Calibrate Once, Choose the Beam: A Pre-Deployment Compute-Allocation Rule for Same-LM Multimodal Search

Jiahui Qu [1] **Yifang Qin** [2] **Boyang Zheng** [1] **Ziyi Zhou** [3]

Abstract

Agentic planners must decide whether a fixed inference budget should buy more complete samples or a managed search frontier. We give a pre-deployment rule for this choice. The deployed LM, used as its own pruning verifier, is calibrated once to a precision p on locally true-improving partial states. Combining p with the mean horizon $\bar{L}$ yields a frontier-survival score $A_k \approx [1-(1-p)^k]^{\bar{L}}$, and the smallest beam k whose score clears a target success rate is the recommended width. The oracle labels used to measure p appear only on a small calibration split, never inside the test-time solver. On our 100-maze MazeBench (4×4 to 6×6, generator-controlled), the rule predicts the useful regime at $k = 3$ from a single calibration scalar $p = 0.816$. Same-LM guided search then solves 98% of 100 mazes in 14.4 seconds per maze on one A100, while SC-10 solves 9% in 17.7 seconds. A Qwen2.5-3B holdout, a Qwen2.5-VL-7B run on rendered visual mazes ($40/50$ vs $1/50$ for SC-10), and a same-LM two-ply Gomoku tournament ($100/100$ wins) all land on the safe side of the calibrated boundary, yielding a design map from $(p, \bar{L})$ to the minimum admissible beam across model size, modality, and task.

1. The Allocation Question

A deployed agent has a small number of ways to spend extra test-time compute. It can sample more complete trajectories and select among them (Wang et al., 2023; Snell et al., 2025; Brown et al., 2025): more samples buy more chances at predictable wall-clock cost. It can also keep a search frontier, ask the same LM to propose successor states, and ask the same LM to prune that frontier (Yao et al., 2023; Hao et al., 2023): when the LM prunes well, a managed beam buys depth-resilience at a fraction of the call budget. The second option is adaptive and hardware-aware, but also dangerous: a weak pruning verifier preserves many wrong states quickly, and detecting it needs a wide sweep over k.

The usual answers are unsatisfying for a short-latency deployment. A sweep over beam widths spends the budget one was trying to save. Training a process verifier can work (Lightman et al., 2024), but it changes the system, adds a training loop, and introduces a second inference component. We ask a narrower question: given the model already deployed, can a small calibration split tell us whether same-LM search is worth running and, if so, how wide the beam should be?

Our contribution is a deployment-time measurement protocol for deciding when an existing same-LM search stack should be used. Process reward models (Lightman et al., 2024) can replace the model's pruning role with a trained component. Tree-of-thought (Yao et al., 2023) and reasoning-with-world-model (Hao et al., 2023) typically set beam width by hand or by sweep. Self-consistency (Wang et al., 2023) spends compute on independent completions. We treat *evaluator calibration*, a measurement most papers report only as confidence quality, as a *compute-allocation* signal: one scalar, measured once, decides whether same-LM search is worth running and at what width. The rule lives in the deployment pipeline rather than in the model.

Our answer is one scalar. During calibration, an exact task oracle labels candidate successors as true-improving or not; in mazes, the label is whether BFS distance to the goal decreases. We then measure the precision p of the LM evaluator on the successors it endorses. The test-time solver never calls this oracle; the calibration number becomes a regime-placement rule, not a learned reward model.

This paper makes three claims. First, Eq. (1) turns $(p, \bar{L})$ into a predicted beam width without a sweep. Second, on matched A100 hardware the rule identifies a regime where guided search dominates repeated sampling in success and wall-clock. Third, the rule is auditable: stated proposer-

[1]London School of Economics and Political Science, London, UK [2]University College London, London, UK [3]Huaqiao University, Quanzhou, China. Correspondence to: Jiahui Qu <j.qu4@lse.ac.uk>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

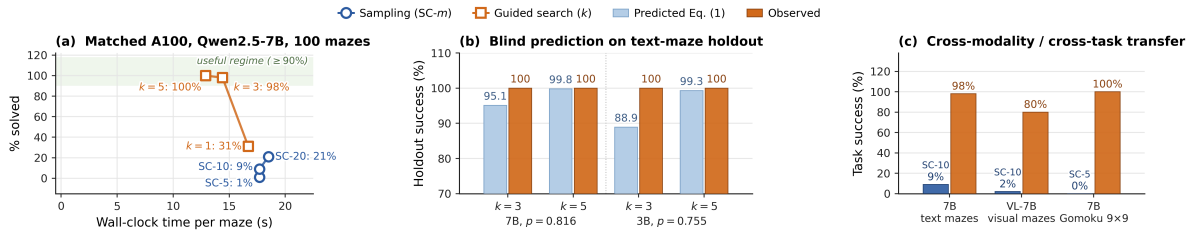

Figure 1. **One calibration scalar transfers across model size, modality, and task. (a)** On the same Qwen2.5-7B, single-A100 setup, same-LM guided search enters the $\geq 90\%$ useful regime at $k = 3$ while self-consistency remains below 22%. **(b)** The one-scalar prediction from a 24-maze calibration split blindly places the useful regime on a disjoint 50-maze text-maze holdout for both 7B and 3B. **(c)** The same recipe (same-LM proposer/pruner at $k = 3$, or 2-ply for Gomoku) transfers to rendered visual mazes (Qwen2.5-VL-7B, $40/50$ vs SC-10's $1/50$) and to 9×9 Gomoku ($100/100$ wins vs SC-5).

coverage and fallback assumptions, propagated calibration uncertainty, and a hard-horizon stress split where it under-certifies.

2. A One-Number Regime Test

The rule comes from a simple compounding model of guided search. At each layer the solver uses the LM in two roles. As a *proposer*, it samples up to $n = 3$ legal successors from each frontier state. As a *pruning verifier*, it scores the resulting partial states with a low-temperature evaluator prompt. The top k states survive to the next layer. Width k therefore buys a chance that at least one true-improving state remains in the beam.

Let p be the calibration precision of the pruning verifier on endorsed successors. If retained candidates behave like k independent chances to preserve a true-improving state, then a single layer survives with probability $1 - (1-p)^k$. Requiring this event across the $\bar{L}$ steps of a shortest solution gives

$$A_k \approx \left[1 - (1-p)^k\right]^{\bar{L}}. \tag{1}$$

For a target success rate τ, the corresponding precision threshold is

$$p_{\mathrm{req}}(k, \bar{L}, \tau) = 1 - \left(1 - \tau^{1/\bar{L}}\right)^{1/k}. \tag{2}$$

The practical test is to measure p and $\bar{L}$, then choose $k^\star = \min\{k : A_k \geq \tau\}$ subject to latency. In new deployments we recommend using the Wilson lower endpoint p_{low} in place of the point estimate when the calibration split is small. Table 1 summarizes the four steps.

Equation (1) is a design rule, not a theorem about arbitrary agents. It ignores proposer failures, fallback channels, and dependence between layers. We therefore log proposer recall r and the fallback floor, and we read the equation as a certification test in which unsafe over-certification is worse than conservative slack. Where error-conditioned transitions are available, a first-order Markov diagnostic (Appendix C) checks the direction of this dependence. The expected behavior is therefore observed accuracy at or above the prediction, and this holds in every run reported below.

Table 1. **Deployment protocol.** The oracle is used only to label the calibration rows. The solver in rows 3–4 is the same solver used at test time.

Step	Action
1	Run same-LM search at a non-saturated width, here $k = 2$, on a small labeled split.
2	Label successors by a local oracle and measure pruning precision p, proposer recall r, and mean horizon $\bar{L}$.
3	Compute A_k or p_{req} for feasible widths, preferably with p_{low} for a conservative read.
4	Deploy the smallest affordable k that clears the target, then confirm with one holdout width and log fallback calls.

3. Evidence: Prediction Before Sweeps

Calibration. We calibrate on 24 mazes, balanced across difficulty and disjoint from both evaluation sets. Running the stack at $k = 2$ yields 160 labeled successors. For Qwen2.5-7B (Qwen Team, 2024), the pruning precision is $p = 0.816$ with Wilson 95% interval $[0.749, 0.868]$ and single-call proposer recall $r = 0.724$. For Qwen2.5-3B, $p = 0.755$ and $r = 0.536$. The mean optimal horizon is about eight steps on calibration and holdout. Four evaluator prompt variants keep 7B precision in $[0.789, 0.878]$ (Appendix F), suggesting that the measured effect is not a one-prompt accident.

Matched hardware. Figure 1a compares repeated sampling and guided search on 100 text mazes on the same Qwen2.5-7B and one A100. Self-consistency improves slowly: SC-10 solves 9%, SC-20 solves 21%. Same-LM guided search reaches 31% at $k = 1$, 98% at $k = 3$, and 100% at $k = 5$, and $k = 3$ is also faster than SC-10 (14.4 vs 17.7 s/maze) because a calibrated frontier avoids recovery expansions.

Blind placement. We assess whether calibration places the useful-width boundary in advance, before any sweep. With $p = 0.816$ and $\bar{L} = 7.96$, Eq. (1) predicts $A_3 = 0.951$ at the point estimate and $A_3 = 0.88$ under the Wilson lower endpoint $p_{\mathrm{low}} = 0.749$, with $A_5 = 0.998$ on a disjoint 50-maze holdout; the solver solves $50/50$ mazes at both widths. Repeating the same protocol with Qwen2.5-3B gives $p = 0.755$, so Eq. (2) places the $\tau = 0.95$ boundary at

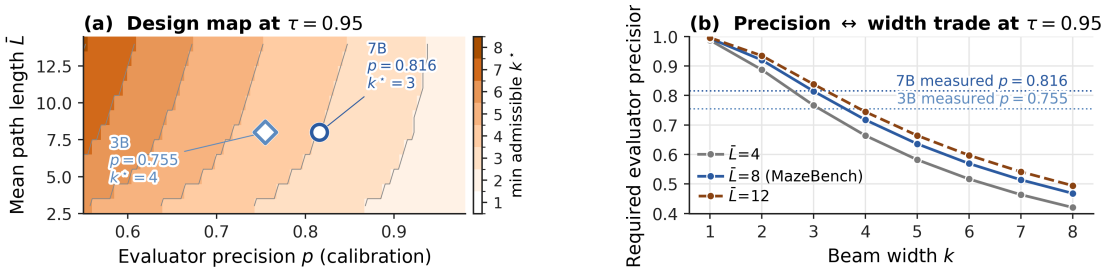

Figure 2. **Design map.** Measure pruning precision p, estimate horizon $\bar{L}$, and read the minimum admissible beam $k^\star$ for $\tau = 0.95$. Panel (b) gives the equivalent required-precision curves. The 7B point lands at $k^\star = 3$; the 3B point lands at $k^\star = 4$, with observed holdout accuracy on the safe side.

$k^\star = 4$. The observed $k = 3$ result already saturates, giving conservative slack relative to $A_3 = 0.889$ rather than a false positive; a targeted $k = 4$ rerun also solves $50/50$ mazes.

Stress (when the rule under-certifies). The 50-maze holdout saturates, so Appendix A adds a harder branch-complete split with $\bar{L} = 26.0$. There, $p = 0.667$ makes Eq. (1) refuse to certify $k \leq 5$ at $\tau = 0.95$ ($k^\star = 6$), while the observed curve rises from 4% at $k = 1$ to 100% at $k = 3$. The IID rule under-certifies, which is the safe failure mode for a deployment rule.

Multimodal and cross-task transfer. Figure 1c shows that the same recipe (same-LM proposer, same-LM pruning, $k = 3$) transfers without modification across modalities and tasks. On rendered visual mazes (512×512 PNG inputs), Qwen2.5-VL-7B (Bai et al., 2025) solves $40/50$ at $k = 3$ versus $1/50$ for SC-10. A branch-complete visual calibration on the same 24-maze protocol gives $p_{\mathrm{VL}} = 0.658$ (Wilson 95% CI $[0.628, 0.687]$) over 1032 labeled visual successors, yielding $A_3 = 0.722$ on the 50-maze horizon. The observed 0.80 visual holdout accuracy is therefore again on the conservative side of Eq. (1). On 9×9 Gomoku, a same-LM two-ply guided search wins $100/100$ games against same-LM Greedy and SC-5 opponents under matched call budgets.

Scope. The calibrated numeric law of Eq. (1) is reported on mazes where the local oracle is cheap, including rendered visual mazes via the underlying grid. The Gomoku probe tests recipe-level transfer rather than the calibration accuracy of A_k.

4. How to Use the Map, and When Not To

Figure 2 shows the design map a practitioner uses. After a small calibration to obtain $(p, p_{\mathrm{low}}, \bar{L})$, the smallest latency-feasible k follows directly from Eq. (2). If p_{low} clears the threshold, one confirmation sweep at that width is enough. If it clears nothing in the feasible range, the practitioner should change the evaluator prompt, model, state abstraction, or proposer.

Three checks should accompany the map. First, the proposer must cover true successors often enough: in our runs $1 - (1 - r)^3$ is about 0.98 for 7B and 0.90 for 3B, but much lower coverage would break the argument before pruning matters. Second, fallback or parser rescue channels must be measured. Our maze fallback floor is below 1%, but a domain with strong random legal moves could inflate observed success outside the rule. Third, horizon matters sharply. At $\tau = 0.95$ and $k = 3$, moving from $\bar{L} = 8$ to $\bar{L} = 12$ raises the required precision from about 0.814 to 0.838.

As a sanity probe, leave-one-maze-out logistic regression using cheap surface features (path length, distinct cells visited, revisits, legal next moves, bordering walls) and thresholded to match the LM's keep-rate reaches precision 0.767 (Wilson 95% CI $[0.677, 0.838]$) over 587 enumerated candidate states. The strict learned verifier is lower than the same-LM calibration in point estimate, though the intervals overlap on this small split. Adding Manhattan-distance features, which approximate the shortest-path oracle on open mazes, raises LR precision to 0.986 $[0.968, 0.994]$: the strict gap measures the value of geometric reasoning that cheap features cannot capture, rather than a capacity limit of the LR model class. In agentic domains without a cheap goal-direction signal, the strict baseline is the relevant comparison, and closing the gap with a learned verifier would require labels well beyond the protocol's calibration split.

A counterexample requires p_{low} clearing $p_{\mathrm{req}}(k, \bar{L}, \tau)$ with high proposer coverage, negligible fallback, and holdout success still below τ. The reported runs go the other way: when the rule certifies, holdout lands safely; when the horizon is harder, the rule declines to certify and the observed curve shows slack. The hard-horizon split (Appendix A) is the prototype: observed accuracy is far above the IID certificate, but the rule refuses to certify any tested width and thereby avoids recommending an under-performing beam. When p_{low} sits below every affordable threshold, the rule advises against deploying this LM as its own pruner. A richer deployment model could fit dependence explicitly; here the dependence and learned-verifier diagnostics audit the one-number rule rather than tune it. Within the scope established by these probes, one calibration scalar suffices

to decide whether same-LM frontier management is worth a confirmation run.

References

Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Ré, C., and Mirhoseini, A. Large language monkeys: Scaling inference compute with repeated sampling. In *International Conference on Learning Representations*, 2025.

Hao, S., Gu, Y., Ma, H., Hong, J. J., Wang, Z., Wang, D. Z., and Hu, Z. Reasoning with language model is planning with world model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 8154–8173, 2023.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. In *International Conference on Learning Representations*, 2025.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023.

A. Hard-Horizon Stress Split

The stress split uses 50 disjoint winding-corridor mazes with dead-end distractor branches and mean optimal horizon $\bar{L} = 26.0$. A separate 12-maze calibration split gives $p = 2/3$. To isolate pruning from proposal coverage, this run is branch-complete: all legal code successors are added to the candidate set before the same LM evaluator prunes to beam width k. The test therefore asks whether the calibrated evaluator can preserve the frontier, not whether the proposer finds every legal action. Figure 3 plots the IID certificate against observed accuracy, and Table 2 reports per-width predicted and observed values along with wall-clock and call counts.

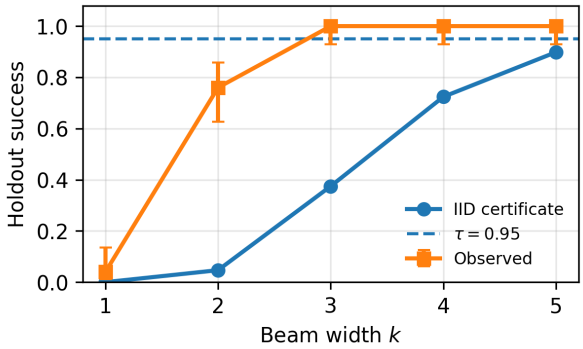

Figure 3. **Hard-horizon stress test.** The IID certificate is not a fitted curve. With $p = 0.667$ and $\bar{L} = 26.0$, it gives $k^\star = 6$ for $\tau = 0.95$ and does not certify any tested width, while observed success reaches $50/50$ at $k = 3$.

k	A_k	Obs.	Time (s)	Calls
1	0.00003	4%	20.6	14.0
2	0.047	76%	41.2	26.0
3	0.375	100%	41.7	26.0
4	0.724	100%	41.2	26.0
5	0.898	100%	40.6	26.0

Table 2. Hard split prediction table. Time and calls are per-maze averages on the 50-maze holdout.

B. Reproducibility Details

The text-maze benchmark contains 100 guaranteed-solvable mazes: 34 easy 4×4, 33 medium 5×5, and 33 hard 6×6, with mean optimal path length $\bar{L} = 7.98$. The disjoint 50-maze holdout has $\bar{L} = 7.96$. The calibration split contains 24 mazes and 160 labeled successors. A successor is labeled true positive if exact BFS distance to the goal decreases. The oracle is used only for this split.

For guided search, the proposer samples up to $n = 3$ legal successors at temperature 1.0 with a short output cap; the evaluator scores candidate states at temperature 0.1. The same model checkpoint and prompt stack are used for both roles. Retained states are the top k evaluator-ranked states at each layer. Self-consistency samples complete trajectories and accepts a successful maze path if the full path is legal and reaches the goal. All wall-clock numbers in the main text are measured on the same A100 configuration; they are intended as system-level timings, not FLOP-normalized estimates.

C. Calibration Uncertainty

For Qwen2.5-7B, $p = 0.816$ is close to the IID $k = 3$ boundary at $\tau = 0.95$, $\bar{L} \approx 8$, where $p_{\text{req}} \approx 0.814$. Its Wilson interval therefore spans both sides of the strict threshold. This is why the paper recommends using the Wilson lower endpoint in new deployments. In the reported run, the observed holdout result sits on the safe side of the point prediction. For Qwen2.5-3B, $p = 0.755$ clears the $k = 4$ threshold but not the strict $k = 3$ threshold; the observed $k = 3$ saturation is a visible instance of conservative slack. The hard-horizon split shows the same phenomenon more strongly: Eq. (1) under-certifies $k = 3$ because retained-state errors and successes are not IID across layers.

A first-order Markov diagnostic gives the same direction when there are enough error-conditioned transitions to estimate it. Let $q = (1-p)^k$ be the per-step probability that no retained state is true-improving. In the visual-maze calibration, $p_{\text{VL}} = 0.658$ gives $q = (1-p_{\text{VL}})^3 = 0.040$, while the empirical adjacent-layer diagnostic gives $c = \Pr(\text{err}_{t+1} \mid \text{err}_t) = 0.80$ over five error-conditioned transitions. The sample is intentionally treated as a sign check, not a fitted correction. In the text calibration rerun, layer errors are too rare to estimate c reliably, which is itself consistent with the saturated text holdout. The exact size of this correction is task-dependent, so it is not used to tune the prediction; it only motivates reading Eq. (1) as a conservative rule under same-LM pruning.

D. Transfer Probe Protocols

For visual mazes, the same 50 holdout mazes are rendered as fixed-size PNG images with cell edge 48 px: walls black, passable cells white, start green, goal red, current position blue. Qwen2.5-VL-7B receives the image plus a short path-history text. The proposer emits one of {U,D,L,R,DONE}; the evaluator rates the resulting visual state using the same ordinal scale as the text evaluator. Guided search uses $k = 3$, $n = 3$. The run solves 40/50 mazes; SC-10 solves 1/50.

To test the numeric law in the visual setting, we also run a branch-complete calibration on 24 rendered mazes: every legal one-step successor is rendered and scored by the VL evaluator, while the BFS oracle labels whether the underlying grid distance to the goal decreases. This yields $p_{\text{VL}} = 0.658$ over 1032 visual successors, with $A_3 = 0.722$ at $\bar{L} = 7.96$.

The observed 50-maze accuracy, $40/50 = 0.80$, is above this prediction. The dominant failure mode on the remaining 10 mazes is *coordinate drift*: the VL evaluator sometimes rates a state from an implicit position estimate rather than the on-image marker, and the rating becomes unstable when the two disagree.

For Gomoku, the model receives the current 9×9 board and candidate moves. Greedy chooses the top single move. SC-5 samples five complete move proposals and keeps the most frequent legal move. Guided search evaluates same-LM two-ply continuations under the same call budget used in the baseline comparison. The guided policy wins 100/100 games against the same-LM Greedy and SC-5 opponents in the reported run.

E. Robustness Diagnostics

Table 3 consolidates the diagnostics referenced from the main body. The 30-maze three-seed rerun preserves the ordering SC-10 $<$ guided k=1 $<$ guided k=3 across all seeds. The four non-degenerate evaluator prompt variants (binary, scored, stripped, few-shot) keep 7B precision in $[0.789, 0.878]$, indicating the calibration is not pinned to a single prompt string. The fallback floor is the rate at which a code-level rescue channel proposes legal moves when the LM emits no parseable successor; below 1% on text mazes, it does not contaminate the regime-placement claim at $k \geq 3$.

F. Prompt Templates

The same Qwen2.5-7B-Instruct (or Qwen2.5-3B-Instruct) checkpoint serves both proposer and evaluator roles. Braces $\{\cdot\}$ mark runtime substitutions. We list the maze prompts for the baseline evaluator variant E0 used in all main-text numbers; three additional variants (E1 binary, E2 stripped, E3 few-shot) keep p within $[0.789, 0.878]$ and are used only for the prompt-stability diagnostic in Appendix E.

Maze proposer (single-step search-guidance heuristic, temp 1.0, `max_tokens`$= 10$).

```
You are navigating a maze.

Maze:
{maze}

Legend: S=start, E=end, .=path, #=wall
Current position: row {row}, col {col}
Path so far: {path}

Choose ONE direction to move toward E:
- U (up, row-1)
- D (down, row+1)
- L (left, col-1)
- R (right, col+1)
Avoid walls and visited positions.
If at E, say DONE.

Your move (U/D/L/R or DONE):
```

Table 3. Calibration and robustness diagnostics referenced in §3 and §4.

Diagnostic	Value
7B calibration	24 mazes, 160 labeled successors; $p = 0.816$, recall 0.724
3B calibration	Same protocol; $p_{3B} = 0.755$, recall 0.536
3B $k = 4$ confirmation	Targeted SUBSET_50 rerun solves $50/50$ mazes (Wilson 95% CI $[0.929, 1.000]$)
VL calibration	24 rendered mazes, 1032 visual successors; $p_{\mathrm{VL}} = 0.658$, $A_3 = 0.722$, observed $40/50$
Wilson 95% CI on p_{7B}	$[0.749, 0.868]$ (binomial, $n = 160$)
Learned-verifier probe	Strict LR precision 0.767; adding Manhattan features raises it to 0.986
Prompt stability (7B)	4 non-degenerate evaluator variants: $p \in [0.789, 0.878]$
Seed stability (30-maze, 3 seeds)	SC-10 $5.6 \pm 1.9\%$, $k = 1$ $24.4 \pm 6.9\%$, $k = 3$ $100.0 \pm 0.0\%$
Difficulty split (100-maze)	SC-10 solves $5/34$ easy, $2/33$ medium, $2/33$ hard; guided $k = 5$ solves $100/100$
Fallback floor (text maze)	$<1\%$ of expansions invoke the code-level rescue channel

Maze pruning verifier (E0 baseline, temp 0.1, `max_tokens`$= 50$).

```
Evaluate this maze-solving state.

Maze:
{maze}

Position: row {row}, col {col}
Path taken: {path}
Goal: reach E

Rate this position:
- "sure" if making progress toward E
- "likely" if reasonable but uncertain
- "impossible" if stuck or wrong direction

Rating:
```

Ratings are mapped `sure`$\rightarrow$ 10, `likely`$\rightarrow$ 1, `impossible`$\rightarrow$ 0.1. *Endorsed* means the rating is `sure` or `likely`; pruning precision p is the fraction of endorsed successors whose oracle BFS distance to the goal strictly decreases.

Gomoku move (proposer, temp 1.0, `max_tokens`$= 16$). The board is rendered as a 9×9 ASCII grid (`X`/`O`/`.`); the proposer emits a single coordinate `(row,col)`. The Gomoku evaluator uses the same `sure`/`likely`/`impossible` scale on a two-ply continuation board, with `max_tokens`$= 64$.

Visual maze (Qwen2.5-VL). The proposer and evaluator receive the rendered PNG (§D) plus a short text overlay listing the path so far. Output formats and ordinal scale match the text-maze prompts above.

LLM/Agent Usage Disclosure

The authors used LLM/agent assistance for writing and presentation support. This included English-language editing, grammar correction, sentence-level polishing, LaTeX formatting assistance, and checklist-style critique of clarity.

LLM/agent tools were also used to suggest cosmetic improvements to figures, including layout, labels, captions, and readability, and to help organize camera-ready rerun notebooks and utility code. All experiments were executed and verified by the authors, and all numerical results come from author-run code and author-supplied logs. The assistant did not generate independent numerical results.

All experimental design, implementation, analysis, hard-split results, scientific claims, references, and final decisions were completed and verified by the authors. The authors reviewed the final manuscript and retain full responsibility for the submission.