Evaluating Spatial Reasoning in Language Models

Aarush Gupta
Celeritas Research
admin@celeritasresearch.org

Abstract

Existing reasoning benchmarks for language models (LMs) frequently fail to adequately assess spatial reasoning. In this work, we study spatial and topological reasoning by introducing a text-first benchmark built from *Slitherlink* and *Nurikabe*, two canonical constraint-satisfaction and grid-based connectivity puzzles. We generate this benchmark with a solver-aided framework that encodes constraints into Boolean form and samples solutions from these constraints with near-uniformity over a specified projection, yielding instance distributions that are diverse and minimally biased by handcrafted heuristics. We represent puzzle instances in a custom coordinate-based domain-specific language (DSL) and evaluate them with a rigorous validation engine. Baseline experiments show substantially higher accuracy on *Nurikabe* than on *Slitherlink*, with single-cycle loop topology emerging as the principal bottleneck; however, the results do not indicate any distinctive advantage in either puzzle family, showing that spatial reasoning remains an open challenge.

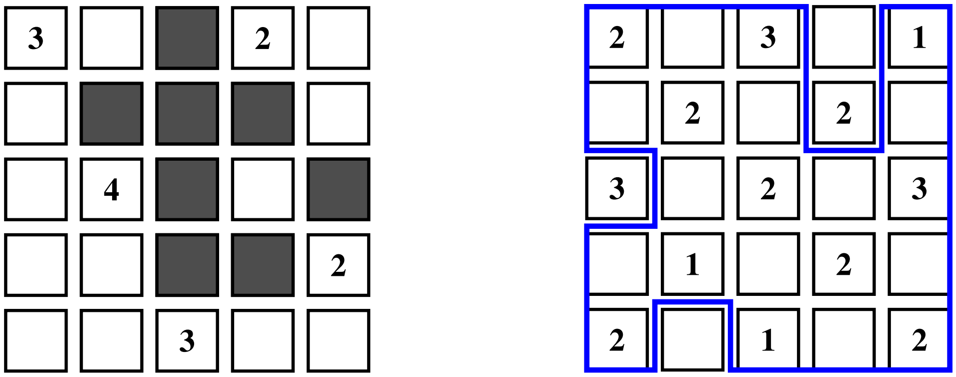
1 Introduction

Reasoning benchmarks for language models have primarily focused on mathematical word problems, code synthesis, and symbolic puzzles that emphasize local constraint satisfaction. Comparatively less attention has been paid to spatial and topological reasoning, particularly when such reasoning must be performed from text alone. While frontier language models have demonstrated impressive capabilities on many tasks, they often struggle with problems requiring long chains of deductive reasoning or global consistency, even with search-based decoding [16].

Existing puzzle-style evaluations have made progress in related areas. Sudoku benchmarks test logical deduction and local consistency [10], while logic-grid tasks probe entity-relation reasoning [12]. Other work has explored search-based prompting for crosswords and arithmetic [16], and neuro-symbolic approaches that translate natural-language rules into executable solver programs [7]. However, these benchmarks do not directly evaluate the ability to reason about explicit global connectivity constraints, such as ensuring that a configuration forms a single connected component or a single closed loop. It remains an open question whether current language models can build and maintain internal topological models from symbolic descriptions, enforce global structural properties alongside local rules, and scale their reasoning as problem size increases.

To address this gap, we introduce a benchmark centered on two canonical connectivity puzzles: *Slitherlink* and *Nurikabe*. These puzzles are particularly well-suited as reasoning testbeds because they require models to simultaneously satisfy local arithmetic constraints and global topological invariants, a combination that resists purely heuristic approaches. In *Slitherlink*, the solver must identify a set of grid edges forming a single simple cycle, such that each numbered cell has exactly the specified count of incident loop edges. In *Nurikabe*, the task is to partition a cell grid into black and white regions, where black cells form one connected sea (with no 2×2 all-black blocks), and each white island is a connected component of a specified size anchored at a given seed. Both puzzle families are computationally hard (with NP-complete variants [13, 6, 4]) and inherently test

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.



(a) Nurikabe puzzle example

(b) Slitherlink puzzle example

Figure 1: (a) Numbered cells indicate island sizes. The black cells represent the "sea" that connects all the way through the grid, while white cells form "islands." The solution shows how the sea separates different numbered islands. (b) Numbers indicate how many edges of that cell are part of the loop. The blue line shows an example solution—a single closed loop that satisfies all the number constraints.

reasoning about global topological structure, making them ideal stress tests for capabilities beyond local pattern matching. Moreover, their text-first representation forces models to construct and manipulate internal spatial models without relying on visual perception, isolating the reasoning component from recognition.

Our contributions are threefold. First, we provide a text-first benchmark with standardized input/output via a domain-specific language and deterministic validators that enforce all constraints exactly. Second, we develop a distribution-aware generation pipeline that encodes puzzles as Boolean constraint systems and uses projection-based sampling methods to produce diverse instances without the biases of heuristic generators [11, 8, 1]. Third, we report evaluation results on frontier models with constraint-level diagnostics, complementing prior benchmarks that target non-topological reasoning or rely on perceptual input [12, 10, 16, 3, 2].

2 Related Work

Abstraction and generalization benchmarks. The Abstraction and Reasoning Corpus (ARC-AGI) evaluates abstraction with minimal priors using visual grid tasks [3, 2]. While ARC probes general intelligence, its tasks are vision-centric and do not directly operationalize topological constraints like a single closed cycle or a single connected component, which are central to our evaluation.

Puzzle benchmarks and reasoning strategies. Recent benchmarks use constraint-based puzzles to evaluate reasoning. Sudoku evaluations provide controlled difficulty and exact scoring for local consistency [10], while logic-grid puzzles test entity-relation inference without spatial structure [12]. Various prompting strategies (Tree-of-Thought [16], self-consistency [14], ReAct [17]) and neuro-symbolic approaches [7] have been proposed. While these establish valuable practices (standardized representations, deterministic validation), they do not target global topological constraints in text-first settings. Most rely on heuristic generators that may introduce systematic biases rather than distribution-aware sampling.

Connectivity puzzles and formal encodings. The complexity literature documents NP-complete puzzle families where connectivity is central [4]. Practical approaches use SAT, ILP, and flow encodings to enforce single-loop and single-component constraints [13]. Our benchmark builds on this by foregrounding explicit global connectivity with text-first representations, exact validators, and distribution-aware generation.

3 Benchmark

We construct our benchmark on two spatial puzzles, *Slitherlink* and *Nurikabe*, to evaluate language models on topological reasoning. Each puzzle family is generated in size-tiered categories (small, medium, and large) to enable systematic difficulty scaling. All instances are text-first: problems and solutions are represented symbolically without images, which isolates reasoning from perceptual confounds. Our design serves three core objectives. First, by eliminating visual input, we ensure that performance reflects genuine topological reasoning rather than pattern recognition or reliance on pretrained visual priors [3]. Second, we ground each puzzle in formal Boolean constraint systems (detailed in Appendix B) that encode both local rules and global connectivity requirements, supporting exact verification and uniqueness testing consistent with established complexity analyses [13, 4]. Third, we adopt a projection-aware sampling methodology (Section 3.1) to generate diverse instances with minimal distributional bias [11, 8, 1].

Each instance is serialized in a custom coordinate-based DSL. The DSL grammar specifies explicit grid dimensions, normalized listings of clues or island seeds in canonical order, and delimited solution blocks for evaluation. This representation supports robust assessment: the same logical puzzle can be presented with different surface orderings or symmetries, allowing us to probe sensitivity to representation choices while maintaining semantic equivalence.

3.1 Projection-Aware Sampling

Generating diverse puzzle instances requires care: naively sampling solutions from a Boolean encoding can introduce hidden biases. We encode puzzle constraints as Boolean formulas in Conjunctive Normal Form (CNF), where a CNF formula is a conjunction of clauses, each a disjunction of literals (variables or their negations). For example, the formula $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$ is a CNF with two clauses. For puzzle generation, we partition variables into two sets: task variables T that directly represent puzzle elements visible to the model (edge indicators in *Slitherlink*, cell colors in *Nurikabe*), and auxiliary variables A introduced by the encoding to enforce complex constraints efficiently (such as flow variables for connectivity, as detailed in Appendix B). Let $X = T \cup A$ denote all variables and F the CNF formula over X, with satisfying assignments $S(F) \subseteq \{0,1\}^X$. A projection $\pi : \{0,1\}^X \to \{0,1\}^T$ maps each full assignment to its task-variable portion.

The central challenge is projection bias. If we sample uniformly over $S(F)$, the distribution over projected solutions $\pi(S(F))$ can be highly skewed: some puzzle configurations $y \in \{0,1\}^T$ may correspond to many auxiliary completions, making them overrepresented, while others with fewer completions become rare. This bias can favor certain puzzle topologies (configurations requiring simpler flow patterns) and undermine benchmark diversity. Our goal is instead to sample approximately uniformly over distinct puzzle solutions $\pi(S(F))$, ensuring each configuration has probability $(1 \pm \varepsilon)/|\pi(S(F))|$. We achieve this via a three-step pipeline [11, 8, 1]: first, identify an independent support $I \subseteq X$ (variables that uniquely determine all others; typically $I \approx T$ in our encodings); second, add m sparse random XOR constraints $h_j(I) = \alpha_j$ to partition $\pi(S(F))$ into near-equiprobable cells; third, sample from one cell via UniGen-style enumeration, yielding (ε, δ) uniformity guarantees. A worked example illustrating projection bias and how XOR hashing corrects it is provided in Appendix A.

After sampling, we apply deterministic post-processing: uniqueness filtering (re-solving with the known solution blocked to reject instances with multiple solutions) and dihedral canonicalization (selecting a canonical orientation under the 8-element symmetry group). These operations act only on T. While uniformity is guaranteed over the pre-filter sample, post-processing can introduce bias if solution multiplicity or symmetry orbit sizes correlate with puzzle features. To mitigate this, we implement symmetry-breaking constraints during SAT solving where feasible, and empirically monitor distributional properties (topology statistics, orbit-size distributions) to detect systematic skew. The resulting benchmark achieves distributional control largely independent of encoding choices, coverage of diverse topologies, and reproducible comparisons across sizes and formats, though we acknowledge that perfect uniformity after canonicalization would require orbit-weighted sampling.

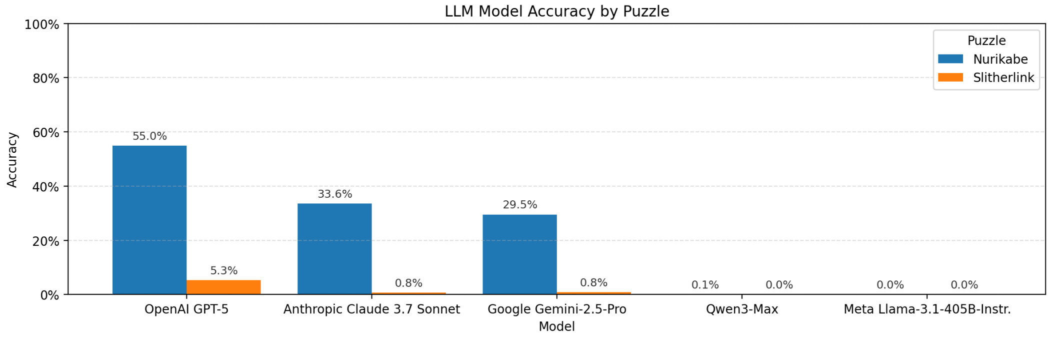


Figure 2: Evaluation results of frontier models on *Nurikabe* ($N = 7958$) and *Slitherlink* ($N = 6579$). Note the substantial accuracy disparity between puzzle types and the consistent ordering across models.

4 Results

We evaluate several frontier language models on both puzzle families and observe a striking pattern: all models achieve substantially higher accuracy on *Nurikabe* than on *Slitherlink* (Figure 2). This ordering is stable across the model spectrum. On *Slitherlink*, even the strongest system achieves only single-digit accuracy, while most models score near zero. This performance gap invites a principled explanation grounded in the structural differences between the two puzzle types.

From a combinatorial perspective, *Slitherlink* poses a more constrained problem. The puzzle requires a globally consistent edge set forming exactly one simple cycle; a single incorrect edge propagates contradictions via degree and parity constraints. The space of locally valid configurations (degree 0 or 2 at each vertex) is exponential (2^{ab} in an $a \times b$ grid), but only a tiny fraction forms a single connected loop. One error cascades through the constraint network, making recovery difficult under left-to-right generation. In contrast, *Nurikabe* enforces structure through sea connectivity and island sizes. While the 2×2 exclusion rule permits many fragmented configurations, *Nurikabe* admits incremental region-growth strategies where violations (overgrown islands, emerging 2×2 blocks) can often be repaired locally without global reconfiguration. The degree-$\{0, 2\}$ regime in *Slitherlink* tightly couples distant choices with weak gradient signals, whereas *Nurikabe* supports localized reasoning. Output representation also matters: emitting a correct sparse edge list demands tight global planning, while a binary cell mask aligns naturally with token generation.

Model responses reveal instructive failure modes. Meta Llama 3.1 405B [5] frequently rejects valid puzzles as unsolvable, while Qwen 3 Max [15] produces outputs consistently but with high error rates. GPT-5 [9] invests substantially more reasoning tokens (up to 45,000 versus hundreds for others), suggesting overconfidence in weaker models or fundamental training differences. Table 1 provides a constraint-level error breakdown. Strikingly, while global connectivity defines theoretical hardness, dominant failures occur at simpler levels: degree violations (73% for *Slitherlink*) and island size mismatches (66% for *Nurikabe*). Current models struggle with maintaining consistency across many local constraints during generation, not primarily with abstract topological reasoning.

To probe scaling behavior, we analyze accuracy across size buckets: small (up to 6×6 for *Nurikabe*, 10×10 for *Slitherlink*), medium (up to 8×8 and 12×12 respectively), and large (beyond these thresholds). Figure 3 shows accuracy by size for both puzzles. Accuracy on *Nurikabe* shows essentially no size dependence, except for a dip at the 8×8 size, whereas *Slitherlink* exhibits a strong negative size effect. This suggests that *Nurikabe*'s challenge is not primarily computational but structural, remaining consistently difficult regardless of grid dimensions, while *Slitherlink*'s single-cycle constraint becomes exponentially harder as the search space grows. The flat scaling profile for *Nurikabe* further supports the hypothesis that local constraint satisfaction, rather than raw problem size, drives the difficulty gap between puzzle families.

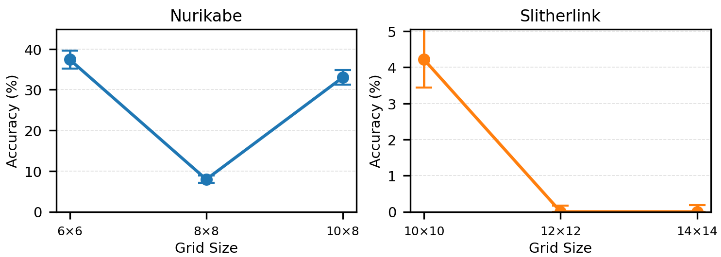


Figure 3: Accuracy versus grid size for *Nurikabe* and *Slitherlink*. Error bars show 95% Wilson confidence intervals. *Nurikabe* exhibits size-invariant difficulty, except for a dip at 8×8 puzzles, while *Slitherlink* accuracy degrades sharply with increasing grid dimensions.

Puzzle	Constraint	Error Rate (95% CI)
Nurikabe	Island size mismatch	65.7% (63.9–67.4)
	Incorrect row length	16.1% (14.8–17.6)
	DSL: parse error	5.2% (4.4–6.0)
	DSL: missing `<solution>` tags	4.8% (4.1–5.6)
	Seed cell not white	3.5% (2.9–4.3)
	Incorrect row count	3.2% (2.6–3.9)
	Sea disconnected	0.8% (0.6–1.2)
	Unexpected white cell	0.3% (0.1–0.6)
	Sea contains a 2×2 block	0.2% (0.1–0.5)
	Overlapping islands	0.2% (0.1–0.4)
Slitherlink	Dot degree violation	73.6% (71.0–76.0)
	Edge out of bounds	6.9% (5.5–8.5)
	No edges provided	6.7% (5.4–8.3)
	Clue mismatch	5.0% (3.9–6.4)
	DSL: parse error	2.7% (2.0–3.8)
	DSL: missing `<solution>` tags	2.7% (1.9–3.7)
	Duplicate edge	2.1% (1.5–3.1)
	Loop is not a single cycle	0.3% (0.1–0.9)

Table 1: Constraint-wise diagnostics for *Nurikabe* and *Slitherlink*. Error rates are shown among incorrect outputs with 95% Wilson confidence intervals.

5 Conclusion

We presented a text-first benchmark for spatial and topological reasoning built around *Slitherlink* and *Nurikabe*. Our methodology combines formal Boolean encodings, projection-aware sampling for diverse unbiased generation, a standardized DSL, and deterministic validators.

Empirical results show substantially higher accuracy on *Nurikabe* than *Slitherlink*, aligning with structural differences: *Slitherlink* requires globally consistent cycles where errors cascade through tight degree constraints, while *Nurikabe* admits localized reasoning with incremental repair. Constraint diagnostics show models predominantly fail on local combinatorial constraints rather than abstract topological properties, suggesting challenges in maintaining consistency across coupled decisions.

Despite this disparity, overall accuracy remains modest on both families, indicating spatial reasoning remains an open challenge. This benchmark provides a controlled testbed for future work on inference-time search, neuro-symbolic integration, or training for long-range constraint propagation.

References

[1] UniGen: Almost-uniform sampler. GitHub repository, 2024. Includes references to ApproxMC4 and UniGen3.

[2] ARC Prize: Benchmark and competition hub. `https://arcprize.org/`, 2025.

[3] F. Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

[4] E. Demaine et al. A survey of np-complete puzzles. Technical Survey, 2001.

[5] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Wyatt, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Guzmán, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Thattai, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. Kloumann, I. Misra, I. Evtimov, J. Zhang, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Prasad, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, K. El-Arini, K. Iyer, K. Malik, K. Chiu, K. Bhalla, K. Lakhotia, L. Rantala-Yeary, L. van der Maaten, L. Chen, L. Tan, L. Jenkins, L. Martin, L. Madaan, L. Malo, L. Blecher, L. Landzaat, L. de Oliveira, M. Muzzi, M. Pasupuleti, M. Singh, M. Paluri, M. Kardas, M. Tsimpoukelli, M. Oldham, M. Rita, M. Pavlova, M. Kambadur, M. Lewis, M. Si, M. K. Singh, M. Hassan, N. Goyal, N. Torabi, N. Bashlykov, N. Bogoychev, N. Chatterji, N. Zhang, O. Duchenne, O. Çelebi, P. Alrassy, P. Zhang, P. Li, P. Vasic, P. Weng, P. Bhargava, P. Dubal, P. Krishnan, P. S. Koura, P. Xu, Q. He, Q. Dong, R. Srinivasan, R. Ganapathy, R. Calderer, R. S. Cabral, R. Stojnic, R. Raileanu, R. Maheswari, R. Girdhar, R. Patel, R. Sauvestre, R. Polidoro, R. Sumbaly, R. Taylor, R. Silva, R. Hou, R. Wang, S. Hosseini, S. Chennabasappa, S. Singh, S. Bell, S. S. Kim, S. Edunov, S. Nie, S. Narang, S. Raparthy, S. Shen, S. Wan, S. Bhosale, S. Zhang, S. Vandenhende, S. Batra, S. Whitman, S. Sootla, S. Collot, S. Gururangan, S. Borodinsky, T. Herman, T. Fowler, T. Sheasha, T. Georgiou, T. Scialom, T. Speckbacher, T. Mihaylov, T. Xiao, U. Karn, V. Goswami, V. Gupta, V. Ramanathan, V. Kerkez, V. Gonguet, V. Do, V. Vogeti, V. Albiero, V. Petrovic, W. Chu, W. Xiong, W. Fu, W. Meers, X. Martinet, X. Wang, X. Wang, X. E. Tan, X. Xia, X. Xie, X. Jia, X. Wang, Y. Goldschlag, Y. Gaur, Y. Babaei, Y. Wen, Y. Song, Y. Zhang, Y. Li, Y. Mao, Z. D. Coudert, Z. Yan, Z. Chen, Z. Papakipos, A. Singh, A. Srivastava, A. Jain, A. Kelsey, A. Shajnfeld, A. Gangidi, A. Victoria, A. Goldstand, A. Menon, A. Sharma, A. Boesenberg, A. Baevski, A. Feinstein, A. Kallet, A. Sangani, A. Teo, A. Yunus, A. Lupu, A. Alvarado, A. Caples, A. Gu, A. Ho, A. Poulton, A. Ryan, A. Ramchandani, A. Dong, A. Franco, A. Goyal, A. Saraf, A. Chowdhury, A. Gabriel, A. Bharambe, A. Eisenman, A. Yazdan, B. James, B. Maurer, B. Leonhardi, B. Huang, B. Loyd, B. D. Paola, B. Paranjape, B. Liu, B. Wu, B. Ni, B. Hancock, B. Wasti, B. Spence, B. Stojkovic, B. Gamido, B. Montalvo, C. Parker, C. Burton, C. Mejia, C. Liu, C. Wang, C. Kim, C. Zhou, C. Hu, C.-H. Chu, C. Cai, C. Tindal, C. Feichtenhofer, C. Gao, D. Civin, D. Beaty, D. Kreymer, D. Li, D. Adkins, D. Xu, D. Testuggine, D. David, D. Parikh, D. Liskovich, D. Foss, D. Wang, D. Le, D. Holland, E. Dowling, E. Jamil, E. Montgomery, E. Presani, E. Hahn, E. Wood, E.-T. Le, E. Brinkman, E. Arcaute, E. Dunbar, E. Smothers, F. Sun, F. Kreuk, F. Tian, F. Kokkinos, F. Ozgenel, F. Caggioni, F. Kanayet, F. Seide, G. M. Florez, G. Schwarz, G. Badeer, G. Swee, G. Halpern, G. Herman, G. Sizov, Guangyi, Zhang, G. Lakshminarayanan, H. Inan, H. Shojanazeri, H. Zou, H. Wang, H. Zha, H. Habeeb, H. Rudolph, H. Suk, H. Aspegren, H. Goldman, H. Zhan, I. Damlaj, I. Molybog, I. Tufanov, I. Leontiadis, I.-E. Veliche, I. Gat, J. Weissman, J. Geboski, J. Kohli, J. Lam, J. Asher, J.-B. Gaya, J. Marcus, J. Tang, J. Chan, J. Zhen, J. Reizenstein, J. Teboul, J. Zhong, J. Jin, J. Yang, J. Cummings, J. Carvill, J. Shepard, J. McPhie, J. Torres, J. Ginsburg, J. Wang, K. Wu, K. H. U, K. Saxena, K. Khandelwal, K. Zand, K. Matosich, K. Veeraraghavan, K. Michelena, K. Li, K. Jagadeesh, K. Huang, K. Chawla, K. Huang, L. Chen, L. Garg, L. A, L. Silva, L. Bell, L. Zhang, L. Guo, L. Yu, L. Moshkovich, L. Wehrstedt, M. Khabsa, M. Avalani, M. Bhatt, M. Mankus, M. Hasson, M. Lennie, M. Reso, M. Groshev,

M. Naumov, M. Lathi, M. Keneally, M. Liu, M. L. Seltzer, M. Valko, M. Restrepo, M. Patel, M. Vyatskov, M. Samvelyan, M. Clark, M. Macey, M. Wang, M. J. Hermoso, M. Metanat, M. Rastegari, M. Bansal, N. Santhanam, N. Parks, N. White, N. Bawa, N. Singhal, N. Egebo, N. Usunier, N. Mehta, N. P. Laptev, N. Dong, N. Cheng, O. Chernoguz, O. Hart, O. Salpekar, O. Kalinli, P. Kent, P. Parekh, P. Saab, P. Balaji, P. Rittner, P. Bontrager, P. Roux, P. Dollar, P. Zvyagina, P. Ratanchandani, P. Yuvraj, Q. Liang, R. Alao, R. Rodriguez, R. Ayub, R. Murthy, R. Nayani, R. Mitra, R. Parthasarathy, R. Li, R. Hogan, R. Battey, R. Wang, R. Howes, R. Rinott, S. Mehta, S. Siby, S. J. Bondu, S. Datta, S. Chugh, S. Hunt, S. Dhillon, S. Sidorov, S. Pan, S. Mahajan, S. Verma, S. Yamamoto, S. Ramaswamy, S. Lindsay, S. Lindsay, S. Feng, S. Lin, S. C. Zha, S. Patil, S. Shankar, S. Zhang, S. Zhang, S. Wang, S. Agarwal, S. Sajuyigbe, S. Chintala, S. Max, S. Chen, S. Kehoe, S. Satterfield, S. Govindaprasad, S. Gupta, S. Deng, S. Cho, S. Virk, S. Subramanian, S. Choudhury, S. Goldman, T. Remez, T. Glaser, T. Best, T. Koehler, T. Robinson, T. Li, T. Zhang, T. Matthews, T. Chou, T. Shaked, V. Vontimitta, V. Ajayi, V. Montanez, V. Mohan, V. S. Kumar, V. Mangla, V. Ionescu, V. Poenaru, V. T. Mihailescu, V. Ivanov, W. Li, W. Wang, W. Jiang, W. Bouaziz, W. Constable, X. Tang, X. Wu, X. Wang, X. Wu, X. Gao, Y. Kleinman, Y. Chen, Y. Hu, Y. Jia, Y. Qi, Y. Li, Y. Zhang, Y. Zhang, Y. Adi, Y. Nam, Yu, Wang, Y. Zhao, Y. Hao, Y. Qian, Y. Li, Y. He, Z. Rait, Z. DeVito, Z. Rosnbrick, Z. Wen, Z. Yang, Z. Zhao, and Z. Ma. The llama 3 herd of models, 2024.

[6] M. Holzer, A. Klein, and M. Kutrib. On the np-completeness of the nurikabe pencil puzzle and variants thereof. 2008.

[7] A. Ishay et al. Leveraging large language models to generate answer set programs. *arXiv preprint arXiv:2307.07699*, 2023.

[8] K. S. Meel and S. Akshay. Sparse hashing for scalable approximate model counting: Theory and practice. *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, 2020.

[9] OpenAI. Gpt-5 system card, 2025.

[10] J. Seely et al. Sudoku-bench: Evaluating language models on creative long-horizon logic. *arXiv preprint arXiv:2505.16135*, 2025.

[11] M. Soos and K. S. Meel. Arjun: An efficient independent support computation technique and its applications to counting and sampling. *arXiv preprint arXiv:2110.09026*, 2021.

[12] S. Tyagi et al. Gridpuzzle: Evaluating large language models on logic grid puzzles. *arXiv preprint arXiv:2407.14790*, 2024.

[13] G. van der Knijff. Solving and generating puzzles with a connectivity constraint. BSc Thesis, Radboud University, 2021.

[14] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. arXiv:2203.11171.

[15] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu. Qwen3 technical report, 2025.

[16] S. Yao et al. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS 2023 Workshop on Reasoning*, 2023.

[17] S. Yao, J. Zhao, D. Yu, and et al. React: Synergizing reasoning and acting in language models. *arXiv preprint*, 2022. arXiv:2210.03629.

We release all code and data for reproducibility at https://github.com/bxptr/spatial.

Acknowledgments

We thank Modal for compute, OpenRouter for LLM inference, and the UniGen and Arjun teams for tools that made this work possible.

A Projection Sampling Example

We illustrate projection bias and how XOR hashing corrects it with a toy 2×2 *Nurikabe* puzzle. The task variables are $T = \{B_{0,0}, B_{0,1}, B_{1,0}, B_{1,1}\}$ indicating which cells are black. Suppose the seed is at $(0, 0)$ with size 2, requiring two white cells forming a connected island including $(0, 0)$, with remaining cells black forming a connected sea. The encoding introduces auxiliary variables A for flow (enforcing sea connectivity) and island membership. The CNF F over $X = T \cup A$ has clauses ensuring that $B_{0,0} = 0$ (seed cell is white), that no 2×2 all-black block exists (via $\neg B_{0,0} \vee \neg B_{0,1} \vee \neg B_{1,0} \vee \neg B_{1,1}$), and that sea connectivity holds via flow balances with auxiliary flow variables.

Suppose two valid puzzle solutions exist: (a) cells $(0, 0)$ and $(0, 1)$ white, $(1, 0)$ and $(1, 1)$ black; (b) cells $(0, 0)$ and $(1, 0)$ white, $(0, 1)$ and $(1, 1)$ black. Due to flow encoding details, solution (a) might admit three distinct auxiliary completions while solution (b) admits only one. Naive uniform sampling over $S(F)$ would select (a) with probability $3/4$ and (b) with probability $1/4$, introducing bias. By identifying $I = T$ as an independent support, we add one XOR constraint, say $B_{0,1} \oplus B_{1,0} = \alpha$, with random α. Each choice of α selects a subset of $\pi(S(F))$: $\alpha = 0$ might isolate solution (a) while $\alpha = 1$ isolates solution (b). Sampling uniformly over α and then within the resulting cell yields near-uniform coverage over the two puzzle configurations, independent of auxiliary variable counts. Note that both solutions are symmetric under 90-degree rotation, so canonicalization would map them to the same representative, requiring careful handling to maintain distributional properties.

B Formal Constraint Encodings

We formalize the Boolean encodings that support distribution-aware generation and exact validation for both puzzle families. All grids are $R \times C$ with 4-neighborhood adjacency; cell indices are $(i, j) \in \{0, \dots, R - 1\} \times \{0, \dots, C - 1\}$. The encodings translate each puzzle's rules into CNF clauses over task and auxiliary variables, enabling SAT-based solving and sampling.

Slitherlink. Let the dot set be $\mathcal{V} = \{0, \dots, R\} \times \{0, \dots, C\}$. For each horizontal pair $(r, c) \leftrightarrow (r, c + 1)$ define $H_{r,c} \in \{0, 1\}$ $(0 \leq r \leq R,\ 0 \leq c < C)$; for each vertical pair $(r, c) \leftrightarrow (r + 1, c)$ define $V_{r,c} \in \{0, 1\}$ $(0 \leq r < R,\ 0 \leq c \leq C)$. A clued cell (i, j) with $\kappa_{i,j} \in \{0, 1, 2, 3, 4\}$ satisfies

$$H_{i,j} + H_{i+1,j} + V_{i,j} + V_{i,j+1} = \kappa_{i,j}.$$

At each dot $u \in \mathcal{V}$, let $\deg(u)$ be the sum of incident edge variables; enforce $\deg(u) = 2a_u$ with $a_u \in \{0, 1\}$. To rule out multiple cycles, enforce connectivity by a single-commodity flow on oriented chosen edges. Let $E_{u,v}$ denote the indicator of the undirected grid edge $\{u, v\}$ (equal to the appropriate H or V). Pick a root $s \in \mathcal{V}$ and require $a_s = 1$. Introduce flows $f_{u\to v} \geq 0$ with capacities $0 \leq f_{u\to v} \leq M\, E_{u,v}$ for $M = |\mathcal{V}|$, and node balances

$$\sum_{y} f_{u\to y} - \sum_{x} f_{x\to u} = \{\, a_{\,u}\, , u \neq s, -\Big(\sum_{w\in\mathcal{V}} a_w - 1\Big), u = s,$$

which, together with $\deg(u) \in \{0, 2\}$, enforces exactly one simple cycle [13]. The single-cycle property follows from a standard graph-theoretic argument: the degree constraint restricts the chosen edges to a union of disjoint cycles, and the flow connectivity requirement forces all chosen vertices into one component, hence exactly one cycle. At validation time, any vertex on the proposed loop can serve as root s.

Nurikabe. Let $B_{i,j} \in \{0,1\}$ indicate black (sea) at cell (i,j). For every 2×2 block $\{(i,j),(i+1,j),(i,j+1),(i+1,j+1)\}$,

$$B_{i,j} + B_{i+1,j} + B_{i,j+1} + B_{i+1,j+1} \leq 3.$$

Sea connectivity is enforced by a commodity flow on the cell graph. Pick a root cell s and require $B_s = 1$. Let $Y_{u,v} \in \{0,1\}$ gate adjacency with $Y_{u,v} \leq B_u$, $Y_{u,v} \leq B_v$, and $Y_{u,v} = Y_{v,u}$ (symmetric flow edges), and introduce flows $g_{u\to v} \geq 0$ with $0 \leq g_{u\to v} \leq M'Y_{u,v}$ for $M' = RC$. Impose balances

$$\sum_y g_{u\to y} - \sum_x g_{x\to u} = \{\, B_{\,u}\,, u \neq s, -\big(\sum_w B_w - 1\big), u = s,$$

which ensures that all black cells form a single 4-connected component [13].

For each numbered seed s at location $\ell(s)$ with target size t_s, introduce island membership indicators $X_{s,i,j} \in \{0,1\}$ satisfying:

$$B_{\ell(s)} = 0, \quad X_{s,\ell(s)} = 1, \quad X_{s,i,j} \leq 1 - B_{i,j}, \quad \sum_{i,j} X_{s,i,j} = t_s.$$

To ensure every white cell belongs to exactly one island:

$$\sum_s X_{s,i,j} = 1 - B_{i,j}.$$

To prevent distinct islands from touching orthogonally, for each pair of orthogonally adjacent cells $u \sim v$ and all seed pairs $s \neq t$:

$$X_{s,u} + X_{t,v} \leq 1.$$

Finally, add a connectivity constraint on $\{(i,j) : X_{s,i,j} = 1\}$ to ensure each island forms a single 4-connected component. These constraints capture the standard Nurikabe rules [4], including the requirement that islands do not touch orthogonally (diagonal contact is permitted). The validator independently verifies all properties to ensure robustness to encoding choices.

Encoding Complexity and Variable Counts. The encodings introduce different numbers of variables and constraints depending on puzzle type and size. For *Slitherlink* on an $R \times C$ grid, the task variables number $|T| = R(C+1) + C(R+1)$ (horizontal and vertical edge indicators), while auxiliary variables include $(R+1)(C+1)$ vertex activation indicators and $O(RC)$ flow variables, yielding $|X| = O(RC)$ total variables. The clue constraints produce $O(RC)$ clauses, degree constraints contribute $O(RC)$ clauses, and the flow encoding adds $O(RC)$ additional clauses, for a total of $O(RC)$ clauses overall. For *Nurikabe* on an $R \times C$ grid, task variables number $|T| = RC$ (cell color indicators). Auxiliary variables include $O(RC)$ flow variables for sea connectivity and, when island membership is encoded explicitly, $O(kRC)$ indicator variables for k island seeds. The 2×2 exclusion constraints contribute $O(RC)$ clauses, sea connectivity flow constraints add $O(RC)$ clauses, and island-size constraints introduce $O(kRC)$ clauses, yielding $O(kRC)$ clauses overall where k is typically small (under 10 in our benchmark). This disparity in auxiliary complexity, particularly the ratio of auxiliary to task variables, motivates the projection-aware sampling methodology: the number of auxiliary completions per task assignment can vary substantially, and naive uniform sampling over the full space $S(F)$ would systematically bias the distribution over puzzle solutions.

C Textual Puzzle Representation

Below we show an example of the textual representation of a *Nurikabe* puzzle. The exact grammar file can be found in our code, but this snippet shows the general problem description, which is supplemented with a prompt that describes rules and expected outputs, as well as output formats.

```
puzzle N;
grid {
    rows: 10;
    cols: 10;
}
rules {
    no_2x2: true;
    single_sea: true;
}
id "37";
givens {
    source(0, 0, 58);
    source(6, 7, 1);
    source(7, 2, 1);
    source(7, 4, 2);
    source(7, 9, 14);
}
```