intervals use paired bootstrap resampling of base puzzles, retaining their views and repeated samples together (Appendix D.6).

5.2 Large models do not retain every success

LLaDA-8B solves 97.7% of AC16 states in at least one presentation, but only 1.8% in all eight. On AC32, Dream-7B, iLLaDA-8B, and LLaDA-MoE each solve about 99% in some view and 12.9–23.8% in every view (Table 2). The favorable-view score reveals successful completions that a single layout fails to deliver consistently.

Table 2: **Correctness across equivalent presentations.** Percentages for the recorded completions. Sudoku uses eight views; code and mathematics use original–transformed pairs. MATH-500 pools three transformations. cLE is $100\,\mathrm{ALL}/\mathrm{ANY}$.

Task	Model	Setting	Acc	ANY	ALL	cLE
Sudoku	LLaDA-8B	AC16	49.7	97.7	1.8	1.8
	LLaDA-8B	AC24	20.3	75.2	0.1	0.1
	LLaDA-8B	AC32	3.9	24.9	0.0	0.0
	Dream-7B	AC24	89.3	100.0	53.8	53.8
	Dream-7B	AC32	68.0	98.9	17.4	17.6
	iLLaDA-8B	AC32	75.2	99.0	23.8	24.0
	LLaDA-MoE	AC32	66.3	98.9	12.9	13.0
Code infilling	LLaDA-8B	Pairs	43.5	57.4	29.5	51.4
	Dream-7B	Pairs	46.8	63.2	30.3	47.9
	iLLaDA-8B	Pairs	30.2	44.1	16.2	36.7
	LLaDA-MoE	Pairs	43.6	60.8	26.4	43.4
MATH-500	LLaDA-8B	Pairs	29.0	38.7	19.4	50.1
	Dream-7B	Pairs	18.4	24.9	11.9	47.8
	iLLaDA-8B	Pairs	26.3	34.5	18.1	52.5
	LLaDA-MoE	Pairs	34.4	47.4	21.3	44.9

Table 3: **Controlled MDLM scaling.** Pretrained models evaluated with $K = 16$ samples per view. Success rates, sECR, and SLS are percentages; leakage is in nats. SLS $= \widehat{C}_{\max} - \widehat{R}_{\min}$ includes finite-sample extrema bias.

Params	Window	MeanSucc ↑	$\widehat{R}_{\min}$ ↑	sECR ↑	SLS ↓	$\widehat{L}_{\mathrm{tok}}$ ↓
5M	AC16	16.87	6.18	20.53	23.91	0.096
	AC24	12.76	3.69	14.89	21.07	0.097
10M	AC16	30.13	16.20	35.52	29.41	0.083
	AC24	26.31	12.88	31.08	28.56	0.085
50M	AC16	**36.27**	**21.61**	**41.24**	30.79	**0.076**
	AC24	**32.85**	**18.71**	**38.53**	29.86	**0.077**
100M	AC16	34.61	20.32	40.61	29.72	**0.076**
	AC24	31.07	17.24	37.15	29.16	**0.077**

The same issue appears beyond grids. Recorded correctness differs between paired presentations on 27.9–34.4% of code instances and 13.0–26.1% of mathematics instances. For iLLaDA-8B under inert-line insertion, original accuracy is 28.8% and the two-view average is 29.0%, yet ANY and ALL are 45.7% and 12.2%. Aggregate accuracy barely changes while many individual outcomes do. Appendix B.2 gives the transformation breakdown.

5.3 Scaling improves solving, but not uniformly

Training on a common symbolic corpus lets us examine scale without differences in language-model pretraining. From 5M to 50M, mean success on AC24 rises from 12.76% to 32.85% and estimated worst-view success from 3.69% to 18.71%, and sECR rises from 14.89% to 38.53% (Table 3). The 100M model does not improve further. Scale therefore raises both average and worst-view success, but the best- and worst-view estimates remain about 30 points apart at every scale above 5M.

Part of this spread is finite-sample extrema bias. The canonical pseudo-orbit calibrates it: on the full-completion evaluation of Table 4, the pretrained 50M model's spread exceeds its pseudo-orbit reference by only 1.24 points. Presentation dependence in these symbolic models is therefore modest after pretraining, and, as the next section shows, it is shaped mainly by post-training choices. The autoregressive comparison appears in Appendix B.3.

Table 4: **The initial 50M recipe study.** All branches use the same pretrained checkpoint and 9,000 updates. Success rates and sECR are percentages; leakage is in nats. RSM+PR leads this study in mean success, estimated worst-view success, sECR, and token-marginal agreement. Full extrema diagnostics and paired intervals are in Appendix B.4.

Recipe	MeanSucc ↑	$\widehat{C}_{\max}$ ↑	$\widehat{R}_{\min}$ ↑	sECR ↑	$\widehat{L}_{\text{tok}}$ ↓
Pretrained	31.53	47.29	17.41	36.82	0.0775
Canonical 80k	41.58	57.79	25.94	44.88	0.0712
Canonical 10k	43.12	59.76	27.43	45.89	0.0714
PR	42.16	58.46	26.64	45.56	0.0709
PR grouped	41.15	57.35	25.60	44.64	0.0709
RSM	45.74	**65.81**	26.16	39.75	0.0783
RSM+PR	**47.00**	63.89	**30.50**	**47.74**	**0.0684**

5.4 RSM AND PR ARE COMPLEMENTARY

Table 4 compares complete training recipes. Canonical SFT uses 10k or 80k puzzles, and PR randomizes the views of the 10k set. RSM uses the 4.6k puzzles for which the deduction procedure produces a trajectory, yielding 41k intermediate states. All branches use the same update count; RSM changes both the selected puzzle subset and the partial inputs.

More solving, less uniform delivery. RSM raises MeanSucc from 43.12% to 45.74% relative to canonical 10k SFT and reaches the highest estimated favorable-view success, 65.81%. On the canonical layout alone, it improves success by 2.0 points (95% CI $[0.7, 3.3]$). Yet its sECR falls by 6.1 points ($[-7.7, -4.6]$), estimated worst-view success changes from 27.43% to 26.16%, and token leakage increases. The preferred recipe therefore depends on whether evaluation follows one presentation or the entire orbit.

View exposure strengthens the combined recipe. Adding PR to RSM raises MeanSucc to 47.00% and estimated worst-view success to 30.50%. The sECR gain is 8.0 points ($[6.4, 9.6]$), compared with -0.3 points ($[-1.2, 0.6]$) when PR is added to canonical SFT: a positive recipe-level interaction of 8.3 points ($[6.6, 10.1]$). RSM+PR also improves sECR over canonical SFT by 1.9 points ($[0.6, 3.1]$). Its real-minus-pseudo spread falls from 11.11 points under RSM to 4.05, while the worst-view real-minus-pseudo contrast moves from -6.04 to -2.34. The combination retains the solving gains and reduces the measured cross-view variation.

Controls for puzzle sampling, mask density, and target weighting preserve a positive Sudoku sECR interaction at 50M and 100M, while RSM+PR and random-state+PR finish at nearly the same normalized reliability; at 50M, RSM+PR retains modest mean- and worst-view-success gains (Appendix C).

The row–column distinction. The view spectrum localizes the added token-level disagreement in the initial study. About two-thirds of the increase from canonical SFT to RSM lies in the D_4 axis mode, which separates axis-preserving transformations from those exchanging rows and columns. Its share rises from 16.2% to 19.8%, then falls to 17.5% with PR (Appendix B.5).

Beyond D_4. The recipe also works with digit relabeling and row, column, and band permutations. RSM+PR has the highest mean success and highest or tied-highest sECR in each group-specific comparison; PR improves sECR over RSM in all four groups (Appendix B.6). Digit relabeling leaves every cell in place, extending the observation beyond geometric changes.

5.5 LARGER MODELS AND A SECOND TASK

LLaDA-8B. With LoRA post-training on Sudoku-AC48, all-view correctness rises from 47.7% under canonical SFT to 74.8% under RSM and 86.1% under RSM+PR. Average accuracy rises alongside it, from 88.2% to 94.0% and 96.6%. PR without RSM reaches 46.0% ALL. These recipe-level results show a larger set of problems solved correctly across all eight recorded views (Appendix B.8).

Maze completion. Walls and start/end markers remain visible while the model completes corridor targets. The baseline uses random-ratio masking, with or without PR; RSM uses intermediate reasoning states. After 5,000 post-training steps, we evaluate 100 held-out mazes under eight views with $K = 16$ samples per view. On the harder setting, RSM+PR exceeds random-mask+PR in all six initialization settings, by 2.7–11.2 points in mean success and 7.3–18.8 points in estimated worst-view success (Table 5). Much of the improvement is already present under RSM alone. Adding PR further raises estimated worst-view success in five of six settings. Appendix B.7 gives all recipe results, including the easier setting and sampled joint success.

Table 5: **Recipe utility on harder maze completion (n15).** Point estimates in percent. Parentheses give RSM+PR minus random-mask+PR. Models are initialized from the listed pretraining steps; both recipes use 5,000 post-training steps.

		MeanSucc		$\widehat{R}_{\min}$	
Model	Step	Random+PR	RSM+PR	Random+PR	RSM+PR
50M	100k	91.33	97.46 (+6.13)	80.44	92.50 (+12.06)
	150k	94.21	98.70 (+4.49)	85.38	96.12 (+10.74)
	200k	96.54	99.26 (+2.72)	90.25	97.56 (+7.31)
100M	100k	83.53	94.68 (+11.15)	67.94	86.75 (+18.81)
	150k	88.28	95.42 (+7.14)	75.38	88.75 (+13.37)
	200k	90.46	96.97 (+6.51)	78.81	91.50 (+12.69)

6 Limitations and Discussion

Our post-training experiments focus on grid-structured tasks, where exact symmetry groups permit precise orbit evaluation; the code and mathematics evaluations show that presentation dependence extends further, and applying RSM with verified intermediate facts or execution traces in these domains is a natural next step. Sampled orbit metrics depend on the sampling budget, which we address with pseudo-orbit calibration and paired intervals. Finally, the rigidity theorem concerns a single attention component; the positional pilot in Appendix B.9 suggests that combining symmetry-aware positional indexing with RSM+PR is a promising direction.

7 Conclusion

A model can gain accuracy while losing answers it previously got right. We expose this failure through equivalent presentations, establish an exact positional characterization, and study how training states and views affect solving. In the initial Sudoku recipe study, RSM improves competence while PR strengthens reliability; their combination improves both. The recipe also yields gains at 8B scale and on maze completion. The goal is to solve more problems and retain those successes when the same problems are written differently.

AI use statement

In this work, we used generative AI tools for writing assistance, text editing, and code debugging. We have not used generative AI tools for research idea generation, experimental design, or data collection, and peer review assistance or data analysis tasks are not applicable to this work. Additionally, we used generative AI tools for grammar checking and polishing of the manuscript. We have reviewed all AI-assisted work. Specifically, all AI-generated or AI-revised text and code were manually inspected, verified, and tested for correctness and integrity by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

This work involves no human subjects or personal data. We use publicly available datasets (Kaggle-3M Sudoku, HumanEval-Infilling, MATH-500) and publicly released models under their licenses;

the transformed evaluation items are generated automatically from these datasets. Improved scores on our benchmark should not be read as a guarantee of presentation-stable behavior in other domains.

Reproducibility statement

Metric definitions are given in Section 3.3 and Table 6. Complete proofs are in Appendix A. Data construction, training configurations, and evaluation protocols are described in Section 5.1 and Appendices C–D. The supplementary material provides anonymized code for the model backbone, Sudoku post-training (canonical SFT, PR, and RSM), and orbit evaluation, including the pseudo-orbit control and paired bootstrap intervals.

References

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Shuxiao Chen, Edgar Dobriban, and Jane H Lee. A group-theoretic framework for data augmentation. *Journal of Machine Learning Research*, 21(245):1–71, 2020.

Steven Cho, Stefano Ruberto, and Valerio Terragni. Metamorphic testing of large language models for natural language processing. In *2025 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 174–186. IEEE, 2025. doi: 10.1109/icsme64153.2025.00025. URL http://dx.doi.org/10.1109/ICSME64153.2025.00025.

Taco S. Cohen and Max Welling. Group equivariant convolutional networks, 2016. URL https://arxiv.org/abs/1602.07576.

Jia Deng, Junyi Li, Wayne Xin Zhao, Jinpeng Wang, Hongyu Lu, and Ji-Rong Wen. Beyond fully random masking: Attention-guided denoising and optimization for diffusion language models, 2026. URL https://arxiv.org/abs/2606.12273.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models, 2021. URL https://arxiv.org/abs/2102.01017.

Tom Heskes. Selecting weighting factors in logarithmic opinion pools. In *Proceedings of the 11th International Conference on Neural Information Processing Systems*, NIPS'97, pp. 266–272, Cambridge, MA, USA, 1997. MIT Press.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *arXiv preprint arXiv:2502.06768*, 2025a.

Seo Hyun Kim, Sunwoo Hong, Hojung Jung, Youngrok Park, and Se-Young Yun. KLASS: KL-guided fast inference in masked diffusion models. *Advances in Neural Information Processing Systems*, 38:92267–92301, 2025b.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023. URL https://arxiv.org/abs/2307.03172.

Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stärk, Yilun Xu, Tommi Jaakkola, and Rafael Gómez-Bombarelli. Think while you generate: Discrete diffusion with planned denoising. *arXiv preprint arXiv:2410.06264*, 2024.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity, 2022. URL https://arxiv.org/abs/2104.08786.

Zanlin Ni, Shenzhi Wang, Yang Yue, Tianyu Yu, Weilin Zhao, Yeguo Hua, Tianyi Chen, Jun Song, Cheng Yu, Bo Zheng, et al. The flexibility trap: Rethinking the value of arbitrary order in diffusion language models. *arXiv preprint arXiv:2601.15165*, 2026.

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. *arXiv preprint arXiv:2502.09992*, 2025a.

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, 2025b. URL https://arxiv.org/abs/2502.09992.

Shen Nie, Qiyang Min, Shaoxuan Xu, Zihao Huang, Yuxuan Song, Yong Shan, Yankai Lin, Wayne Xin Zhao, Chongxuan Li, and Ji-Rong Wen. Improved large language diffusion models. *arXiv preprint arXiv:2606.25331*, 2026.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling. *arXiv preprint arXiv:2502.03540*, 2025.

Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Equivariance through parameter-sharing, 2017. URL https://arxiv.org/abs/1702.08389.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist, 2020. URL https://arxiv.org/abs/2005.04118.

David W. Romero and Jean-Baptiste Cordonnier. Group equivariant stand-alone self-attention for vision, 2021. URL https://arxiv.org/abs/2010.00977.

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

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting, 2024. URL https://arxiv.org/abs/2310.11324.

Jeffrey Seely, Yuki Imajuku, Tianyu Zhao, Edoardo Cetin, and Llion Jones. Sudoku-bench: Evaluating creative reasoning with sudoku variants, 2025. URL https://arxiv.org/abs/2505.16135.

Kulin Shah, Nishanth Dikkala, Xin Wang, and Rina Panigrahy. Causal language modeling can elicit search and reasoning capabilities on logic puzzles, 2024. URL https://arxiv.org/abs/2409.10502.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Andy Shih, Dorsa Sadigh, and Stefano Ermon. Training and inference on any-order autoregressive models the right way, 2022. URL https://arxiv.org/abs/2205.13554.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Shiqi Wang, Zheng Li, Haifeng Qian, Chenghao Yang, Zijian Wang, Mingyue Shang, Varun Kumar, Samson Tan, Baishakhi Ray, Parminder Bhatia, Ramesh Nallapati, Murali Krishna Ramanathan, Dan Roth, and Bing Xiang. Recode: Robustness evaluation of code generation models, 2022. URL https://arxiv.org/abs/2212.10264.

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

Fengqi Zhu, Zebin You, Yipeng Xing, Zenan Huang, Lin Liu, Yihong Zhuang, Guoshan Lu, Kangyu Wang, Xudong Wang, Lanning Wei, et al. LLaDA-MoE: A sparse MoE diffusion language model. *arXiv preprint arXiv:2509.24389*, 2025.

A Appendix

A Additional Proofs and Protocol Details

All logarithms are natural. The group size is denoted by $m = |G|$. The principal grid evaluation uses $G = D_4$ and $m = 8$; the additional-group experiments specify their own transformation families. The output space includes the invalid parse symbol $\perp$, with $g\perp = \perp$. All probability statements below are for a fixed base logical state x unless stated otherwise.

A.1 Forced-target validity of SUDOKU-AC-ORBIT

This appendix proves the claim used in Section 3: if the target cells are forced in the canonical board, then the corresponding target cells are forced in every rotated or reflected view. Thus every view has the same transported target and an isomorphic constraint system.

Let $\Lambda = \{1, \dots, 9\}^2$ be the set of Sudoku cells. A completed Sudoku solution is a function $s : \Lambda \to \{1, \dots, 9\}$ satisfying the row, column, and 3×3 box constraints. For a set of anchors A, let $\mathcal{S}(A)$ be the set of valid completed Sudoku grids that agree with the anchored digits. For $U \subseteq \Lambda$, write s_U for the restriction of s to U.

Proposition A.1 (Forced targets are preserved under D_4)**.** *Let $g \in D_4$. Then*

$$\mathcal{S}(gA) = g\mathcal{S}(A) := \{gs : s \in \mathcal{S}(A)\}. \tag{14}$$

Consequently, if

$$\{s_U : s \in \mathcal{S}(A)\} = \{s^\star_U\}, \tag{15}$$

then

$$\{t_{gU} : t \in \mathcal{S}(gA)\} = \{(gs^\star)_{gU}\}. \tag{16}$$

Proof. Every element of D_4 is a rotation or reflection of the 9×9 square. It maps rows to rows or columns, columns to columns or rows, and 3×3 boxes to 3×3 boxes. Thus s satisfies the Sudoku constraints if and only if gs satisfies the Sudoku constraints. Also, s agrees with anchors A if and only if gs agrees with transformed anchors gA. Hence $\mathcal{S}(gA) = g\mathcal{S}(A)$.

If $t \in \mathcal{S}(gA)$, then by the first part there exists $s \in \mathcal{S}(A)$ with $t = gs$. Therefore $t_{gU} = (gs)_{gU} = g(s_U)$. Since $s_U = s^\star_U$ for all $s \in \mathcal{S}(A)$, the transformed restriction is always $(gs^\star)_{gU}$. This proves target uniqueness in every view. □

A.2 View leakage and the output view spectrum

We connect distributional logical equivariance to view leakage and its spectrum. Zero view leakage is exactly distributional logical equivariance on the orbit of x, and the χ^2-version of view leakage

decomposes exactly into the nontrivial Fourier modes of the group. The KL leakage L_{out} is not itself an exact Fourier norm; it is equivalent to that Fourier norm up to constants depending only on $|G|$.

Sample $V \sim \mathrm{Unif}(G)$, generate $Y \sim K_\theta(\cdot \mid Vx)$, and canonicalize $Y^c = V^{-1}Y$. Let

$$P_g := g_{\#}^{-1} K_\theta(\cdot \mid gx), \qquad \overline{P} := \frac{1}{m} \sum_{g \in G} P_g. \tag{17}$$

Then the conditional law of Y^c given $V = g$ is P_g, and the marginal law of Y^c is $\overline{P}$.

Proposition A.2 (Zero leakage is distributional LE)**.**

$$L_{\mathrm{out}}(x) := I(V; Y^c) = \frac{1}{m} \sum_{g \in G} D_{\mathrm{KL}}(P_g \| \overline{P}). \tag{18}$$

Moreover,

$$L_{\mathrm{out}}(x) = 0 \quad \Longleftrightarrow \quad P_g = P_h \quad \forall g, h \in G. \tag{19}$$

Proof. The joint law of (V, Y^c) is

$$\Pr(V = g, Y^c \in B) = \frac{1}{m} P_g(B). \tag{20}$$

The marginal law of Y^c is $\overline{P}$. Therefore

$$I(V; Y^c) = D_{\mathrm{KL}}(\Pr_{V, Y^c} \| \Pr_{V} \otimes \Pr_{Y^c}) = \frac{1}{m} \sum_{g \in G} D_{\mathrm{KL}}(P_g \| \overline{P}). \tag{21}$$

Mutual information is zero if and only if V and Y^c are independent. Since all views have positive probability, independence is equivalent to $P_g = \overline{P}$ for every g, which is equivalent to $P_g = P_h$ for all $g, h \in G$. □

We now decompose the view dependence by irreducible representations. Let $\widehat{G}$ be a set of unitary irreducible representations of G. For $\rho \in \widehat{G}$, let d_ρ be its dimension. The trivial representation is denoted by $\mathbf{1}$.

Because $\overline{P}$ is an average containing each P_g, we have $P_g \ll \overline{P}$. Define

$$r_g(y) := \frac{dP_g}{d\overline{P}}(y). \tag{22}$$

Then $0 \le r_g(y) \le m$ for $\overline{P}$-almost every y, and

$$\frac{1}{m} \sum_{g \in G} r_g(y) = 1. \tag{23}$$

For each y, view $g \mapsto r_g(y)$ as a function on the group, and define its Fourier coefficient by

$$\widehat{r}_\rho(y) := \frac{1}{m} \sum_{g \in G} r_g(y) \rho(g)^*. \tag{24}$$

Theorem A.3 (Output view spectrum)**.** *Define*

$$\chi^2_G(x) := \int \frac{1}{m} \sum_{g \in G} \big(r_g(y) - 1\big)^2 \, d\overline{P}(y). \tag{25}$$

Then

$$\chi^2_G(x) = \sum_{\rho \neq \mathbf{1}} d_\rho \int \|\widehat{r}_\rho(y)\|^2_{\mathrm{HS}} \, d\overline{P}(y). \tag{26}$$

Furthermore, there exist constants $0 < a_m \le b_m < \infty$, depending only on $m = |G|$, such that

$$a_m \chi^2_G(x) \le L_{\mathrm{out}}(x) \le b_m \chi^2_G(x). \tag{27}$$

Proof. For each fixed y, apply the normalized Plancherel identity on the finite group:

$$\frac{1}{m}\sum_{g\in G}|f(g)|^2=\sum_{\rho\in\widehat{G}}d_\rho\|\widehat{f}(\rho)\|_{\mathrm{HS}}^2,\qquad \widehat{f}(\rho)=\frac{1}{m}\sum_{g\in G}f(g)\rho(g)^*. \tag{28}$$

Use $f(g)=r_g(y)$. The trivial coefficient is

$$\widehat{r}_{\mathbf{1}}(y)=\frac{1}{m}\sum_{g\in G}r_g(y)=1. \tag{29}$$

Thus Plancherel gives

$$\frac{1}{m}\sum_{g\in G}\big(r_g(y)-1\big)^2=\sum_{\rho\neq\mathbf{1}}d_\rho\|\widehat{r}_\rho(y)\|_{\mathrm{HS}}^2. \tag{30}$$

Integrating over $\overline{P}$ proves Eq. (26).

For the KL comparison, let

$$\varphi(t):=t\log t-t+1,\qquad \varphi(0):=1. \tag{31}$$

Since $\int r_g\,d\overline{P}=1$,

$$D_{\mathrm{KL}}(P_g\|\overline{P})=\int r_g\log r_g\,d\overline{P}=\int\varphi(r_g)\,d\overline{P}. \tag{32}$$

The ratio $\varphi(t)/(t-1)^2$, with value $1/2$ at $t=1$ by continuous extension, is positive and finite on the compact interval $[0,m]$. Hence there are constants a_m,b_m such that

$$a_m(t-1)^2\le\varphi(t)\le b_m(t-1)^2\qquad\forall t\in[0,m]. \tag{33}$$

Apply this pointwise to $t=r_g(y)$, average over g, and integrate over $\overline{P}$. □

The following proposition connects the spectrum to the correctness metrics. It is useful because it turns an observed Any–All gap into a lower bound on nontrivial view information.

Proposition A.4 (Orbit failure forces view leakage)**.** *For deterministic decoding,*

$$L_{\mathrm{out}}(x)\ge h_2(1/m)\big(\mathrm{Any}(x)-\mathrm{All}(x)\big), \tag{34}$$

where $h_2(p)=-p\log p-(1-p)\log(1-p)$. For stochastic decoding, if $p_g(x)=P_g(\{y^\star(x)\})$, then

$$L_{\mathrm{out}}(x)\ge\frac{(C_{\max}(x)-R_{\min}(x))^2}{m}. \tag{35}$$

Proof. For deterministic decoding, $Y^c=\widehat{y}_V(x)$ is a deterministic function of V, so $L_{\mathrm{out}}(x)=H(Y^c)$. If $\mathrm{Any}(x)-\mathrm{All}(x)=0$, the stated inequality is immediate. If the gap is one, then one canonicalized prediction equals the unique target and at least one does not. Hence Y^c is a nonconstant function of a uniform m-point variable. The minimum entropy of such a nonconstant image occurs when one fiber has size 1 and another has size $m-1$, giving $H(Y^c)\ge h_2(1/m)$.

For stochastic decoding, let $C=\mathbf{1}[Y^c=y^\star(x)]$. Data processing gives $I(V;Y^c)\ge I(V;C)$. With $\overline{p}=m^{-1}\sum_g p_g$,

$$I(V;C)=\frac{1}{m}\sum_{g\in G}\mathrm{kl}(p_g\|\overline{p}). \tag{36}$$

Pinsker's inequality for Bernoulli laws gives $\mathrm{kl}(p_g\|\overline{p})\ge 2(p_g-\overline{p})^2$. Let $\delta=C_{\max}-R_{\min}$. Choose views g_+,g_- attaining the maximum and minimum. Then

$$|p_{g_+}-\overline{p}|+|p_{g_-}-\overline{p}|\ge\delta, \tag{37}$$

and therefore

$$(p_{g_+}-\overline{p})^2+(p_{g_-}-\overline{p})^2\ge\delta^2/2. \tag{38}$$

Thus

$$L_{\mathrm{out}}(x)\ge I(V;C)\ge\frac{2}{m}\sum_g(p_g-\overline{p})^2\ge\frac{\delta^2}{m}. \tag{39}$$

□

A.3 ROW-MAJOR TOEPLITZ–D_4 RIGIDITY

We first establish the scalar offset-class result, then use it to prove the attention-head theorem in Section 4.4.

Let

$$\Lambda_n := \{0, \dots, n-1\}^2, \qquad p(r,c) := nr + c. \tag{40}$$

A scalar row-major relative-position kernel has the form

$$B_{uv} = a_{p(v)-p(u)}. \tag{41}$$

Let $\mathcal{T}_n$ be the vector space of all such kernels. Let

$$\mathcal{C}_n := \{B : B_{gu,gv} = B_{uv} \ \forall g \in D_4,\ u, v \in \Lambda_n\} \tag{42}$$

be the space of D_4-invariant scalar pair kernels.

Theorem A.5 (Row-major Toeplitz–D_4 rigidity)**.** *For $n \geq 2$,*

$$\dim(\mathcal{T}_n \cap \mathcal{C}_n) = \begin{cases} 2, & n \text{ even}, \\ 3, & n \text{ odd}. \end{cases} \tag{43}$$

More explicitly, if n is even, then every $B \in \mathcal{T}_n \cap \mathcal{C}_n$ has the form

$$B_{uv} = \alpha \mathbf{1}[p(v) - p(u) = 0] + \beta \mathbf{1}[p(v) - p(u) \neq 0]. \tag{44}$$

If n is odd, then every $B \in \mathcal{T}_n \cap \mathcal{C}_n$ has the form

$$B_{uv} = \alpha \mathbf{1}[d = 0] + \beta \mathbf{1}[d \neq 0 \text{ and } d \text{ even}] + \gamma \mathbf{1}[d \text{ odd}], \qquad d = p(v) - p(u). \tag{45}$$

In particular, for the 9×9 Sudoku grid, the certified row-major scalar relative-position algebra has only the three classes 0, nonzero-even, and odd.

Proof. For a pair of cells $u = (r, c)$ and $v = (r', c')$, define the displacement

$$(x, y) := (r' - r, c' - c), \qquad x, y \in \{-(n-1), \dots, n-1\}. \tag{46}$$

The row-major offset is

$$\delta(x, y) := p(v) - p(u) = nx + y. \tag{47}$$

The group D_4 acts on displacements by sign changes and coordinate swaps:

$$(x, y) \mapsto (\pm x, \pm y), \qquad (x, y) \mapsto (\pm y, \pm x), \tag{48}$$

with the signs constrained by the chosen rotation or reflection.

The only subtle point is that the same offset may have several feasible displacement representations. For example, $nx + y = nx' + y'$ can hold with $(x, y) \neq (x', y')$. Define an equivalence relation $\sim$ on offsets by the following generators: two offsets are equivalent if they are the same integer, or if they arise from feasible displacements related by a D_4 transformation. Then $B \in \mathcal{T}_n \cap \mathcal{C}_n$ exactly means $a_d = a_{d'}$ whenever $d \sim d'$. Thus the dimension of $\mathcal{T}_n \cap \mathcal{C}_n$ is the number of equivalence classes of offsets.

First, offset 0 is a singleton class. If $nx + y = 0$ with $|x|, |y| \leq n - 1$, then $x \neq 0$ would imply $|nx| \geq n > |y|$, impossible. Hence $x = y = 0$.

Second, every nonzero offset is equivalent to one of $1, \dots, n-1$. By a 180° rotation, $d \sim -d$, so take $d > 0$. Write $d = nq + r$ with $0 \leq r \leq n - 1$. If $q = 0$, then $d = r \in \{1, \dots, n-1\}$. If $q \geq 1$ and $r = 0$, then

$$d = \delta(q, 0) \sim \delta(0, q) = q < d. \tag{49}$$

If $q \geq 1$ and $r > 0$, then

$$d = \delta(q, r) \sim \delta(q, -r) = nq - r < d. \tag{50}$$

Repeating this strict descent reaches a positive offset in $\{1, \dots, n-1\}$.

Third, for every $1 \leq k \leq n - 3$,

$$k \sim k + 2. \tag{51}$$

Indeed,

$$\begin{aligned} k &= \delta(1, k-n) && (52)\\ &\sim \delta(n-k, -1) && (53)\\ &= n(n-k) - 1 && (54)\\ &= \delta(n-k-1, n-1) && (55)\\ &\sim \delta(-(n-k-1), n-1) && (56)\\ &= \delta(-(n-k-2), -1) && (57)\\ &\sim \delta(1, k+2-n) && (58)\\ &= k+2. && (59) \end{aligned}$$

All displacements used above lie in $[-(n-1), n-1]^2$ when $1 \le k \le n-3$.

Now suppose n is odd. Since

$$\delta(x, y) = nx + y \equiv x + y \pmod 2, \tag{60}$$

and D_4 only changes signs and swaps coordinates, offset parity is invariant. By Eq. (51), all positive small odd offsets are equivalent and all positive small even offsets are equivalent. The descent argument then shows that all nonzero odd offsets form one class and all nonzero even offsets form one class. Together with the singleton class $\{0\}$, this gives three classes.

Finally suppose n is even. For $n = 2$, direct inspection gives $1 = \delta(0, 1) \sim \delta(1, 0) = 2$, so all nonzero offsets are equivalent. For $n \ge 4$,

$$1 = \delta(1, 1-n) \sim \delta(1, n-1) = 2n - 1 = \delta(2, -1) \sim \delta(1, -2) = n - 2. \tag{61}$$

The offset $n-2$ is even. By Eq. (51), it is equivalent to 2. Hence $1 \sim 2$, so the odd and even nonzero classes merge. The descent argument then shows that all nonzero offsets form one class. Together with $\{0\}$, this gives two classes. □

Consequence for 1D RoPE. Let M_d be the matrix used by a 1D relative positional component at row-major offset d. For fixed query/key contents a, b, the scalar positional term has form

$$B^{a,b}_{uv} = a^* M_{p(v)-p(u)} b. \tag{62}$$

If this scalar term is D_4-invariant for every a, b, then Theorem A.5 forces $M_d = M_{d'}$ whenever d and d' lie in the same offset class. On a 9×9 grid, this means all nonzero odd offsets must share the same matrix, including 1, 9, and -9. Standard nondegenerate 1D RoPE separates these offsets. Therefore row-major 1D RoPE does not provide a component-level scalar D_4 positional certificate.

This is a component-level constraint on the positional matrices.

A.4 Row-major relative-score attention rigidity

We now establish Theorem 4.1 for full, unmasked relative-score attention.

Let

$$\Lambda_n := \{0, \dots, n-1\}^2, \qquad p(r, c) := nr + c,$$

and define the row-major offset

$$d(u, v) := p(v) - p(u).$$

Let D_4 act on Λ_n, and let Π_g denote the induced position-only permutation, with the convention

$$(\Pi_g Z)_{gu} = Z_u.$$

The action permutes positions only and does not transform hidden channels.

We first record the offset-class fact used in the main theorem.

Lemma A.6 (Row-major Toeplitz–D_4 offset classes)**.** *Let $\sim$ be the equivalence relation on row-major offsets generated by*

$$d(u, v) \sim d(gu, gv) \qquad \textit{for all } g \in D_4,\ u, v \in \Lambda_n.$$

Then, for $n \geq 2$, the offset 0 is a singleton class. For nonzero offsets:

$$\begin{cases} \textit{all nonzero offsets form one class}, & n \textit{ even}, \\ \textit{all nonzero-even offsets form one class and all odd offsets form one class}, & n \textit{ odd}. \end{cases}$$

In particular, for $n = 9$, any row-major offset-indexed object that is constant on D_4-related cell pairs must have the form

$$a_d = \begin{cases} a_0, & d = 0, \\ a_E, & d \neq 0,\ d \equiv 0 \pmod 2, \\ a_O, & d \equiv 1 \pmod 2. \end{cases}$$

Proof. The equivalence relation is exactly the one generated by the feasible displacement transformations in Theorem A.5. Its proof establishes the singleton zero class, descent to offsets $1, \ldots, n-1$, parity preservation for odd n, and merging of the two nonzero parity classes for even n. □

We now prove the attention-head rigidity theorem from the main text.

Proof of Theorem 4.1. Let

$$\Delta_n := \{d(u, v) : u, v \in \Lambda_n\}.$$

For each $d \in \Delta_n$, let

$$\ell_d : \mathbb{R}^{r_q} \times \mathbb{R}^{r_k} \to \mathbb{R}$$

be a finite scalar scoring function. The attention logits are

$$L_{uv}(Q, K) = \ell_{d(u,v)}(Q_u, K_v),$$

and the row-stochastic attention matrix is

$$A_\ell(Q, K) = \mathrm{softmax}_v(L(Q, K)).$$

Thus

$$\mathrm{Att}_\ell(Q, K, V) = A_\ell(Q, K)V.$$

Assume first that the attention head is exactly D_4-equivariant under the position-only action. Then, for every $g \in D_4$ and every value array V,

$$A_\ell(\Pi_g Q, \Pi_g K)\Pi_g V = \Pi_g A_\ell(Q, K)V.$$

Since this identity holds for all V, we may write $W = \Pi_g V$ and obtain

$$A_\ell(\Pi_g Q, \Pi_g K)W = \Pi_g A_\ell(Q, K)\Pi_g^{-1} W \qquad \forall W.$$

Therefore,

$$A_\ell(\Pi_g Q, \Pi_g K) = \Pi_g A_\ell(Q, K)\Pi_g^{-1}. \tag{63}$$

Equivalently, for every $u, v \in \Lambda_n$,

$$A_\ell(\Pi_g Q, \Pi_g K)_{gu,gv} = A_\ell(Q, K)_{u,v}. \tag{64}$$

For finite logits, equality of two row-wise softmax distributions is equivalent to equality of the corresponding logits up to an additive constant depending only on the row. Hence, for each fixed g, u, Q, K, there exists a scalar $c_{g,u}(Q, K)$ such that

$$L_{gu,gv}(\Pi_g Q, \Pi_g K) = L_{u,v}(Q, K) + c_{g,u}(Q, K) \qquad \forall v \in \Lambda_n. \tag{65}$$

Using the row-major relative-score form and the position-only action,

$$(\Pi_g Q)_{gu} = Q_u, \qquad (\Pi_g K)_{gv} = K_v,$$

so Eq. (65) becomes

$$\ell_{d(gu,gv)}(Q_u, K_v) = \ell_{d(u,v)}(Q_u, K_v) + c_{g,u}(Q, K) \qquad \forall v. \tag{66}$$

Because the attention head is full and unmasked, the self-edge $v = u$ is available. Setting $v = u$, we have

$$d(u, u) = d(gu, gu) = 0.$$

Therefore Eq. (66) gives

$$\ell_0(Q_u, K_u) = \ell_0(Q_u, K_u) + c_{g,u}(Q, K),$$

and hence

$$c_{g,u}(Q, K) = 0.$$

Substituting this back into Eq. (66),

$$\ell_{d(gu,gv)}(Q_u, K_v) = \ell_{d(u,v)}(Q_u, K_v) \qquad \forall g, u, v, Q, K.$$

Since the theorem assumes equivariance for all Q, K, the vectors Q_u and K_v can be varied arbitrarily. Thus

$$\ell_{d(gu,gv)}(q, k) = \ell_{d(u,v)}(q, k)$$

for every $g \in D_4$, every $u, v \in \Lambda_n$, and all $q \in \mathbb{R}^{r_q}$, $k \in \mathbb{R}^{r_k}$.

Conversely, suppose the score functions are tied on all D_4-related row-major offsets:

$$\ell_{d(gu,gv)}(q, k) = \ell_{d(u,v)}(q, k) \qquad \forall g, u, v, q, k.$$

Then

$$\begin{aligned} L_{gu,gv}(\Pi_g Q, \Pi_g K) &= \ell_{d(gu,gv)}((\Pi_g Q)_{gu}, (\Pi_g K)_{gv}) && (67)\\ &= \ell_{d(gu,gv)}(Q_u, K_v) && (68)\\ &= \ell_{d(u,v)}(Q_u, K_v) && (69)\\ &= L_{u,v}(Q, K). && (70) \end{aligned}$$

Hence

$$L(\Pi_g Q, \Pi_g K) = \Pi_g L(Q, K) \Pi_g^{-1}.$$

Row-wise softmax commutes with simultaneous row and column permutation, so

$$A_\ell(\Pi_g Q, \Pi_g K) = \Pi_g A_\ell(Q, K) \Pi_g^{-1}.$$

Multiplying by $\Pi_g V$, we get

$$\begin{aligned} \mathrm{Att}_\ell(\Pi_g Q, \Pi_g K, \Pi_g V) &= A_\ell(\Pi_g Q, \Pi_g K)\Pi_g V && (71)\\ &= \Pi_g A_\ell(Q, K) V && (72)\\ &= \Pi_g \,\mathrm{Att}_\ell(Q, K, V). && (73) \end{aligned}$$

This proves the equivalence.

For $n = 9$, Lemma A.6 gives exactly three row-major offset classes:

$$d = 0, \qquad d \neq 0,\ d \equiv 0 \pmod 2, \qquad d \equiv 1 \pmod 2.$$

Therefore the score functions must collapse as

$$\ell_d = \begin{cases} \ell_0, & d = 0, \\ \ell_E, & d \neq 0,\ d \equiv 0 \pmod 2, \\ \ell_O, & d \equiv 1 \pmod 2. \end{cases}$$

In particular, the offsets 1, 9, and -9 are all odd offsets on a 9×9 grid, so an exactly D_4-equivariant row-major relative-score attention head must tie their score functions. □

Corollary A.7 (Bilinear relative scores and RoPE)**.** *For $\ell_d(q, k) = q^\top M_d k + b_d$, exact equivariance under the assumptions of Theorem 4.1 is equivalent to tying M_d and b_d within the theorem's offset classes. In particular, row-major RoPE with $R_1 \neq R_9$ does not satisfy the 9×9 tying condition.*

Proof of Corollary A.7. Assume the relative score has the bilinear-affine form

$$\ell_d(q, k) = q^\top M_d k + b_d.$$

By Theorem 4.1, exact D_4-equivariance implies

$$q^\top M_{d(gu,gv)} k + b_{d(gu,gv)} = q^\top M_{d(u,v)} k + b_{d(u,v)}$$

for every $g \in D_4$, every $u, v \in \Lambda_n$, and all q, k.

Setting $q = 0$ gives

$$b_{d(gu,gv)} = b_{d(u,v)}.$$

After subtracting the equal biases, we obtain

$$q^\top \left(M_{d(gu,gv)} - M_{d(u,v)}\right) k = 0 \qquad \forall q, k.$$

Choosing q and k to range over standard basis vectors gives

$$M_{d(gu,gv)} = M_{d(u,v)}.$$

For $n = 9$, Lemma A.6 then implies

$$(M_d, b_d) = \begin{cases} (M_0, b_0), & d = 0, \\ (M_E, b_E), & d \neq 0,\ d \equiv 0 \pmod 2, \\ (M_O, b_O), & d \equiv 1 \pmod 2. \end{cases}$$

For row-major RoPE, let R_t denote the block-diagonal rotary matrix at sequence position t, with the usual property

$$R_a^\top R_b = R_{b-a}.$$

Applying RoPE to query and key vectors at row-major positions $p(u)$ and $p(v)$ gives, up to the conventional sign convention,

$$(R_{p(u)} q_u)^\top (R_{p(v)} k_v) = q_u^\top R_{p(u)}^\top R_{p(v)} k_v \tag{74}$$

$$= q_u^\top R_{p(v)-p(u)} k_v. \tag{75}$$

Thus row-major RoPE is a bilinear relative-score instance with

$$M_d = R_d$$

up to sign convention and the usual attention scaling constant, which can be absorbed into M_d. Therefore, if the RoPE relative matrices distinguish two offsets in the same D_4-offset class, such as 1 and 9 on a 9×9 grid, the RoPE score does not satisfy the tying condition required by Theorem 4.1. Hence such a nondegenerate row-major RoPE head does not by itself certify exact D_4-equivariance. □

Scope of the result. Theorem 4.1 is a component-level certificate theorem. It assumes a single full, unmasked attention head, a row-major relative-score form, and a position-only D_4 action on hidden states. It does not imply that a full multi-layer Transformer with MLPs, residual streams, normalization, multi-head interactions, or learned internal representations cannot learn a D_4-equivariant function from data. It says that standard row-major relative-score attention does not enforce the D_4 certificate by itself unless its offset dependence collapses to the offset classes above. If hidden channels transform under a nontrivial D_4 representation, or if a mask removes the self-edge $v = u$, the necessary tying condition changes.

A.5 Representation-wise denoising accounting

This appendix gives a denoising-accounting identity. The result is a posterior L^2 identity, not a KL identity. It tracks the nontrivial representation modes of the hidden view label through the reverse denoising chain.

Let $V \sim \mathrm{Unif}(G)$. For any random variable S, define the posterior likelihood ratio of the view label by

$$\alpha_S(g) := \frac{\Pr(V = g \mid S)}{\Pr(V = g)} = m \Pr(V = g \mid S). \tag{76}$$

Then $m^{-1} \sum_g \alpha_S(g) = 1$. For $\rho \in \widehat{G}$, define

$$\widehat{\alpha}_{S,\rho} := \frac{1}{m} \sum_{g \in G} \alpha_S(g) \rho(g)^*. \tag{77}$$

For nontrivial $\rho \neq \mathbf{1}$, define the ρ-view energy in S by

$$\mathcal{E}_\rho(S) := d_\rho \, \mathbb{E}_S \left[\|\widehat{\alpha}_{S,\rho}\|_{\mathrm{HS}}^2 \right]. \tag{78}$$

For two random variables S, T, define the conditional innovation

$$\mathcal{J}_\rho(T; V \mid S) := d_\rho \, \mathbb{E}_{S,T} \left[\|\widehat{\alpha}_{S,T,\rho} - \widehat{\alpha}_{S,\rho}\|_{\mathrm{HS}}^2 \right]. \tag{79}$$

Lemma A.8 (Posterior Pythagorean identity)**.** *For any S, T and any nontrivial ρ,*

$$\mathcal{E}_\rho(S,T) = \mathcal{E}_\rho(S) + \mathcal{J}_\rho(T; V \mid S). \tag{80}$$

The same identity holds with S and T interchanged.

Proof. Bayes posterior averaging gives

$$\alpha_S(g) = \mathbb{E}[\alpha_{S,T}(g) \mid S]. \tag{81}$$

Fourier transform in the group variable is linear, so

$$\widehat{\alpha}_{S,\rho} = \mathbb{E}[\widehat{\alpha}_{S,T,\rho} \mid S]. \tag{82}$$

The Hilbert-space conditional variance identity gives

$$\mathbb{E}\|A\|_{\mathrm{HS}}^2 = \mathbb{E}\|\mathbb{E}[A \mid S]\|_{\mathrm{HS}}^2 + \mathbb{E}\|A - \mathbb{E}[A \mid S]\|_{\mathrm{HS}}^2, \tag{83}$$

with $A = \widehat{\alpha}_{S,T,\rho}$. Multiplying by d_ρ proves the result. □

Now run the reverse denoising chain under input Vx and pull each state back to canonical coordinates:

$$Z_H^c, Z_{H-1}^c, \ldots, Z_0^c, Y^c. \tag{84}$$

For each step define

$$A_h^\rho := \mathcal{J}_\rho(Z_{h-1}^c; V \mid Z_h^c), \qquad B_h^\rho := \mathcal{J}_\rho(Z_h^c; V \mid Z_{h-1}^c), \tag{85}$$

and for the readout define

$$A_R^\rho := \mathcal{J}_\rho(Y^c; V \mid Z_0^c), \qquad B_R^\rho := \mathcal{J}_\rho(Z_0^c; V \mid Y^c). \tag{86}$$

Theorem A.9 (Representation-wise denoising accounting)**.** *For every nontrivial representation $\rho \neq \mathbf{1}$,*

$$\mathcal{E}_\rho(Y^c) = \mathcal{E}_\rho(Z_H^c) + \sum_{h=1}^{H} (A_h^\rho - B_h^\rho) + A_R^\rho - B_R^\rho. \tag{87}$$

Consequently,

$$\mathcal{E}_\rho(Y^c) \leq \mathcal{E}_\rho(Z_H^c) + \sum_{h=1}^{H} A_h^\rho + A_R^\rho. \tag{88}$$

Proof. Apply Lemma A.8 to $(S,T) = (Z_h^c, Z_{h-1}^c)$ and to $(S,T) = (Z_{h-1}^c, Z_h^c)$. The two expressions for $\mathcal{E}_\rho(Z_h^c, Z_{h-1}^c)$ give

$$\mathcal{E}_\rho(Z_{h-1}^c) - \mathcal{E}_\rho(Z_h^c) = A_h^\rho - B_h^\rho. \tag{89}$$

Summing over $h = 1, \ldots, H$ gives

$$\mathcal{E}_\rho(Z_0^c) = \mathcal{E}_\rho(Z_H^c) + \sum_{h=1}^{H} (A_h^\rho - B_h^\rho). \tag{90}$$

Apply the same argument to (Z_0^c, Y^c) to obtain

$$\mathcal{E}_\rho(Y^c) - \mathcal{E}_\rho(Z_0^c) = A_R^\rho - B_R^\rho. \tag{91}$$

Combining the two displays proves Eq. (87). Since all B_h^ρ and B_R^ρ are nonnegative by definition, dropping them proves Eq. (88). □

Corollary A.10 (A nonzero output mode requires a nonzero source)**.** *Assume the initial canonical noise state carries no ρ-view mode, $\mathcal{E}_\rho(Z_H^c) = 0$, and the readout injects no new ρ-view mode, $A_R^\rho = 0$. If $\mathcal{E}_\rho(Y^c) > 0$, then at least one reverse denoising step has $A_h^\rho > 0$. In particular,*

$$\max_{1\le h\le H} A_h^\rho \ge \frac{\mathcal{E}_\rho(Y^c)}{H}. \tag{92}$$

Proof. Under the assumptions, Eq. (88) becomes

$$\mathcal{E}_\rho(Y^c) \le \sum_{h=1}^{H} A_h^\rho. \tag{93}$$

If the left side is positive, the sum is positive, so at least one term is positive. The displayed lower bound follows by averaging. □

If the canonicalized reverse transition at step h is independent of V given Z_h^c, then $Z_{h-1}^c \perp V \mid Z_h^c$, which implies $A_h^\rho = 0$ for every nontrivial ρ. Thus $A_h^\rho > 0$ is a one-way witness: the transition cannot be view-independent on the posterior support. The converse need not hold, because a transition may depend on the view in a way that is outside the posterior support or invisible to the particular representation mode.

A.6 Finite-K extrema and the pseudo-orbit control

This appendix explains why the stochastic metrics are averaged over base states and why pseudo-orbits are reported. With K samples per view, empirical maxima and minima are biased: maxima tend to be too high and minima tend to be too low. This can create a gap even when all views have the same true success probability.

For fixed x, let $\widehat{p}_{g,K}(x)$ be the empirical mean of K Bernoulli success samples with mean $p_g(x)$. Hoeffding's inequality and a union bound give

$$\Pr\left[\max_{g\in G} |\widehat{p}_{g,K}(x) - p_g(x)| > \epsilon\right] \le 2|G|e^{-2K\epsilon^2}. \tag{94}$$

On the complementary event,

$$|\widehat{C}_{\max,K}(x) - C_{\max}(x)| \le \epsilon, \qquad |\widehat{R}_{\min,K}(x) - R_{\min}(x)| \le \epsilon. \tag{95}$$

This controls concentration, but it does not remove extrema bias.

The canonical pseudo-orbit supplies a same-input reference for finite-sample extrema. For each base state x, draw $|G|K$ samples from the canonical layout only. Randomly split them into $|G|$ groups of size K and compute the same max/min metrics as if those groups were views. Because all pseudo-views come from the same layout, any pseudo gap is due to finite-sample extrema. The real-minus-pseudo contrast compares the observed orbit against that same-input reference; it is not an unbiased correction for arbitrary heterogeneous view success rates.

Paired bootstrap intervals keep each base puzzle and its views together. The reported resampling units and training repetitions are specified with each study.

B Additional Results

B.1 Orbit metric definitions

All stochastic metrics are computed per base puzzle and then averaged. For a puzzle x, we first estimate the success rate $\widehat{p}_g(x)$ of each view g from its K samples. MeanSucc averages these rates over views, $\widehat{C}_{\max}$ takes the best view, and $\widehat{R}_{\min}$ takes the worst view, so that $\widehat{R}_{\min}(x) \le \mathrm{MeanSucc}(x) \le \widehat{C}_{\max}(x)$ for every puzzle. We average each quantity over puzzles and then form sECR as the ratio of the averaged worst-view and best-view success, rather than averaging per-puzzle ratios, which are undefined when a puzzle is never solved. Because the extrema are taken over finite samples, they are biased estimates of the true best- and worst-view rates; Appendix A.6 discusses this bias and the pseudo-orbit reference we use to calibrate it. Table 6 lists the metrics.

Table 6: **Orbit metrics.** SLS and sECR summarize how success varies across views; they do not test whether the full output distributions agree.

Metric	Meaning	Desired
MeanSucc	Average success across views	↑
$\widehat{C}_{\max}$	Estimated best-view success	↑
$\widehat{R}_{\min}$	Estimated worst-view success	↑
SLS	Absolute layout-sensitivity gap	↓
sECR	Relative performance retention	↑

B.2 Per-transformation breakdown for code and mathematics

Table 7: **Per-transformation ANY/ALL** (percentages) for the paired evaluations in Table 2.

Task	Transformation	LLaDA-8B	Dream-7B	iLLaDA-8B	LLaDA-MoE
Code infilling	Identifier renaming	59.4 / 34.1	65.9 / 34.9	43.1 / 17.6	62.8 / 32.4
	Operand exchange	55.7 / 34.3	61.3 / 37.7	41.6 / 20.0	58.9 / 32.6
	Statement swap	54.2 / 38.2	56.9 / 33.8	40.0 / 23.1	54.2 / 28.9
	Statement relocation	54.1 / 30.7	63.3 / 32.8	46.1 / 19.6	60.3 / 26.7
	Inert-line insertion	58.3 / 24.0	63.5 / 24.0	45.7 / 12.2	61.6 / 20.5
MATH-500	Variable renaming	38.3 / 20.4	22.6 / 11.5	34.7 / 19.6	45.1 / 23.0
	Premise permutation	31.8 / 11.4	25.0 / 9.1	25.6 / 14.0	45.5 / 18.2
	Identity-operation insertion	39.9 / 19.8	26.7 / 12.5	35.6 / 17.6	49.5 / 20.5

For LLaDA-8B under variable renaming, accuracy rises from 28.5% to 30.2% while 8.1% of all instances change from correct to wrong and 9.8% from wrong to correct; the corresponding ANY−ALL of 17.9 points equals $r + b$ in Eq. (4).

B.3 Autoregressive comparison

The clue-conditioned autoregressive comparison uses the same puzzle data ladder. Recorded ANY increases from 16.1% at 5M to 38.4% at 100M, while one-shot ALL is zero at the evaluated scales. These recorded-output summaries contextualize the model-family scope; they are not a controlled attribution of the MDLM result to one architectural component.

B.4 Confidence intervals for post-training

Table 8: **Complete initial 50M recipe diagnostics.** Success and spread values are percentages; token leakage is in nats. Real-minus-pseudo contrasts use a canonical repeated-input reference.

Recipe	MeanSucc	$\widehat{C}_{\max}$	$\widehat{R}_{\min}$	sECR	SLS	$\widehat{L}_{\text{tok}}$	RobustGap	SLSGap
Pretrained	31.53	47.29	17.41	36.82	29.88	0.0775	−0.51	1.24
Canonical 80k	41.58	57.79	25.94	44.88	31.85	0.0712	−1.69	2.79
Canonical 10k	43.12	59.76	27.43	45.89	32.34	0.0714	−1.56	3.64
PR	42.16	58.46	26.64	45.56	31.82	0.0709	−1.20	2.88
PR grouped	41.15	57.35	25.60	44.64	31.75	0.0709	−1.27	2.16
RSM	45.74	65.81	26.16	39.75	39.65	0.0783	−6.04	11.11
RSM+PR	47.00	63.89	30.50	47.74	33.39	0.0684	−2.34	4.05

Canonical-view success is MeanSucc[16] restricted to the identity view, the quantity a conventional single-layout protocol would report (43.82% for canonical SFT and 45.81% for RSM).

Table 9: **Post-training endpoints with 95% bootstrap intervals** (paired over base puzzles, 10,000 resamples). sECR and $\widehat{R}_{\min}$ are percentages; $\widehat{L}_{\mathrm{tok}}$ is in nats.

Recipe	sECR^{16}	$\widehat{R}^{16}_{\min}$	$\widehat{L}_{\mathrm{tok}}$
Pretrained	36.82 [34.44, 39.15]	17.41 [15.79, 19.09]	0.0775 [0.0755, 0.0794]
Canonical 80k	44.88 [42.56, 47.14]	25.94 [23.89, 28.01]	0.0712 [0.0690, 0.0734]
Canonical 10k	45.89 [43.58, 48.11]	27.43 [25.34, 29.58]	0.0714 [0.0691, 0.0737]
PR	45.56 [43.27, 47.78]	26.64 [24.56, 28.73]	0.0709 [0.0686, 0.0732]
PR grouped	44.64 [42.29, 46.88]	25.60 [23.56, 27.66]	0.0709 [0.0687, 0.0732]
RSM	39.75 [37.31, 42.12]	26.16 [24.05, 28.36]	0.0783 [0.0755, 0.0811]
RSM+PR	**47.74** [45.44, 50.02]	**30.50** [28.30, 32.77]	**0.0684** [0.0660, 0.0709]

Table 10: **Paired contrasts** (percentage points unless noted; 95% intervals). The interaction is (RSM+PR − RSM) − (PR − Canonical).

Contrast	Measure	Estimate
RSM − Canonical 10k	canonical-view success	$+2.0$ $[+0.7, +3.3]$
RSM − Canonical 10k	sECR	-6.1 $[-7.7, -4.6]$
RSM − Canonical 10k	$\widehat{L}_{\mathrm{tok}}$ (nats)	$+0.0069$ $[+0.0054, +0.0084]$
RSM+PR − RSM	sECR	$+8.0$ $[+6.4, +9.6]$
PR − Canonical 10k	sECR	-0.3 $[-1.2, +0.6]$
PR − Canonical 80k	sECR	$+0.7$ $[-0.1, +1.5]$
Interaction (vs. Canonical 10k)	sECR	$+8.3$ $[+6.6, +10.1]$
Interaction (vs. Canonical 80k)	sECR	$+7.3$ $[+5.6, +9.0]$
RSM+PR − Canonical 10k	sECR	$+1.9$ $[+0.6, +3.1]$

B.5 Spectral localization

Applying the decomposition of Theorem A.3 to token marginals splits the view-dependent energy $\widehat{\chi}^2_{\mathrm{tok}}$ across the four nontrivial irreducible representations of D_4: three sign characters (sgn, axis, diag) and the two-dimensional standard representation E. The axis character separates $\{e, r^2, \sigma_h, \sigma_v\}$, which keep rows as rows, from $\{r, r^3, \sigma_d, \sigma_{d'}\}$, which exchange rows and columns. An isotropic reference across the seven nontrivial Fourier dimensions assigns the Plancherel shares: 14.3% for each character and 57.1% for E.

Table 11: **Share of token-level view disagreement by D_4 mode** (percent of $\widehat{\chi}^2_{\mathrm{tok}}$; point estimates from the same token marginals and smoothing as Table 4).

Recipe	Total $\widehat{\chi}^2_{\mathrm{tok}}$	sgn	axis	diag	E
Pretrained	0.1703	13.9	15.3	14.0	56.8
Canonical 80k	0.1580	13.9	15.6	14.2	56.3
Canonical 10k	0.1585	13.7	16.2	13.9	56.3
PR	0.1576	13.8	15.7	14.0	56.6
PR grouped	0.1572	13.9	15.7	14.0	56.4
RSM	0.1709	12.6	**19.8**	12.6	54.9
RSM+PR	0.1510	13.4	17.5	13.8	55.3
Flat reference	—	14.3	14.3	14.3	57.1

From the rounded table entries, RSM's total excess $\widehat{\chi}^2_{\mathrm{tok}}$ over Canonical 10k is 0.0124, of which approximately 0.0082 lies in the axis mode. Thus about two-thirds of the increase is concentrated in this component. Adding PR reduces the axis share to 17.5% and the total estimate to 0.1510. These

are spectral point estimates; the flat allocation is a reference, not a fitted sampling-null confidence interval.

B.6 Additional Sudoku symmetry groups

We retrain the post-training suite from the same 50M checkpoint under four further answer-preserving groups: digit relabeling (G_{digit}, which leaves every cell in place), row permutations within each band (G_{row}), column permutations within each stack (G_{col}), and band permutations (G_{band}). For the larger groups, we sample eight group elements, matching $|D_4|$; for G_{band} we use all six. PR and RSM+PR draw views from the corresponding group.

Table 12: **Post-training under additional symmetry groups.** Each entry is MeanSucc[16] (%) / sECR[16] (fraction).

Recipe	G_{digit}	G_{row}	G_{col}	G_{band}
Pretrained	31.5 / .362	31.5 / .381	31.9 / .382	31.7 / .433
Canonical 80k	41.8 / .468	41.9 / .480	42.1 / .478	42.1 / .521
PR	42.0 / .468	43.1 / .491	43.3 / .490	43.2 / .534
RSM	46.1 / .411	46.0 / .463	45.6 / .488	46.1 / .512
RSM+PR	**47.2 / .468**	**47.2 / .495**	**46.5 / .505**	**46.4 / .537**

In every group, RSM+PR has the highest mean success and the highest or tied-highest sECR, and PR raises sECR more on top of RSM ($+.057$, $+.032$, $+.017$, $+.025$) than on top of canonical SFT ($+.000$, $+.011$, $+.012$, $+.013$). Digit relabeling leaves the serialization unchanged, so the comparison also covers changes in symbol identity without moving cells.

B.7 Maze completion

Mazes are completed on a grid in which walls and start/end markers remain visible and corridor cells are the targets. A_0 masks a random proportion of corridor cells in the canonical orientation; PR (B_0) trains on an eight-view-expanded dataset; RSM trains on cached intermediate states from the maze solver with dense CE on every still-hidden cell; RSM+PR adds one online random D_4 view per instance. Each recipe post-trains for 5,000 steps from pretraining checkpoints at the listed steps and is evaluated on 100 held-out mazes with eight views and $K = 16$ samples per view. The decoder uses 128 steps and temperature 1.0. With $S_{x,g,k}$ indicating full exact success, sampled joint success is

$$\widehat{J}_{8,16} = \frac{1}{16N} \sum_{x=1}^{N} \sum_{k=1}^{16} \prod_{g \in D_4} S_{x,g,k}. \tag{96}$$

This repeats the experiment of drawing one answer per view; K is an estimation budget, not pass@K. Acc[1] equals MeanSucc here. Under conditional independence across views, the expectation is $\mathbb{E}_x \prod_g p_g(x)$, which becomes $\mathbb{E}_x p(x)^8$ when all view success probabilities agree. Consequently, sampled joint success is distinct from deterministic all-view correctness.

On `n15`, RSM+PR exceeds random-mask+PR by 2.72–11.15 points in MeanSucc and 7.31–18.81 points in estimated worst-view success. Much of the improvement is already present with RSM alone. Adding PR increases $\widehat{R}_{\min}$ in five of six settings and lowers it by 0.87 points at 100M/150k. On `n10`, RSM+PR exceeds random-mask+PR in both success measures at every initialization. The comparisons across checkpoints describe initialization dependence, not independent training-seed replications.

B.8 8B recipe results

B.9 Positional-index pilot

Theorem 4.1 characterizes which positional indexings certify D_4-equivariance. In a small pilot, we replace the row-major RoPE index of a small MDLM with indexings tied to D_4 orbits: the

Table 13: **Maze completion: harder (`n15`).** All entries are percentage point estimates. $\widehat{J}_{8,16}$ is sampled joint success, not deterministic ALL.

Model	Step	Recipe	MeanSucc	$\widehat{R}_{\min}$	sECR	$\widehat{J}_{8,16}$
50M	100k	Random (A_0)	91.33	81.00	83.99	71.69
		Random+PR (B_0)	91.33	80.44	82.98	69.44
		RSM	96.97	91.94	92.52	88.12
		RSM+PR	97.46	92.50	92.85	89.19
50M	150k	Random (A_0)	94.26	85.19	86.38	78.81
		Random+PR (B_0)	94.21	85.38	86.95	78.94
		RSM	98.20	94.19	94.31	91.81
		RSM+PR	98.70	96.12	96.19	94.12
50M	200k	Random (A_0)	96.63	90.81	91.90	87.12
		Random+PR (B_0)	96.54	90.25	90.87	85.06
		RSM	98.97	95.38	95.38	93.81
		RSM+PR	99.26	97.56	97.56	95.81
100M	100k	Random (A_0)	81.81	66.06	71.52	46.00
		Random+PR (B_0)	83.53	67.94	72.56	50.56
		RSM	94.38	85.25	86.44	79.88
		RSM+PR	94.68	86.75	87.96	81.25
100M	150k	Random (A_0)	87.75	75.62	78.93	60.31
		Random+PR (B_0)	88.28	75.38	78.72	60.88
		RSM	95.65	89.62	90.76	84.00
		RSM+PR	95.42	88.75	89.87	84.06
100M	200k	Random (A_0)	89.75	76.81	79.44	64.50
		Random+PR (B_0)	90.46	78.81	81.67	68.38
		RSM	96.77	90.81	91.44	87.00
		RSM+PR	96.97	91.50	92.02	87.56

Table 14: **Maze completion: easier (`n10`).** All entries are percentage point estimates. $\widehat{J}_{8,16}$ is sampled joint success, not deterministic ALL.

Model	Step	Recipe	MeanSucc	$\widehat{R}_{\min}$	$\widehat{J}_{8,16}$
50M	30k	Random (A_0)	97.42	92.06	87.25
		Random+PR (B_0)	97.33	92.19	86.62
		RSM	99.48	97.56	96.62
		RSM+PR	99.70	98.50	97.75
50M	100k	Random (A_0)	99.35	97.19	95.81
		Random+PR (B_0)	99.54	97.88	96.62
		RSM	99.66	98.12	97.62
		RSM+PR	99.88	99.31	99.00
50M	150k	Random (A_0)	99.62	98.31	97.31
		Random+PR (B_0)	99.80	98.88	98.50
		RSM	99.83	99.00	98.75
		RSM+PR	99.92	99.44	99.38
100M	100k	Random (A_0)	98.91	96.12	93.75
		Random+PR (B_0)	99.03	96.50	94.56
		RSM	99.74	98.75	98.12
		RSM+PR	99.73	98.44	97.94
100M	150k	Random (A_0)	99.09	96.19	94.25
		Random+PR (B_0)	99.46	97.19	96.31
		RSM	99.77	98.62	98.31
		RSM+PR	99.87	99.25	98.94
100M	200k	Random (A_0)	99.34	97.12	95.75
		Random+PR (B_0)	99.44	97.50	96.56
		RSM	99.85	99.19	98.88
		RSM+PR	99.95	99.69	99.56

Table 15: **RSM+PR on LLaDA-8B-Instruct** (LoRA, Sudoku-AC48, percentages). Each branch trains for 8,000 steps; checkpoints are selected on validation.

Recipe	Acc. ↑	ANY ↑	ALL ↑
Canonical SFT	88.2	99.8	47.7
PR	87.2	100.0	46.0
RSM	94.0	100.0	74.8
RSM+PR	**96.6**	**100.0**	**86.1**

three row-major offset classes of the theorem (tied 1D), or the 45 unordered displacement classes $\{|\Delta r|, |\Delta c|\}$ of a 9×9 grid (tied 2D). All other components, data, and optimization are held fixed.

Table 16: **Positional-index pilot** (200 states, $K = 8$; percentages except sECR, a fraction).

Scale	Positional index	MeanSucc	$\widehat{C}_{\max}$	$\widehat{R}_{\min}$	sECR
5M	Row-major RoPE	8.7	23.6	0.6	0.026
5M	Tied 1D (3 classes)	22.2	42.1	6.8	0.162
5M	Tied 2D (45 classes)	**25.6**	**46.8**	**8.9**	**0.191**
10M	Row-major RoPE	19.0	38.4	5.1	0.132
10M	Tied 1D (3 classes)	24.9	46.7	8.2	0.175
10M	Tied 2D (45 classes)	**28.9**	**50.1**	**11.1**	**0.222**

Symmetry-tied indexing raises worst-view success and sECR at both scales, and enlarging the certified vocabulary from 3 to 45 classes helps further. The tied models also solve more puzzles on average, so this pilot does not separate stability from competence; it indicates that positional mechanisms respecting the task symmetry are a promising complement to RSM+PR.

C ROBUSTNESS OF THE RSM–PR INTERACTION TO MATCHED CONTROLS

The recipe study in Section 5 compares complete training recipes, which differ in their puzzle subsets, masking densities, and target weighting. This appendix holds these factors fixed and asks two questions: whether the RSM–PR interaction persists, and what reasoning-state structure contributes beyond the distribution of partial states it induces.

C.1 DESIGN

For puzzle e, let T_e be its number of cached nonterminal reasoning states and N_e its number of initially hidden cells. All arms sample uniformly from the pooled state cache, so that

$$\nu(e) = \frac{T_e}{\sum_j T_j}, \qquad \pi_e(t) = \frac{1}{T_e}.$$

Arms A and B hide a uniformly random subset of original blanks with the same cardinality as the sampled reasoning state; original clues remain visible. Arms C and D use the reasoning state itself. Arm E uses the random subset of B but reweights each target by the cumulative coefficient ω_i of Eq. (10):

$$\ell_E = \frac{1}{|H|} \sum_{i \in H} N_e \, \omega_i \, \mathrm{CE}_i.$$

Because $\Pr(i \in H \mid |H|) = |H|/N_e$, the expected coefficient of each target equals ω_i; E therefore matches marginal target weights while randomizing which cells are visible together. A and C use the canonical orientation, and B, D, and E draw one view uniformly from D_4 per instance, transporting inputs, targets, and coefficients together.

Both studies start from a pretrained checkpoint and train three fine-tuning seeds per arm for 9,000 updates at batch size 512, using AdamW with learning rate 10^{-5}, 150 warmup steps, a final learning

rate of 10^{-6}, weight decay 0.01, and gradient clipping at 1.0. The cache contains 67,551 states from 9,995 of 10,000 puzzles (five empty trajectories are excluded). All arms use dense CE over every still-hidden cell at diffusion time 1.0, and we evaluate the final checkpoint. Evaluation uses the same 500 AC24 inputs for every run and scores the full completed board; the decoder uses 128 steps, temperature 1.0, and $K = 16$ samples per view. Intervals are paired bootstrap intervals over test puzzles, conditional on the trained runs.

C.2 RESULTS

Table 17: **Matched Sudoku study.** Three-seed averages, in percent. Mean: MeanSucc[16].

		50M			100M		
Arm	States / presentation	Mean	$\widehat{R}_{\min}$	sECR	Mean	$\widehat{R}_{\min}$	sECR
A	Random / canonical	44.31	26.60	42.98	40.00	22.88	39.52
B	Random / PR	42.86	27.01	45.74	39.17	23.63	42.73
C	RSM / canonical	43.63	24.47	38.61	38.05	19.48	33.39
D	RSM / PR	43.82	**27.67**	45.67	39.49	**23.84**	42.76
E	Weight-matched random / PR	42.97	27.07	45.81	39.14	23.69	43.02

Table 18: **Paired contrasts in the matched study** (percentage points, 95% intervals). The interaction is $(D - C) - (B - A)$. †Point estimate from Table 17.

Contrast	Measure	50M	100M
C−A	sECR	$-4.36\ [-5.75, -3.07]$	$-6.13\ [-7.50, -4.78]$
C−A	$\widehat{R}_{\min}$	$-2.13^{\dagger}$	$-3.40\ [-4.19, -2.62]$
Interaction	sECR	$+4.30\ [+2.88, +5.76]$	$+6.16\ [+4.63, +7.73]$
Interaction	$\widehat{R}_{\min}$	$+2.80\ [+1.96, +3.63]$	$+3.60\ [+2.77, +4.44]$
Interaction	Mean	$+1.64\ [+1.14, +2.12]$	$+2.27\ [+1.72, +2.84]$
D−B	Mean	$+0.96\ [+0.62, +1.33]$	$+0.32\ [-0.03, +0.68]$
D−B	$\widehat{R}_{\min}$	$+0.66\ [+0.10, +1.22]$	$+0.20\ [-0.34, +0.74]$
D−B	sECR	$-0.07\ [-1.01, +0.83]$	$+0.04\ [-1.01, +1.06]$
D−E	Mean	$+0.85\ [+0.52, +1.20]$	$+0.35\ [+0.00, +0.71]$
D−E	$\widehat{R}_{\min}$	$+0.60\ [+0.04, +1.15]$	$+0.15\ [-0.39, +0.69]$
D−E	sECR	$-0.14\ [-1.06, +0.75]$	$-0.26\ [-1.29, +0.79]$

RSM alone reduces stability, even against matched random states. Without PR, reasoning states lower sECR by 4.36 points at 50M and 6.13 points at 100M relative to random states with the same puzzle frequencies and hidden counts, and lower estimated worst-view success by 2.13 and 3.40 points. This reproduces, under matched supervision, the pattern of Section 5: reasoning-state supervision on its own amplifies view dependence.

The interaction persists at both scales. PR raises sECR by 7.06 points under RSM and 2.76 points under random states at 50M (9.37 and 3.21 at 100M), giving interactions of $+4.30$ and $+6.16$ with intervals well above zero. The interaction is equally clear in worst-view success ($+2.80$, $+3.60$) and mean success ($+1.64$, $+2.27$). The benefit of view exposure thus depends on the supervision it accompanies, and it is largest when paired with reasoning states.

What reasoning-state structure adds. After PR, reasoning states, density-matched random states, and weight-matched random states reach nearly the same normalized reliability at each scale (45.67–45.81 at 50M; 42.73–43.02 at 100M). In Sudoku, the stability benefit of the recipe therefore arises

from pairing view exposure with partial states at the density that RSM induces. Reasoning-state structure contributes additional solving ability: at 50M, RSM+PR exceeds both random-state controls in mean and worst-view success, with intervals above zero, and it attains the highest worst-view success at both scales. At 100M, these absolute differences are smaller and not resolved.

D EXPERIMENTAL DETAILS

This appendix gives the details needed to reproduce the experiments in Section 5. The main paper states the logic of each experiment. Here we state how the data, parser, sampling controls, and uncertainty estimates are implemented.

D.1 ANCHORED-COMPLETION GENERATION

We generate anchored-completion states only after the base Sudoku puzzle split. For each base puzzle, we choose anchors A, target cells U, and unresolved cells R. We keep the state only if the values of U are forced by A:

$$\{s_U : s \in \mathcal{S}(A)\} = \{s_U^\star\}. \tag{97}$$

The verification uses a Sudoku solver that enumerates completions until either two different target assignments are found or the target assignment is certified unique. The full grid need not be unique. Only the target assignment must be unique.

After a state is retained, we apply all eight elements of D_4. Because rotations and reflections preserve Sudoku row, column, and box constraints, every transformed state has the corresponding transformed target. During scoring, every prediction is mapped back to canonical coordinates before comparison.

D.2 PARSING AND SCORING

The model is asked to output the values of the target cells. A prediction is valid only if it can be parsed into exactly one digit for each requested target cell. If the output is malformed, incomplete, contains an invalid digit, or assigns two different values to the same target cell, it is mapped to the invalid symbol $\perp$ and scored as incorrect.

Invalid outputs contribute failures to every correctness summary. The group action fixes the invalid symbol: $g\perp = \perp$.

D.3 STOCHASTIC SAMPLING

For stochastic evaluation, we draw $K = 16$ samples per view unless otherwise stated. For each base state x and view g, the empirical success probability is

$$\widehat{p}_{g,K}(x) = \frac{1}{K}\sum_{k=1}^{K} \mathbf{1}[g^{-1}Y_{g,k} = y^\star(x)]. \tag{98}$$

We then compute

$$\widehat{C}^K_{\max}(x) = \max_{g\in G} \widehat{p}_{g,K}(x), \qquad \widehat{R}^K_{\min}(x) = \min_{g\in G} \widehat{p}_{g,K}(x). \tag{99}$$

The extrema are computed within each base state and then averaged over base states. We do not pool transformed views as independent examples.

D.4 CANONICAL PSEUDO-ORBIT CONTROL

The pseudo-orbit control measures the extrema gap produced by repeated sampling from the same canonical input. For each base state, we draw $|G|K$ samples from the canonical layout, randomly split them into $|G|$ groups of size K, and compute the same extrema used for the real orbit:

$$\widehat{C}^{K,\mathrm{pseudo}}_{\max}(x), \qquad \widehat{R}^{K,\mathrm{pseudo}}_{\min}(x). \tag{100}$$

The pseudo-orbit has no true geometric view shift. The resulting contrast compares cross-view behavior with this same-input reference. Away from the equal-distribution null, differences in success probabilities can also change the finite-sample extrema bias.

D.5 TOKEN VIEW-LEAKAGE DIAGNOSTIC

For each target cell i and view g, we estimate a token distribution $\widehat{p}_{g,i}$ from the K canonicalized samples. We add a small symmetric smoothing constant ϵ to every digit count before normalization. The reported token view-leakage proxy is

$$\widehat{L}_{\mathrm{tok}}(x) = \frac{1}{|U|} \sum_{i \in U} \frac{1}{|G|} \sum_{g \in G} D_{\mathrm{KL}}(\widehat{p}_{g,i} \| \widehat{\overline{p}}_i), \qquad \widehat{\overline{p}}_i = \frac{1}{|G|} \sum_{g \in G} \widehat{p}_{g,i}. \tag{101}$$

This diagnostic measures token-marginal agreement. Plug-in divergence can be positive under finite sampling even when the generating laws agree, and agreement of token marginals does not establish agreement of the full joint laws. The diagnostic therefore complements the task-success measurements.

D.6 BOOTSTRAP INTERVALS AND PAIRED COMPARISONS

The initial Sudoku intervals use 10,000 bootstrap resamples of base puzzles. Paired comparisons reuse each resampled puzzle index for both recipes and retain all views and repeated samples together. The matched study averages three fine-tuning seeds and reports puzzle-paired intervals; these intervals describe test-puzzle variation conditional on those runs. Maze tables give point estimates across initialization checkpoints. A collection of checkpoints from a pretraining trajectory is not a collection of independent training seeds.

D.7 PRETRAINING

Controlled MDLMs use DiT backbones with SUBS parameterization and sequence length 81 (one token per cell). Pretraining runs for 500K steps (5M, 10M, 50M) or 700K steps (100M) with global batch size 512 (256 for 100M), and we select the best checkpoint on validation. Sampling-based evaluation fixes the random seed to 0.

D.8 POST-TRAINING

The initial 50M recipes start from the same pretrained checkpoint and train for 9,000 steps with batch size 512 and learning rate 10^{-5}. RSM uses dense completion of unresolved cells; the SFT branches use their random-corruption pipeline. Canonical SFT uses 10k or 80k puzzles in their stored orientation with random masking of non-anchor cells. PR draws one uniformly random D_4 view per training instance; PR grouped places all eight views of a puzzle in the same batch and averages their losses. RSM retains the 4.6k of the 10k puzzles for which the oracle produces a trajectory and expands them into 41k intermediate states; RSM+PR applies one random view to each state.

The 8B experiment fine-tunes LLaDA-8B-Instruct with LoRA under the same recipe suite, training each branch for 8,000 optimizer steps and selecting the checkpoint by validation accuracy. It is evaluated on AC48 states, where the fine-tuned recipes do not saturate, with one decode per view.

D.9 PAIRED CODE AND MATHEMATICS EVALUATIONS

HumanEval-Infilling pairs are generated by applying one semantics-preserving edit to the surrounding context: identifier renaming, adjacent-statement swap, statement relocation, operand exchange with operator dualization, or inert-line insertion. A pair is admitted only if the transformed program passes the original test harness and matches the original's outputs. MATH-500 transformations (variable renaming, premise permutation, identity-operation insertion) are admitted only if they preserve the canonical answer. Predictions are aligned by the transformation before scoring, and ANY/ALL are computed over each original–transformed pair.

E REASONING STATES AND THE SUPERVISION OBJECTIVE

RSM selects conditional inputs rather than prescribing an output order. For a fixed problem with N initially hidden targets, unrestricted masking admits $2^N - 1$ nonterminal visibility patterns. A single monotone trajectory that reveals at least one new target per transition contains at most N nonterminal

states. This counts selected inputs; it is not a sample-complexity statement. Different states can supervise the same target under different visible information.

If a revealed assignment is already entailed by the visible facts, adding it need not shrink the set of logically feasible solutions. It makes an intermediate conclusion explicit. Dense RSM supervision then predicts every unresolved target from that input, rather than only the next frontier. Analogous task encodings can use verified intermediate facts or program-execution values; the training experiments here instantiate RSM on grids.

E.1 THE ORBIT-AVERAGED COMPLETION OBJECTIVE

For one state–target pair, let q_g be positive canonicalized predictions on a common finite label alphabet, and let

$$\widetilde{r}(a) = \prod_g q_g(a)^{1/m}, \quad Z = \sum_a \widetilde{r}(a), \quad r(a) = \widetilde{r}(a)/Z.$$

The logarithmic-opinion-pool identity is

$$\frac{1}{m}\sum_g -\log q_g(y) = -\log r(y) + \frac{1}{m}\sum_g D_{\mathrm{KL}}(r\|q_g). \tag{102}$$

Indeed, $\log r = \frac{1}{m}\sum_g \log q_g - \log Z$, so the average reverse KL equals $-\log Z$. RSM selects the state–target distribution over which this identity is averaged. Minimizing the sum does not require both summands to decrease on every update, and the identity applies to random-state supervision as well. The geometric consensus is an analytical quantity; inference does not ensemble over views.