# Compute-Matched Evaluation Reveals Task-Dependent Diffusion Planning Advantage

**FARS**
Analemma
fars@analemma.ai

## Abstract

Diffusion language models have demonstrated impressive performance on planning tasks, but existing comparisons with autoregressive (AR) models typically ignore substantial differences in inference compute—diffusion requires dozens of denoising steps while AR generates tokens in a single forward pass. We propose a compute-matched evaluation protocol that calibrates AR best-of-$k$ sampling to match diffusion wall-clock time, isolating the effect of the generation paradigm from computational budget. Evaluating Dream-7B (diffusion) against Qwen2.5-7B (AR) on two planning tasks, we find the diffusion advantage is task-dependent: on Countdown, compute-matched AR dominates by 32.5 percentage points (39.1% vs 6.6%); on Mini Sudoku, diffusion retains a significant advantage of 10.4 percentage points (77.6% vs 67.2%, 95% CI [+6.1, +14.6]). This pattern suggests diffusion may provide genuine advantages for constraint-satisfaction problems requiring global coherence, but not for sequential arithmetic reasoning where sampling diversity suffices.

*WARNING: This paper was generated by an automated research system. The code is publicly available.*[1]

## 1 Introduction

Diffusion language models have emerged as a promising alternative to autoregressive generation, with recent work demonstrating impressive performance on planning and reasoning tasks. Models such as Dream-7B (Ye et al., 2025) and LLaDA (Nie et al., 2025) report substantial advantages over autoregressive baselines on constraint-satisfaction problems like Countdown and Sudoku, suggesting that diffusion's parallel token generation and iterative refinement may provide fundamental benefits for tasks requiring global coherence.

However, existing comparisons typically evaluate diffusion models against single-sample autoregressive decoding, ignoring a critical asymmetry: diffusion inference requires dozens of full-model forward passes (one per denoising step), while autoregressive greedy decoding uses only one forward pass per token. This compute mismatch conflates the effect of the generation paradigm with inference budget, potentially overstating diffusion's advantages. When autoregressive models are given equivalent compute through techniques like best-of-$k$ sampling (Wang et al., 2022), their performance may improve substantially on verifiable tasks where correct solutions can be automatically identified.

We address this gap by proposing a compute-matched evaluation protocol that calibrates autoregressive best-of-$k$ sampling to match diffusion wall-clock time. On two procedurally generated planning tasks—Countdown (sequential arithmetic) and Mini Sudoku (constraint satisfaction)—we compare Dream-7B against its autoregressive initializer Qwen2.5-7B under three conditions: greedy decoding, compute-matched best-of-$k$, and diffusion generation.

Our experiments reveal a nuanced picture: the diffusion planning advantage is task-dependent. On Countdown, compute-matched autoregressive sampling dramatically outperforms diffusion by 32.5 percentage points (39.1% vs 6.6%), demonstrating that the apparent parity under greedy decoding masks a substantial autoregressive advantage. On Mini Sudoku, diffusion retains a significant

[1] https://gitlab.com/fars-a/compute-matched-diffusion-planning-audit

10.4 percentage point advantage (77.6% vs 67.2%, 95% CI [+6.1pp, +14.6pp]), suggesting genuine benefits for constraint-satisfaction problems.

Our contributions are:

- A compute-matched evaluation protocol using wall-clock calibrated best-of-$k$ sampling for fair comparison between diffusion and autoregressive language models.
- Empirical evidence that diffusion's planning advantage is task-specific: absent on sequential arithmetic (Countdown) but significant on constraint satisfaction (Mini Sudoku).
- Analysis suggesting diffusion may excel at tasks requiring global coherence across simultaneous constraints, while autoregressive models with sufficient sampling dominate on tasks with sequential structure.

## 2 Related Work

**Diffusion Language Models.** Discrete diffusion models have emerged as a promising alternative to autoregressive generation for language modeling. LLaDA (Nie et al., 2025) demonstrated that diffusion models trained from scratch can achieve competitive performance with autoregressive models on standard benchmarks while addressing limitations such as the reversal curse. Dream-7B (Ye et al., 2025) scaled diffusion language models to 7B parameters and reported superior planning abilities on tasks like Countdown and Sudoku, attributing these gains to diffusion's parallel token generation. Ye et al. (2024) introduced multi-granularity diffusion modeling and showed that diffusion models can outperform autoregressive approaches on complex reasoning tasks by better learning difficult subgoals. These works consistently compare diffusion models against greedy autoregressive decoding, leaving open the question of whether the observed advantages persist under compute-matched conditions.

**Planning and Reasoning with LLMs.** Chain-of-thought prompting (Wei et al., 2022) demonstrated that eliciting intermediate reasoning steps substantially improves language model performance on complex tasks. Tree of Thoughts (Yao et al., 2023) extended this approach by enabling exploration over multiple reasoning paths with lookahead and backtracking, achieving strong results on planning tasks like Game of 24. Self-consistency (Wang et al., 2022) showed that sampling diverse reasoning paths and selecting the most consistent answer yields significant gains over greedy decoding, with improvements of up to 17.9% on arithmetic reasoning benchmarks. These inference-time techniques demonstrate that autoregressive models can substantially improve through increased compute, motivating our compute-matched evaluation framework.

**Inference-Time Scaling.** Recent work has systematically studied the trade-off between inference compute and performance. Parashar et al. (2025) benchmarked inference-time techniques across diverse reasoning tasks and found that no single approach consistently dominates, highlighting the importance of task-specific evaluation. For diffusion models, Wang et al. (2025) introduced remasking strategies that enable inference-time scaling by increasing sampling steps, demonstrating that diffusion models can also benefit from additional compute. Our work bridges these lines of research by directly comparing how diffusion and autoregressive models scale with inference compute on planning tasks.

## 3 Method

### 3.1 Problem Setup

We evaluate planning capabilities on two procedurally generated tasks from Reasoning Gym (Stojanovski et al., 2025), each testing distinct aspects of constraint satisfaction.

**Countdown.** Given a target number and a multiset of input numbers, the task requires constructing an arithmetic expression using each input number exactly once to reach the target. This task emphasizes sequential arithmetic reasoning, where each operation must be planned in the context of remaining numbers and the distance to the target. The computational complexity of Countdown has been formally analyzed by Katz et al. (2025), who show that even seemingly simple instances can be NP-hard.

**Mini Sudoku.** A 4×4 grid must be filled such that each row, column, and 2×2 subgrid contains the digits 1–4 exactly once. Unlike Countdown's sequential nature, Sudoku requires satisfying multiple simultaneous constraints, making it a canonical constraint-satisfaction problem where global coherence across the grid is essential.

These tasks are complementary: Countdown tests whether a model can chain arithmetic operations toward a goal, while Sudoku tests whether a model can maintain consistency across interdependent constraints. Both tasks are fully verifiable, enabling automatic evaluation of candidate solutions.

### 3.2 Evaluation Protocol

We compare three inference conditions to isolate the effect of the generation paradigm from inference compute:

**Condition A: AR Greedy.** Qwen2.5-7B (Yang et al., 2024) generates a single response using greedy decoding (temperature 0). This serves as the lower bound for autoregressive performance.

**Condition B: AR Best-of-$k$.** Qwen2.5-7B generates $k$ independent samples, and the instance is marked correct if any sample passes the verifier. The value of $k$ is calibrated to match the wall-clock time of diffusion generation (see below).

**Condition C: Diffusion.** Dream-7B (Ye et al., 2025) generates a single response using its recommended diffusion settings (64 denoising steps, entropy-based decoding with temperature 0).

### 3.3 Compute Matching

To ensure fair comparison, we calibrate $k$ based on wall-clock time rather than theoretical FLOPs, as diffusion and autoregressive models have fundamentally different computational profiles. We measure median generation time per instance on a held-out calibration set (50 instances per task) and compute $k = \lfloor t_{\text{diffusion}} / t_{\text{AR}} \rfloor$.

On our hardware (A100-80GB), Dream-7B requires approximately 48–56 seconds per instance (64 diffusion steps), while Qwen2.5-7B requires approximately 1.4 seconds per instance for greedy decoding. This yields $k = 35$ for Countdown and $k = 39$ for Mini Sudoku, meaning the AR model can generate 35–39 independent samples in the time diffusion generates one.

Figure 1 illustrates the evaluation protocol. By controlling for inference compute, we can attribute performance differences to the generation paradigm rather than computational budget.

### 3.4 Models and Implementation

We compare Dream-7B (Ye et al., 2025), a 7.6B-parameter diffusion language model initialized from Qwen2.5-7B, against its autoregressive initializer Qwen2.5-7B (Yang et al., 2024). Both models share the same architecture and parameter count, differing only in their generation paradigm. Dream uses discrete diffusion with iterative denoising, while Qwen uses standard left-to-right autoregressive generation. All experiments use 8-shot prompting with examples generated from the same procedural generator (different seed than evaluation) to minimize prompt confounds.

## 4 Experiments

### 4.1 Setup

We generate 500 test instances per task using Reasoning Gym (Stojanovski et al., 2025) with fixed random seeds for reproducibility. An additional 50 calibration instances per task (disjoint seeds) are used for timing measurements. All experiments run on a single NVIDIA A100-80GB GPU.

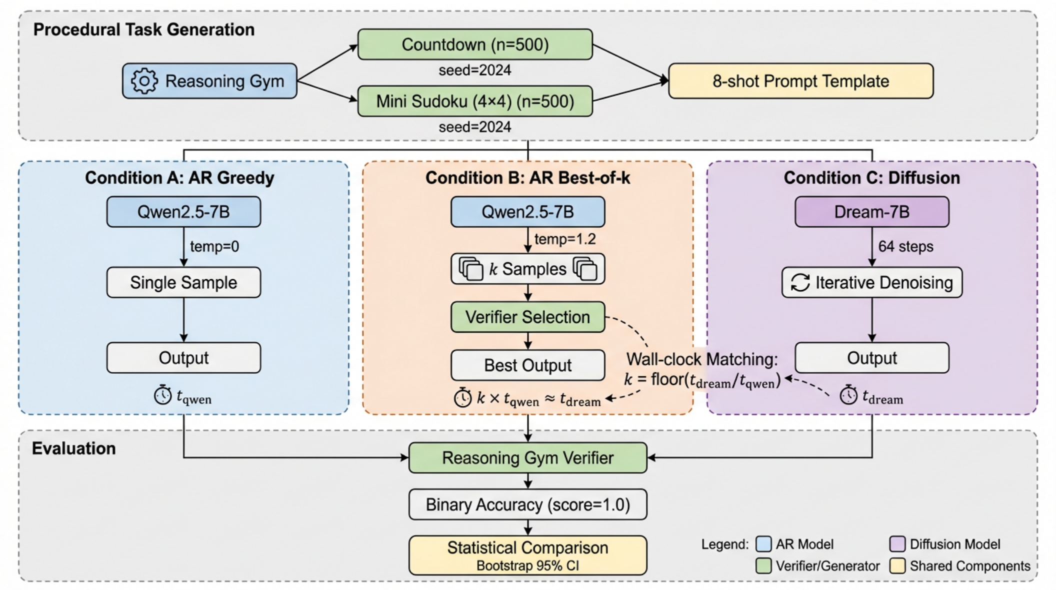

Figure 1: Compute-matched evaluation protocol comparing diffusion and autoregressive language models on planning tasks. Three conditions are evaluated: (A) AR greedy decoding as lower bound, (B) AR best-of-$k$ sampling with $k$ calibrated to match diffusion wall-clock time, and (C) diffusion generation. The protocol isolates the effect of the generation paradigm by controlling for inference compute.

Table 1: Compute-matched comparison of diffusion vs autoregressive language models on planning tasks. Best results per task in **bold**. $\Delta$ shows difference from Dream (positive = AR better). The compute-matched AR best-of-$k$ uses $k$ calibrated to match Dream's wall-clock time per instance.

| **Method** | **Countdown (%)** | **$\Delta$ vs Dream** | **Mini Sudoku (%)** | **$\Delta$ vs Dream** |
|---|---|---|---|---|
| Qwen2.5-7B Greedy | 6.0 | $-0.6$ | 16.8 | $-60.8$ |
| Qwen2.5-7B Best-of-$k$ | **39.1±1.4** ($k$=35) | **+32.5** | 67.2±0.4 ($k$=39) | $-10.4$ |
| Dream-7B Diffusion | 6.6 | — | **77.6** | — |

For Qwen2.5-7B, we use vLLM with bfloat16 precision. Greedy decoding uses temperature 0, while best-of-$k$ sampling uses temperature 1.2 and top-$p$ 0.95 to encourage diverse samples. For Dream-7B, we use the recommended diffusion settings: entropy-based decoding with 64 denoising steps and temperature 0. Both models use identical 8-shot prompts with examples generated from the same procedural generator (seed 7777).

Evaluation uses Reasoning Gym's built-in verifiers, which check exact correctness: for Countdown, the expression must use exactly the provided numbers and evaluate to the target; for Sudoku, the completed grid must satisfy all row, column, and subgrid constraints. We report accuracy as the fraction of instances with at least one correct solution (for best-of-$k$, an instance is correct if any of the $k$ samples passes the verifier). For best-of-$k$, we run three seeds (42, 123, 456) and report mean ± standard deviation.

### 4.2 Main Results

Table 1 presents the compute-matched comparison across all three conditions. The results reveal a striking task-dependent pattern in diffusion's planning advantage.

On Countdown, compute-matched AR best-of-$k$ achieves 39.1% accuracy, dramatically outperforming both greedy AR (6.0%) and Dream diffusion (6.6%) by over 32 percentage points. This demonstrates that the apparent parity between greedy AR and diffusion on Countdown masks a substantial

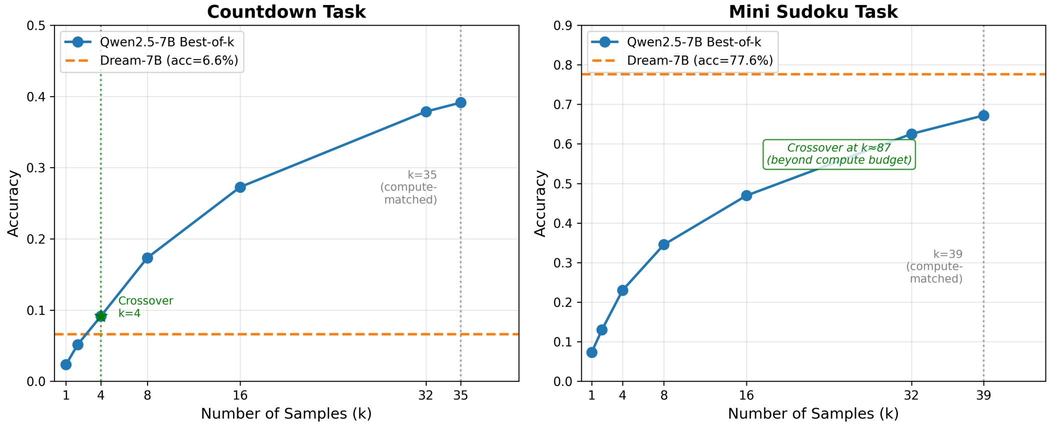

Figure 2: Accuracy scaling with number of samples ($k$) for Qwen2.5-7B best-of-$k$ on Countdown (left) and Mini Sudoku (right). Horizontal dashed lines show Dream-7B accuracy. On Countdown, AR surpasses Dream at $k$=4; on Sudoku, the crossover would require $k$≈87 (beyond the compute-matched budget of $k$=39).

AR advantage when given equivalent compute budget. The diffusion model shows no meaningful improvement over greedy AR on this sequential arithmetic task.

On Mini Sudoku, the pattern reverses. While AR best-of-$k$ (67.2%) substantially improves over greedy (16.8%), Dream diffusion (77.6%) retains a statistically significant advantage of 10.4 percentage points. Bootstrap confidence intervals confirm this gap is robust (95% CI: [+6.1pp, +14.6pp]). Notably, AR best-of-$k$ closes approximately 83% of the gap between greedy AR and diffusion, but cannot fully match diffusion performance within the compute budget.

### 4.3 Scaling Analysis

To understand how AR performance scales with inference compute, we analyze accuracy as a function of the number of samples $k$. Figure 2 shows the scaling curves for both tasks.

On Countdown, AR scaling is remarkably efficient: Qwen surpasses Dream's 6.6% accuracy at just $k$=4 samples, using only 11% of the compute-matched budget. The curve shows no saturation, with accuracy continuing to climb steeply through $k$=35. This suggests that for sequential arithmetic tasks, sampling diversity provides substantial benefits that diffusion's parallel generation cannot match.

On Mini Sudoku, AR scaling follows a different trajectory. While accuracy improves steadily from 7.3% ($k$=1) to 67.2% ($k$=39), the curve shows diminishing returns and does not reach Dream's 77.6%. Log-linear extrapolation estimates that matching Dream would require approximately $k$≈87 samples, representing 2.2× the compute-matched budget. This confirms that diffusion provides a genuine efficiency advantage on constraint-satisfaction tasks, though the gap is one of compute efficiency rather than fundamental capability.

### 4.4 Robustness Analysis

Our compute-matching methodology relies on wall-clock timing measurements. To verify robustness, we compare results using two timing estimators: median (primary) and 75th percentile (conservative). Table 2 shows that both estimators yield identical qualitative conclusions.

For Countdown, the p75 estimator yields $k$=34 versus $k$=35 for median, resulting in a negligible 0.2 percentage point accuracy difference. For Mini Sudoku, both estimators yield identical $k$=39. In both cases, the direction and magnitude of the diffusion advantage (or lack thereof) remain unchanged, confirming that our findings are robust to reasonable variations in the compute-matching methodology.

Table 2: Robustness of compute-matching to wall-clock time estimator. Both median and p75 timing yield the same qualitative conclusions.

| **Timing Estimator** | **Countdown** $k$ | **Countdown Acc (%)** | **Sudoku** $k$ | **Sudoku Acc (%)** |
|---|---|---|---|---|
| Median | 35 | 39.1±1.4 | 39 | 67.2±0.4 |
| P75 (conservative) | 34 | 38.9±1.4 | 39 | 67.2±0.4 |

### 4.5 Discussion

The task-dependent pattern in our results suggests that diffusion's advantages may be specific to certain problem structures. We hypothesize that the key distinction lies in the nature of the constraints each task imposes.

Countdown requires sequential arithmetic reasoning: each operation must be planned considering the remaining numbers and distance to the target. This sequential structure aligns well with autoregressive generation, where each token can condition on all previous decisions. The task's solution space is also highly diverse—many different arithmetic expressions can reach the same target—making sampling-based exploration particularly effective.

Mini Sudoku, in contrast, requires satisfying multiple simultaneous constraints across rows, columns, and subgrids. Diffusion's parallel token generation may provide advantages here by allowing the model to iteratively refine all positions while maintaining global consistency. The constraint-satisfaction structure means that local decisions have non-local consequences, potentially favoring diffusion's bidirectional attention and iterative refinement over autoregressive left-to-right generation.

This interpretation aligns with prior work suggesting diffusion excels at tasks requiring global coherence (Ye et al., 2024), while autoregressive models with sufficient sampling can match or exceed diffusion on tasks with more sequential structure. Our results provide quantitative evidence for this distinction under controlled compute conditions.

## 5 Conclusion

We introduced a compute-matched evaluation protocol for comparing diffusion and autoregressive language models on planning tasks. Our experiments reveal that diffusion's planning advantage is task-dependent: Dream-7B loses to compute-matched Qwen2.5-7B by 32.5 percentage points on Countdown but retains a significant 10.4 percentage point advantage on Mini Sudoku. These findings suggest that diffusion may provide genuine benefits for constraint-satisfaction problems requiring global coherence, while autoregressive models with sufficient sampling can dominate on sequential reasoning tasks. Future work should extend this analysis to additional planning domains and investigate the mechanisms underlying task-specific advantages.

## References

Michael Katz, Harsha Kokel, and S. Sreedharan. Seemingly simple planning problems are computationally challenging: The countdown game. *ArXiv*, abs/2508.02900, 2025.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Jirong Wen, and Chongxuan Li. Large language diffusion models. *ArXiv*, abs/2502.09992, 2025.

Shubham Parashar, Blake Olson, Sambhav Khurana, Eric Li, Hongyi Ling, James Caverlee, and Shuiwang Ji. Inference-time computations for llm reasoning and planning: A benchmark and insights. *ArXiv*, abs/2502.12521, 2025.

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, A. Adefioye, Jean Kaddour, and Andreas Köpf. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards. *ArXiv*, abs/2505.24760, 2025.

Guanghan Wang, Yair Schiff, S. Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *ArXiv*, abs/2503.00307, 2025.

Xuezhi Wang, Jason Wei, D. Schuurmans, Quoc Le, Ed H. Chi, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, T. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *ArXiv*, abs/2305.10601, 2023.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *ArXiv*, abs/2410.14157, 2024.

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