Boosting Inference with Guided Reasoning: Stochastic Exploration for Recursive Models

Andrew Corbett[1,2]* **Archit Sood**[1] **Anna Tzatzopoulou**[1]
Sai-Aakash Ramesh[1,2] **Tim Dodwell**[1,2]

[1]digiLab, UK [2]University of Bristol, UK

Abstract

Recent work on recursive architectures has shown that tiny neural networks can be surprisingly powerful on structured reasoning tasks. The trick is to model reasoning trajectories with a latent dynamical system. We argue that the inference-time behaviour of these architectures is best understood as approximate inference over latent reasoning trajectories, with deterministic recursion as the one-particle, zero-noise limit. We make this view operational through *guided reasoning*: stochastic perturbations of the reasoning dynamics propose neighbouring trajectories, and the model's existing early-stopping head reweights them online. The framework yields three diagnostics computable from inference traces: local stability, guide alignment, with a label-free necessary condition, and cloud-token entropy. These predict whether the procedure will help and which of its outputs to trust. On Sudoku-Extreme it lifts exact-solve accuracy from 85.9% to 98.0% without retraining, and on PPBench, a four-type pencil-puzzle benchmark, from 62.5% to 77.9% with no tuning; on Maze-Hard the diagnostics flag a misaligned guide, as validation performance later confirms. The same machinery thus characterises both when recursive reasoning has room to improve at the trajectory level and when the model's internal guide can recover said improvements. Code is available at `https://github.com/digiLab-ai/bigr`.

1 Introduction

While large, compute-intensive models have dominated recent progress on structured reasoning tasks, recursive architectures (Wang et al., 2025; Jolicoeur-Martineau, 2025) have shown breakthrough performance, matching large language models, with neural backbones of as little as $7M$ parameters. This combination of compactness and performance makes them well-positioned for AI deployment in settings with strict latency, privacy, and auditability constraints, where they can offer practical advantages over hyper-scaled alternatives. We argue that what these architectures do at inference time is best understood as approximate inference over latent reasoning trajectories, with deterministic recursion as the one-particle, zero-noise limit.

Prototypical architectures, such as the Tiny Recursive Model (TRM) of Jolicoeur-Martineau (2025), already contain ingredients that suggest a richer inference procedure at test time, including learned early-stopping Q-heads and mixed-frequency latent updates. These features have emerged predominantly through experimental iteration (Wang et al., 2025; Jolicoeur-Martineau, 2025), leaving open how they should be interpreted or used once training is complete. We study this question through inference-time adaptation: given a frozen pre-trained backbone, how should additional computation be allocated across alternative latent reasoning trajectories, and when can we predict that this allocation will succeed?

*Correspondence to `andy.corbett@digilab.ai`.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

We make this view operational without retraining. Stochastic perturbations of the inner recursion induce a proposal over neighbouring latent trajectories; the model's Q-head, repurposed as a learned guide, reweights them online via a Feynman–Kac tilt. The deterministic recursion of Jolicoeur-Martineau (2025) sits inside this framework as the one-particle, zero-noise limit, allowing direct comparison with the original model. In this work we evaluate our method on structured reasoning tasks. We focus on zero-retraining, meaning the TRM backbone and jointly pre-trained Q-head guide are held fixed.

Stochastic exploration substantially increases the number of successful trajectories in the particle cloud, and guided selection identifies them when the guide is aligned. On Sudoku-Extreme (Wang et al., 2025), our method improves exact-solve accuracy from $85.9\pm0.6\%$ to $98.0\pm0.2\%$, solving previously out-of-reach failure cases and recovering baseline performance at much lower depths (Figure 1).

More generally, our procedure helps when two operating conditions are met. Stochastic exploration must be locally stable so alternative trajectories remain recoverable (Proposition 1), and the Q-head guide must be aligned with the ability to concentrate particle mass on successful trajectories (Lemma 1). Alignment admits both a labelled diagnostic, the class-conditional gap Δ_n, and a label-free necessary condition, the in-cloud guide-spread bound (18), which upper-bounds the mass shift any tempering can produce. On Sudoku-Extreme both conditions hold, and our method closes nearly the entire oracle gap. On Maze-Hard exploration is stable but the Q-head guide is too flat: the spread bound flags this label-free, a priori (Figure 3), and evaluation confirms the predicted non-improvement (Table 1). The lesson is that recursive models leave useful trajectory-level headroom on the table, but recovering it requires both stable exploration and a ranking-aligned internal guide.

Our method also yields a third diagnostic: uncertainty statistics for output reliability. We show that Shannon entropies of the final-step and path-integrated token marginals correlate with token error risk and can be used to support selective abstention (§5.3, Figure 4).

Our contributions are as follows:

- **An approximate inference framework for recursive reasoning models:** we introduce (i) a stochastic proposal kernel for inference-time exploration and (ii) a guided selection algorithm to identify successful trajectories and reweight the distribution; deterministic recursion is recovered exactly in the one-particle, zero-noise limit with the Q-head guide.
- **Three computable diagnostics to audit performance and reliability:** a stability bound (Proposition 1), an alignment characterisation (Lemma 1), and uncertainty statistics (§5.3).
- **Training-free improvement in exact-solve accuracy:** guided inference improves exact-solve accuracy from 85.9% to 98.0% on Sudoku-Extreme and from 62.5% to 77.9% on PPBench, without updating model parameters. Compute-matched comparisons isolate the improvements to 'guiding' rather than sampling alone, while sequential reweighting improves robustness; see Table 1. On Maze-Hard, our label-free diagnostics correctly identify a guide too flat to exploit the available oracle-particle headroom.

2 Related work

Both the HRM and TRM algorithms (Wang et al., 2025; Jolicoeur-Martineau, 2025) build on earlier recurrent relational networks for constraint-satisfaction puzzles (Palm et al., 2018), showing that strong structured reasoning emerges from repeatedly applying a small recurrent core. Efstathiou and Balwani (2026) characterise TRM's recursion as 'searching' in an attractor landscape, observing that on Sudoku-Extreme correct solutions sit within reach of the deterministic path: "it's in there somewhere". We add a probabilistic description in which the Q-head finds them. On Maze-Hard, where no other line of work reports inference-time improvement, our Q-spread diagnostic predicts this behaviour whilst also identifying nearly five points of attainable improvement in the particle cloud: we show "it's in there, too". Dillon (2026) uses halting speed under 'Adaptive Computation Time' (Graves, 2016) as a correctness signal; our method weights the cloud with the head's *value* rather than its *timing*, on fixed weights. These different approaches compose in application.

Most TRM-adjacent work modifies the model or its training, or analyses what the trained model *does*: McGovern (2025) updates TRM weights at test time on each task's demonstration pairs; Asadulaev

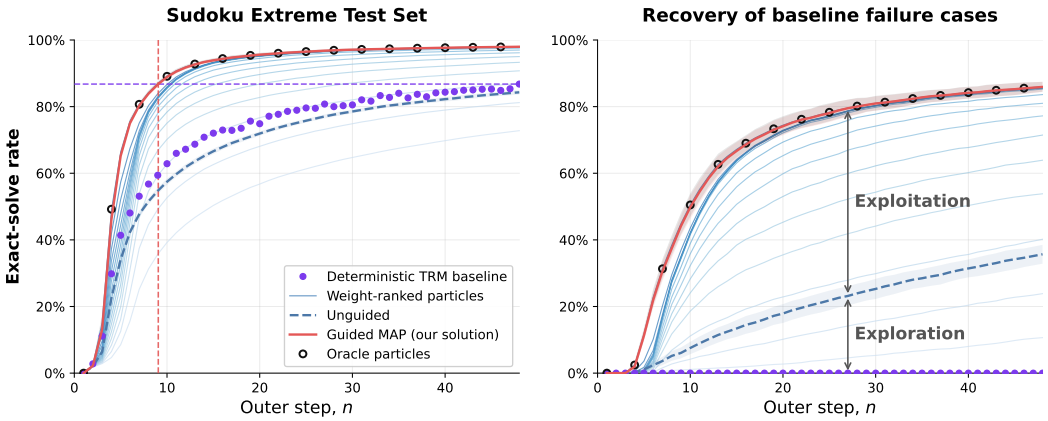

Figure 1: **Inference-time improvement on Sudoku-Extreme.** *Left:* Our method reaches $98.0 \pm 0.2\%$ exact-solve accuracy on the test split, improving over the deterministic TRM baseline ($85.9 \pm 0.6\%$) and a single unguided rollout by step 10. *Right:* On deterministic-failure cases, our method solves $85.9 \pm 1.5\%$ (with $35.7 \pm 2.8\%$ for a single unguided rollout; oracle of the guided cloud 86.0%). Results are over 5 seeds $\times$ 5 folds; $\pm$ and shaded bands are the fold–seed s.d.

et al. (2026) reinterpret the inner step as policy improvement and modify it; Abubacar et al. (2026) freeze a pretrained encoder and train a TRM as a quality estimator; concurrently, Sghaier et al. (2026) add Gaussian noise to the TRM recursion and keep the best of K independent rollouts by the Q-head; their strategy is the terminal special case of our filtering approach, with resampling off and the tilt applied once at step N. Table 1 compares the two on a matched budget and under a corrupted guide. Roye-Azar et al. (2025) investigate the inductive biases and test-time compute of the ARC Prize TRM checkpoint. Looped and latent-compute language models vary recursion depth at inference (Koishekenov et al., 2025; Zhu et al., 2025; Bae et al., 2025). Our setting is complementary on each of these axes: weights, latent depth, and outputs are all unchanged; we only reallocate compute across stochastic local trajectories and reweight them with the existing Q-head.

At a higher level, the work sits alongside inference-time-compute methods for language models that sample chains, branch search, or apply verifiers (Wei et al., 2022; Wang et al., 2023; Yao et al., 2023; Lightman et al., 2024; Snell et al., 2025). We run the same probabilistic machinery (sequential Monte Carlo with a learned twist) but over continuous latent reasoning states inside a small recursive model, rather than over discrete token chains where the analogous twist function appears in twisted SMC (Zhao et al., 2024). Recursive computation also relates to Deep Equilibrium Models (Bai et al., 2019) and unrolled policy iteration (Behzadian et al., 2026).

Methodologically, we draw on sequential importance sampling and the Feynman–Kac view of weighted path measures (Liu and Chen, 1998; Gilks and Berzuini, 2001; Del Moral et al., 2006); the Q-head reweighting is a generalised-Bayes tilt (Bissiri et al., 2016). Our cloud-uncertainty statistics relate to entropy-based uncertainty diagnostics for language models (Kuhn et al., 2023; Farquhar et al., 2024).

3 Background on recursive models

TRM-style architectures (Jolicoeur-Martineau, 2025) collapse the two separate modules of HRM (Wang et al., 2025) into one module simulating two frequencies. This is applied by a shared backbone f_θ evaluating two nested loops. An *inner loop* refines a latent state $z \in \mathcal{Z}$ and an *outer loop* reads out an answer $y \in \mathcal{Y}$, both in $\mathcal{Y} = \mathcal{Z} := \mathbb{R}^{L \times D}$ (L sequence length, D feature dimension). We pair them as the joint state $h_n := (y_n, z_n) \in \mathcal{H} := \mathcal{Y} \times \mathcal{Z}$ at outer step n. In this work we consider the backbone parameters θ fixed, and f_θ as a pre-trained deterministic function. For a fixed embedded input $x \in \mathbb{R}^{L \times D}$, the inner recursion, for $m = 1, \ldots, M-1$, is given by the recursive formula

$$z_{n,1} = f_\theta(x + y_n + z_n), \qquad z_{n,m+1} = f_\theta(x + y_n + z_{n,m}), \tag{1}$$

and the outer step, through (1), reads out the next answer from the final inner latent step $z_{n,M}$:

$$h_n = (y_n, z_n) \longmapsto \big(f_\theta(y_n + z_{n,M}),\, z_{n,M}\big). \tag{2}$$

Iterating (1)–(2) from a learned initial state $h_0 = (y_0, z_0)$ produces a deterministic trajectory $h_0, h_1, \ldots, h_N$, with decoded answers $a_n := \mathrm{Decode}(y_n)$ at each outer step $n = 1, \ldots, N$. The inner depth M is fixed at inference by the pre-training value, $M = 6$ for Sudoku-Extreme and $M = 4$ for Maze-Hard (Appendix A).

An early-stopping utility. The architecture additionally carries a learned scalar Q-head $\tilde{q}_\theta : \mathcal{Y} \to \mathbb{R}$, trained under detached BCE supervision as a binary classifier for whether a_n matches the ground-truth solution. Jolicoeur-Martineau (2025) use it to trigger Adaptive Computation Time (Graves, 2016). We read this as the model's internal logit-scale estimate of halting success, and §4 uses it as the natural score on which to tilt the predictive law.

Notation. We identify tensors in $\mathbb{R}^{L\times D}$ with their vectorisations in $\mathbb{R}^{LD}$ and write $\|\cdot\|$ for the induced Euclidean / Frobenius norm; the same symbol is used throughout §4–§5. For isotropic Gaussian noise $\xi_m \sim \mathcal{N}(0, I_{LD})$ the second-moment bound in Proposition 1 satisfies $\nu^2 = \mathbb{E}\|\xi_m\|^2 = LD$. We write $\mathrm{sigmoid}(y) := (1 + e^{-y})^{-1}$ for the sigmoid link function.

4 Guided reasoning for recursive models

We build on the deterministic recursion of §3 to construct a path measure over latent reasoning trajectories: a stochastic kernel, generalising the inner-recursion (1), proposes neighbouring trajectories (§4.1); the Q-head reweighting tilts the predictive law toward correct solutions (§4.2); and a bootstrap particle approximation makes the resulting Feynman–Kac flow tractable (§4.3).

4.1 Exploration: the stochastic proposal

To widen the inner TRM recursion for exploration, let $\xi_{n,0:M}$ be independent mean-zero noise variables and $\sigma \geq 0$ the exploration scale. Define

$$\zeta_{n,1} = z_{n,1} + \sigma\xi_{n,0}, \qquad \zeta_{n,m+1} = f_\theta(x + y_n + \zeta_{n,m}) + \sigma\xi_{n,m}, \tag{3}$$

for $m = 1, \ldots, M-1$, and the stochastic outer update

$$h_n = (y_n, z_n) \longmapsto \big(f_\theta(y_n + \zeta_{n,M}) + \sigma\xi_{n,M},\, \zeta_{n,M}\big). \tag{4}$$

The proposal kernel is then the distribution induced by this random transition:

$$K_{f_\theta,x}(A \mid h_n) := \mathbb{P}\big(\big(f_\theta(y_n + \zeta_{n,M}) + \sigma\xi_{n,M},\, \zeta_{n,M}\big) \in A\big), \tag{5}$$

for measurable $A \subseteq \mathcal{H}$, with implicit dependence on the inner depth M and noise scale σ through (3). At $\sigma = 0$ the kernel collapses to the Dirac measure on (2), recovering the deterministic TRM update. For $\sigma > 0$ it widens the inner trajectory locally without changing backbone weights.

4.2 Exploitation: the Q-head tilt

Local exploration alone defines the proposal kernel (5); we select among proposals by reweighting with the Q-head $\tilde{q}_\theta$, instantiated as a Feynman–Kac tilt (Del Moral et al., 2006) of the predictive law. Let π_n denote the law of the guided state after n outer steps. Propagation through the proposal kernel gives the predictive law

$$\pi_{n+1|n}(\mathrm{d}h') = \int_{\mathcal{H}} K_{f_\theta,x}(\mathrm{d}h' \mid h)\, \pi_n(\mathrm{d}h), \tag{6}$$

which we tilt by the Q-head. Working on the log-probability scale,

$$q_\theta(h) := \log \mathrm{sigmoid}\big(\tilde{q}_\theta(y)\big), \tag{7}$$

where $h = (y, z)$, so that $G_\beta(h) := \exp(\beta q_\theta(h)) \in (0, 1]$ is a proper potential for every inverse temperature $\beta \geq 0$. The guided update is the exponentially tilted β-law

$$\pi_{n+1,\beta}(\mathrm{d}h) \propto G_\beta(h)\, \pi_{n+1|n}(\mathrm{d}h), \tag{8}$$

a single mutation–selection step of the corresponding Feynman–Kac flow. Applying (8) once, at step N only and without resampling, reduces to a best-in-cloud selection by the Q-head; see Appendix H.

4.3 Finite-particle implementation

We approximate (6) with an empirical measure $\pi_n(\mathrm{d}h) \approx \sum_{s=1}^{S} w_n^{(s)} \delta_{h_n^{(s)}}(\mathrm{d}h)$. Propagating each particle through the proposal kernel gives the predictive mixture

$$\pi_{n+1|n}(\mathrm{d}h) \approx \sum_{s=1}^{S} w_n^{(s)} \, K_{f_\theta, x}(\mathrm{d}h \mid h_n^{(s)}), \tag{9}$$

and the update therefore decomposes into exploration (§4.1) and guide-based exploitation (8),

$$\text{Exploration:} \quad h_{n+1}^{(s)} \sim K_{f_\theta, x}(\cdot \mid h_n^{(s)}), \tag{10}$$

$$\text{Exploitation:} \quad w_{n+1}^{(s)} \propto w_n^{(s)} \, G_\beta\big(h_{n+1}^{(s)}\big). \tag{11}$$

Algorithm 1 Guided reasoning for recursive models. Hyperparameters: S particles, outer depth N, inner depth M, noise scale $\sigma \geq 0$, inverse temperature $\beta \geq 0$, ESS threshold $\tau_{\mathrm{ESS}} \in (0, 1]$.

Require: input x, initial state h_0, backbone f_θ, guide q_θ, decoder 'Decode'

function TRAJECTORYSAMPLER($h = (y, z)$) ▷ sample the proposal kernel (5)
  draw $\xi_{0:M} \sim \mathcal{N}(0, I_{LD})$
  $\zeta_1 \leftarrow f_\theta(x + y + z) + \sigma\xi_0$
  for $m = 1, \ldots, M-1$ **do**
    $\zeta_{m+1} \leftarrow f_\theta(x + y + \zeta_m) + \sigma\xi_m$ ▷ stochastic inner-recursion (3)
  end for
  return $\big(f_\theta(y + \zeta_M) + \sigma\xi_M,\ \zeta_M\big)$ ▷ stochastic read-out step (4)
end function

Outer loop: q_θ-tempered particle filter ▷ tilted Feynman–Kac flow (8)
Initialise $h_0^{(s)} = h_0$, $w_0^{(s)} = 1/S$ for $s = 1, \ldots, S$ ▷ equally weighted initialisation
for $n = 0, \ldots, N-1$ **do**
  for $s = 1, \ldots, S$ **do**
    $h_{n+1}^{(s)} \leftarrow$ TRAJECTORYSAMPLER$(h_n^{(s)})$
    $\widetilde{w}_{n+1}^{(s)} \leftarrow w_n^{(s)} \exp\big(\beta \, q_\theta(h_{n+1}^{(s)})\big)$ ▷ weight-guiding step (8)
  end for
  $w_{n+1}^{(s)} \leftarrow \widetilde{w}_{n+1}^{(s)} / \sum_r \widetilde{w}_{n+1}^{(r)} \ \forall s = 1, \ldots, S$ ▷ normalise across cloud
  if $\mathrm{ESS}_{n+1}/S < \tau_{\mathrm{ESS}}$ **then**
    $h_{n+1}^{(s)} \leftarrow h_{n+1}^{(a_s)}$, $w_{n+1}^{(s)} \leftarrow 1/S$ ▷ systematic resampling; ancestors a_s
  end if
end for
return $\widehat{a}_N(x) := \mathrm{Decode}\big(y_N^{(s^\star)}\big)$, $s^\star \in \arg\max_s w_N^{(s)}$ ▷ highest-weight particle

Algorithm 1 gives the bootstrap particle approximation (9). TRAJECTORYSAMPLER stochastically generalises the inner TRM recursion (1)–(2); the outer Q-tempered particle filter generalises single-path rollout to a weighted cloud, with deterministic TRM as the zero-noise, single-particle path. Resampling triggers only when the effective sample size

$$\mathrm{ESS}_n := \Big(\sum_s (w_n^{(s)})^2\Big)^{-1} \tag{12}$$

falls below $\tau_{\mathrm{ESS}} S$, using the systematic protocol (Kitagawa, 1996; Carpenter et al., 1999): ancestor indices $a_{1:S}$ are drawn from the cumulative-weight CDF at stratified levels with a single shared uniform offset, replicating high-weight particles and dropping low-weight ones at the uniform new weight $1/S$. The terminal decoder returns the answer of the highest-weight particle, a particle-level MAP estimate.

Remark 1 (Cost and the deterministic limit)**.** The per-particle steps in Algorithm 1 are independent across s and can batch along the sample axis. Measured on one H100 GPU at $N = 48$, a puzzle costs 0.014 s at $S = 1$ and 0.058 s at $S = 16$, sixteen times the forward calls for roughly four times the wall-clock time, with peak memory growing linearly from 84 to 554 MB (Table 3). Stopping early when the particle cloud's largest guide confidence exceeds 0.95 matches full-depth accuracy at 14.5% of the $N = 48$ budget on Sudoku-Extreme Test and 35% on DF (Appendix K). With $S = 1$ and $\sigma = 0$, the algorithm reduces exactly to the deterministic TRM (Jolicoeur-Martineau, 2025).

5 Operating theory and diagnostics

The probabilistic formulation in §4 lets us characterise when the Q-head's re-ranking is aligned with successful trajectories. The theory yields two conditions for whether guided exploration helps, *tube stability* (§5.1) and *guide alignment* (§5.2), plus a third, output-side diagnostic, *token-marginal entropy* (§5.3), on the resulting particle cloud.

5.1 Local tube stability of stochastic trajectories

We analyse the local properties of the inner recursion (3). The resulting predictions are measured in Figure 2. Fix $x \in \mathbb{R}^{L\times D}$ and $h_n \in \mathcal{H}$, and suppress these, as well as subscripts n, from the notation. Let $z_{1:M}$ denote the deterministic inner trajectory and $\zeta_{1:M}$ the stochastic rollout of (3). Write $e_m := \zeta_m - z_m$ and

$$\tau_r := \inf\{m \in \{1,\dots,M\} : \|e_m\| > r\} \tag{13}$$

for the deviation at step m and the first-exit step from the r-tube around the deterministic path, respectively, with the convention $\tau_r = M + 1$ when the set is empty.

Proposition 1 (Tube stability)**.** Assume there exist $r > 0$ and $\rho_r \in [0, 1)$ such that $z \mapsto f_\theta(x+y_n+z)$ is ρ_r-Lipschitz on the r-tube $\bigcup_{m=1}^{M}\{z : \|z - z_m\| \le r\}$, and that the noise variables are independent, mean-zero, with $\mathbb{E}\|\xi_m\|^2 \le \nu^2$. Then for $m = 1,\dots,M$ the average in-tube deviations conform to the geometric decay bound

$$\mathbb{E}\big[\|e_m\|^2\mathbf{1}_{\{\tau_r>m\}}\big] \;\le\; \sigma^2\nu^2\sum_{j=0}^{m-1}\rho_r^{2j} \;\le\; \frac{\sigma^2\nu^2}{1-\rho_r^2}. \tag{14}$$

Corollary 1 (Finite-horizon tube-exit probability)**.** Under the hypotheses of Proposition 1 we have

$$\mathbb{P}(\tau_r \le M) \;\le\; \frac{M}{r^2}\frac{\sigma^2\nu^2}{1-\rho_r^2}. \tag{15}$$

Proposition 1 controls the deviation e_m while the rollout stays in the r-tube; Corollary 1 then quantifies how often that control applies. Together (14)–(15) identify a regime in which stochastic widening remains controlled. Fixed M, larger r, smaller σ, and stronger local contraction all tighten both bounds. The in-tube bound does not itself depend on M, and both bounds need only independent, mean-zero noise of finite second moment (Remarks 3 and 4 in Appendix G). We evaluate these bounds empirically in Figure 2, showing that the deviation bound is not exceeded and that the probability of remaining within the tube is high at moderate noise levels.

5.2 Alignment of the Q-head guide

Here we show that the BCE-trained Q-head is aligned with its training label in the integrated-covariance sense of (16) below. The family $\{\pi_{n+1,t}\}_{t\in[0,\beta]}$ in (8) is a one-parameter exponential tilt, such that bounded $\varphi : \mathcal{H} \to \mathbb{R}$ satisfies the tilt-derivative identity (Gelman and Meng, 1998)

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}_{\pi_{n+1,t}}[\varphi(h)] = \mathrm{Cov}_{\pi_{n+1,t}}\big(\varphi(h),\, q_\theta(h)\big). \tag{16}$$

We call a guide q_θ *aligned* with φ on $[0, \beta]$ if and only if $\mathrm{Cov}_{\pi_{n+1,t}}(\varphi, q_\theta) \ge 0$ throughout. Equivalently, $\mathbb{E}_{\pi_{n+1,t}}[\varphi(h)]$ is non-decreasing in t on the tilting path $[0, \beta]$. Integrating yields an improvement measure against an 'unguided' baseline. Appendix J measures the calibration and the discrimination of the guide by outer step on each benchmark.

Lemma 1 (BCE alignment)**.** *Let $Y \in \{0, 1\}$ and let $0 < \eta < 1$ be a random variable with $\mathbb{E}[Y \mid \eta] = \eta$. For any strictly increasing ψ, the guide $q_\theta = \psi(\eta)$ satisfies $\mathrm{Cov}_{\pi_{n+1,t}}(Y, q_\theta) \ge 0$ for all $t \in [0, \beta]$. In particular, for $q_\theta = \log\eta$, the population BCE optimum among guides on which η is the conditional expectation of Y, is aligned with Y.*

TRM's Q-head is trained under BCE, so Lemma 1 applies with $\eta = \mathrm{sigmoid}(\tilde{q}_\theta(y))$ whenever the head attains the population BCE optimum under (8). Note that training does not guarantee this: the optimum is a population statement over the training distribution of puzzles, whereas the covariance

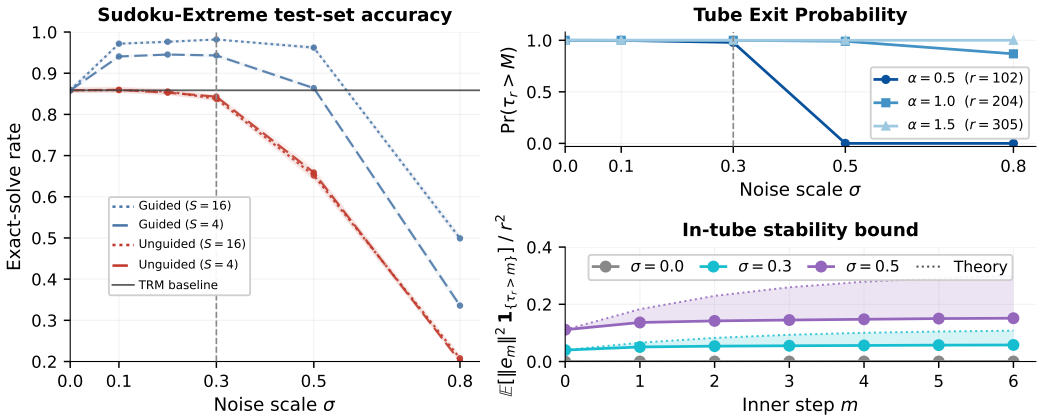

Figure 2: **Local stability and the noise sweet spot on Sudoku-Extreme.** *Left*: terminal exact-solve rate vs. σ for unguided ($\beta = 0$) and guided ($\beta = 0.25$) inference at $S \in \{4, 16\}$. *Right, top*: tube-survival probability $\Pr(\tau_r > M)$ vs. noise scale for radii $r = \alpha\sqrt{LD}$. *Right, bottom*: average in-tube deviations $\mathbb{E}[\|e_m\|^2 \mathbf{1}_{\{\tau_r > m\}}]/r^2$ per inner step at $\alpha = 1.5$; dotted curves give the bound (14) at $\hat{\rho}_r$, the empirical local Lipschitz constant on the r-tube estimated as the maximum secant ratio over 16 random unit-direction probes at each point z_m, $0 \le m \le M$. At $\sigma = 0.3$, accuracy is maximised and tube-containment remains high, with deviations within the theory bound.

in (16) is taken over the particle cloud of a single test instance. Lemma 1 therefore describes the guide at the BCE optimum rather than the trained head itself (cf. Appendix G.1), so we test alignment from inference traces, both with labels (post-hoc) and label-free; Maze-Hard (§6) is a case where the idealisation fails. For $H \sim \pi_{n+1,t}$, write $p_n(t) := \mathbb{P}_{\pi_{n+1,t}}(H \in A_n)$ for the probability mass on the success set $A_n \subseteq \mathcal{H}$ under the tilted law. We then introduce two scalar quantities derived from (16) that test alignment at outer step n (Figure 3).

Class-conditional gap. Here we define a post-hoc diagnostic to measure (16) in the binary-outcome case; see Appendix F for derivation and discussion. For $\varphi = \mathbf{1}_{A_n}$, the covariance (16) factorises as $p_n(t)\big(1 - p_n(t)\big)\,\Delta_n(t)$ with

$$\Delta_n(t) := \mathbb{E}_{\pi_{n+1,t}}[q_\theta \mid A_n] - \mathbb{E}_{\pi_{n+1,t}}[q_\theta \mid A_n^c]. \tag{17}$$

Alignment on $[0, \beta]$ is therefore equivalent to the scalar condition $\Delta_n(t) \ge 0$ for all $t \in [0, \beta]$. The sample diagnostic $\hat{\Delta}_n$ requires success labels and is therefore a post-hoc score. Figure 3 tracks (17) on a particle cloud and shows the predicted mass shift emerging as soon as the Q-head separates successful from unsuccessful trajectories. Intuitively, $\hat{\Delta}_n$ is the gap between the mean guide score q_θ on solving particles and on failing ones. It accounts for the 'space' the guide has to discriminate.

Guide spread over the cloud. A label-free necessary condition for a non-trivial tempering improvement can be observed in the cloud directly. Applying Cauchy–Schwarz to (16) with $\mathrm{std}(\mathbf{1}_{A_n}) \le \frac{1}{2}$ yields the integrated bound

$$\big|\, p_n(\beta) - p_n(0) \,\big| \;\le\; \tfrac{\beta}{2} \sup_{t \in [0,\beta]} \widehat{\mathrm{std}}_s\big[q_\theta(h_{n+1}^{(s)})\big], \tag{18}$$

where $\widehat{\mathrm{std}}_s$ denotes the empirical standard deviation over the particle approximation of $\pi_{n+1,t}$, standing in for the population quantity $\mathrm{std}_{\pi_{n+1,t}}[q_\theta]$ of the exact bound, and the supremum runs over the tilting path $t \in [0, \beta]$. A non-negligible $\widehat{\mathrm{std}}_s$ is therefore necessary (but not sufficient) for a meaningful improvement at outer step n. The spread bounds the rate of mass shift $|p_n(\beta) - p_n(0)|$ in either direction, so a small spread can rule out gains a priori.

In Figure 3 (bottom), the Maze-Hard Q-spread satisfies $\sup_{t \le \beta} \widehat{\mathrm{std}}_s[q_\theta] \le 0.04$. By (18), this caps the achievable mass shift $|\Delta p_n|$ at 2% for $\beta = 1$ and 0.5% for $\beta = 0.25$, well below the 5% oracle headroom on Maze-Hard. On Sudoku-Extreme, by contrast, the spread is large enough that (18) permits at least a 12.5% mass shift at $\beta = 0.25$, consistent with the observed improvement (Figure 1).

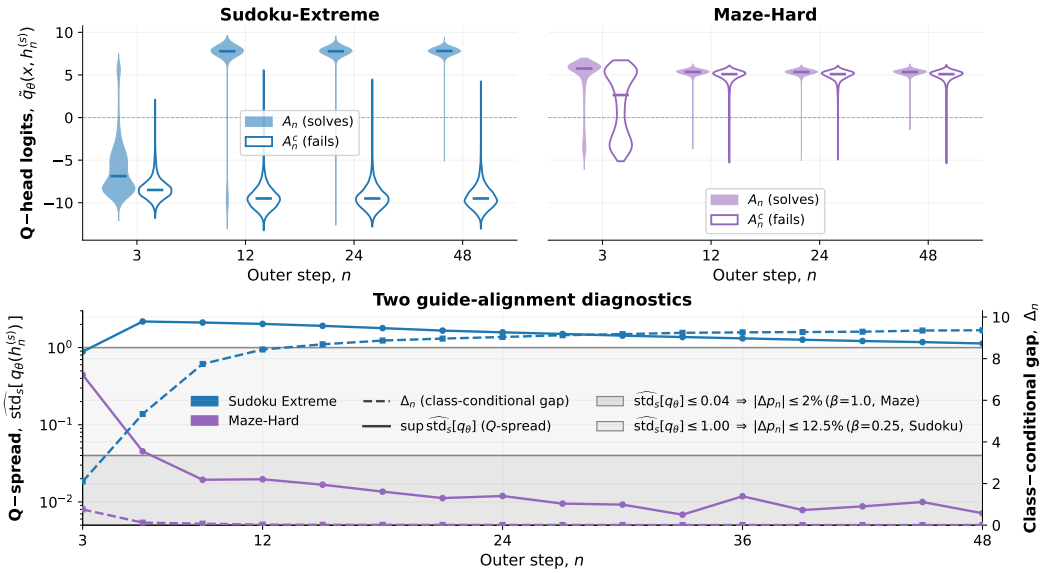

Figure 3: **Guide-alignment diagnostics.** *Top*: the raw $\tilde{q}_\theta$ logits split by puzzle-solve outcome over outer step n; strong class separation on Sudoku-Extreme (*left*), failure to discriminate on Maze-Hard (*right*). *Bottom*: the post-hoc class-conditional gap $\hat{\Delta}_n$ (17) (right-axis, dotted) and the Q-spread bound (18) (left-axis, solid) for the mass shift $|p_n(\beta) - p_n(0)|$, both in the guide-score units $q_\theta = \log \mathrm{sigmoid}(\tilde{q}_\theta)$ on which the tilt acts. On Maze-Hard the bound is prohibitive: $\le 2\%$ at $\beta = 1$, tightening to $\le 0.5\%$ at $\beta = 0.25$; on Sudoku-Extreme the bound exceeds 12.5% at $\beta = 0.25$, in line with the observed improvement (Figure 1).

5.3 Token entropy as an uncertainty diagnostic

Modelling a particle distribution (6) also yields in-cloud uncertainty diagnostics at the token level. Let $\mathrm{Dec} : \mathcal{H} \to \{1, \dots, C\}^L$ be the deterministic decoder. For each token position ℓ, the cloud induces a class marginal $p_\ell(\beta; n) := \big(p_{\ell,1}(\beta; n), \dots, p_{\ell,C}(\beta; n)\big)$ with $p_{\ell,k}(\beta; n) := \mathbb{P}_{\pi_{n+1,\beta}}(\mathrm{Dec}(h)_\ell = k)$, read off the particle weights at no extra cost. We summarise its dispersion by the normalised Shannon entropy, both at the terminal step and aggregated over outer steps $n_1, \dots, n_J$:

$$\text{Terminal:} \quad H_\ell := \frac{1}{\log C} H\big(p_\ell(\beta; N)\big), \tag{19}$$

$$\text{Aggregated:} \quad \bar{H}_\ell := \frac{1}{J \log C} \sum_{j=1}^{J} H\big(p_\ell(\beta; n_j)\big). \tag{20}$$

We treat H_ℓ and $\bar{H}_\ell$ as risk-ranking diagnostics over token-level errors; empirical AUROCs and selective-prediction curves are in Figure 4.

6 Experimental evaluation

We evaluate our guided reasoning method, as described in Algorithm 1, on two fixed TRM models trained on Sudoku-Extreme and Maze-Hard, respectively, and on a third that we train for PPBench, a pencil-puzzle benchmark (Waugh, 2026). Particle count, horizon and resampling threshold are validation-tuned on Sudoku-Extreme and transferred to Maze-Hard; the Maze-Hard noise scale and inverse temperature are fixed a priori without Maze-Hard data, with the diagnostics of §5 acting as a verifier; reported numbers are single-shot test-set evaluations. See Appendix A for further details.

The diagnostics of §5 predict, before any evaluation, where the procedure should help. The inner-deviation bound (14)–(15) is conformed to on both benchmarks at moderate noise (Figure 2). The Q-spread bound (18) flags the Maze-Hard Q-head as too flat to produce a non-trivial tempering

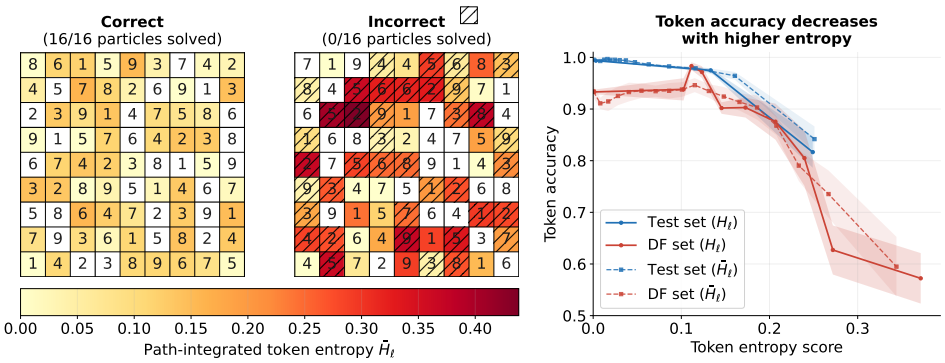

Figure 4: **Token entropy as a risk-ranking diagnostic on Sudoku-Extreme.** *Left, middle*: solved and unsolved test puzzles coloured by per-cell path-integrated entropy $\bar{H}_\ell$ (shaded: incorrect tokens; white: given cells). *Right*: binned token accuracy vs. uncertainty on Test and DF for H_ℓ and $\bar{H}_\ell$; bands are 95% CIs. Token-level AUROC for H_ℓ: Test 0.834 ± 0.011, DF 0.710 ± 0.014.

improvement, while Sudoku-Extreme clears both the Q-spread and class-conditional thresholds (Figure 3). The diagnostics therefore only predict non-improvement on Maze-Hard.

Table 1 confirms both predictions. On Sudoku-Extreme, guided MAP closes nearly the entire oracle gap (85.9% $\rightarrow$ 98.0%), recovering 85.9% on the deterministic-failure split where the baseline scores zero. On Maze-Hard, stochastic exploration exposes nearly five points of oracle-particle headroom (91.3% vs. the deterministic 86.6%), but the misaligned Q-head does not concentrate mass on successful trajectories, so guided MAP does not improve on a single rollout (85.6% against 85.2%). Token-marginal entropy ranks misclassification risk on Sudoku-Extreme (Figure 4): abstaining on the upper-decile H_ℓ raises retained exact-solve rate to 100.0 ± 0.0% on Test and 99.7 ± 0.3% on DF.

Table 1: **Terminal exact-solve rates at matched budgets.** Below, $S = 16$, $N = 48$. For every stochastic evaluation there are 5,376 backbone calls per Sudoku-Extreme puzzle against 336 for the deterministic TRM. Core runs are evaluated on a 5 seeds $\times$ 5 folds design (the deterministic TRM varies over folds only). Particle cloud statistics re-read the stored unguided ($\beta = 0$) particles generated on the full splits (5 folds of one seed). Puzzle counts for every evaluation are in Table 2; the rules, and paired same-run differences, in Appendix H. PPBench evaluates on 5 seeds on the 253-puzzle validation split (Appendix I). Last column: the DF clouds rescored after i.i.d. Gaussian noise of s.d. 16 on every raw guide logit (Appendix H). Values are % solve rates.

Method	Sudoku Test	Sudoku DF	Maze Test	PPBench val	DF, corrupted guide
Core runs					
Deterministic TRM	85.9±0.6	0.0	86.6±2.1	62.5	–
Single rollout ($\beta = 0$)	84.3±0.7	35.7±2.8	85.2±2.5	63.6±1.6	–
Sequential guiding (ours)	98.0±0.2	85.9±1.5	85.6±2.8	77.9±0.6	–
Particle cloud statistics					
Plurality vote	96.8±0.7	79.5±1.2	86.0±4.1	67.7±1.4	–
Terminal-Q (PTRM's rule)	98.1±0.5	88.4±0.6	86.9±4.0	79.5±0.8	65.3±0.9
Path-sum tilt (no resampling)	98.1±0.5	88.1±0.9	85.7±3.8	78.7±0.7	83.9±1.6
Oracle (any correct particle)	98.1±0.5	88.4±0.6	91.3±2.7	82.8±0.7	–

Selection rules at matched compute. The stochastic rows of Table 1 spend the same backbone calls and differ only in how selection is performed in the particle cloud. Three findings follow. First, the gain is the guide, not just the sampling. On PPBench (Waugh, 2026), where we train a TRM on four puzzle types and transfer the Sudoku-Extreme configuration without tuning (Appendix I), the plurality vote recovers 4 of the 19 points between a single rollout and the oracle, and each guide-consistent selection rule recovers 14 to 16, on every type. Second, with a near-perfect guide the guided rules tie. On Sudoku-Extreme the terminal guide score separates correct from incorrect particles with AUROC 1 (Appendix J), and the best-of-K selection rule of PTRM (Sghaier et al., 2026), the path-sum tilt and sequential guiding agree to within 0.2 points on the Test split (Appendix H). Third, the selection

rules part when the guide degrades. With Gaussian noise of s.d. 16 on every raw logit, the path-sum tilt holds 83.9% on DF while terminal selection falls to 65.3%: the sum over 48 steps averages noise that one terminal read-out cannot. Depth is no substitute for sampling (Appendix K).

Limitations and scope. Empirical evaluation covers Sudoku-Extreme, Maze-Hard and PPBench, all structured puzzles solved by a backbone with a trained halting head, which is the regime the guide requires; PPBench needed such a backbone trained first. Language tasks are not evaluated, and ARC-AGI (ARC Prize Foundation, 2025), a common LLM and TRM-comparison benchmark, remains a clear next evaluation step despite requiring a far higher compute budget. The framework is inference-only over a fixed backbone with fixed outer horizon N; coupling with continued training, model surgery, or non-recursive backbones is left to future work, and weight-updating test-time adaptation (McGovern, 2025) is complementary rather than compared. Our theory assumes a local-Lipschitz condition of f_θ on the relevant r-tube (Proposition 1) and a Q-head aligned with success in the sense of Lemma 1, which BCE training motivates but does not guarantee; the Maze-Hard misalignment shows when these conditions bind, and transfer to other recursive architectures remains open. The label-free spread bound (18) is necessary but not sufficient: a guide that clears the threshold could still fail in ways spread alone cannot detect. Our Q-head was trained for early stopping rather than trajectory ranking; a purpose-trained guide is the natural follow-up (§7).

7 Discussion and broader impact

The original motivation for this work was to explore what an approximate Bayesian interpretation of recursive reasoning (Jolicoeur-Martineau, 2025) might reveal about the elusive mechanics of the architecture. The investigation produced a Feynman–Kac-inspired theory that elevates the Q-head from afterthought to central guide for the selection of latent reasoning trajectories. The same machinery characterises both when recursive reasoning has trajectory-level headroom and when the model's internal guide can recover it, a thesis we believe extends to recursive reasoning more broadly.

What surprised us was the immediate success of both ingredients on Sudoku-Extreme: modest stochastic exploration produced successful trajectories, and the Q-head correctly weighted particles towards these (Figure 1). Maze-Hard was a step into the unknown as no prior work reports inference-time improvements there (cf. McGovern, 2025). Stochastic exploration again exposed oracle-level headroom, but, as the theory predicted, the guide could not identify it. The difference in character between these two benchmarks comes down to the guide. Improvement is "in there", but the interesting question is: "how do we build the guide that finds it?"

We therefore propose three directions, each expanding the role of the guide. (i) the guide as a design object, with three levels of intervention when the diagnostics fail: recalibrating a guide whose ranking survives, refitting a guide whose discrimination has failed, as on Maze-Hard (Appendix J), and, where neither suffices, bespoke guiding, a guide built for the task to handle structural constraints, custom training objectives or multi-guide compositions rather than inherited from the Q-head; (ii) train-time ramifications of the theory (Appendix E) and inference-time efficiency, from across-particle parallelism (Remark 1) to scaling beyond Figure 6; and (iii) broader evaluation beyond pencil puzzles, from graph-based (Palm et al., 2018; Sinha et al., 2019) and rule-governed language tasks (Clark et al., 2020; Tafjord et al., 2021) to ARC-AGI (ARC Prize Foundation, 2025).

More broadly, we propose generalisations of the framework beyond recursive reasoning. In AI safety, steering has emerged as a promising approach for online adaptation of foundation-model behaviour (Turner et al., 2023; Rimsky et al., 2024); a guided variant brings an explicit mechanism for control, in which a bespoke guide identifies sampled trajectories aligned with pre-trained constraints rather than rotating activation vectors directly.

A natural next question is the transfer-learning capacity of small recursive models. If a guide q_A is used up to step n to produce (y_n, z_n), can z_n seed another task, or can a second guide q_B redirect the same state towards a related solution? This question is especially natural for graph-based reasoning (Palm et al., 2018; Sinha et al., 2019) and rule-governed language tasks (Clark et al., 2020; Tafjord et al., 2021), where success is determined by consistent relational or logical constraints. If such transfer is proven possible, tiny recursive models would become more than compact solvers: they would be reusable reasoning devices, generalist in capability and specialised online with a guide.

Acknowledgements and disclosure of funding

This work was funded by digiLab. The authors declare no competing interests.

References

U. Abubacar, R. Bauer, and D. Kanojia. Parameter-efficient quality estimation via frozen recursive models, 2026. URL https://arxiv.org/abs/2603.14593. Accepted to LowResLM Workshop at EACL 2026.

ARC Prize Foundation. ARC-AGI-2. https://arcprize.org/arc-agi/2/, 2025. URL https://arcprize.org/arc-agi/2/. Dataset and details; code/data repo: https://github.com/arcprize/ARC-AGI-2.

A. Asadulaev, R. Banerjee, F. Karray, and M. Takáč. Latent reasoning in TRMs is secretly a policy improvement operator. In Proceedings of the 43rd International Conference on Machine Learning, volume 306 of Proceedings of Machine Learning Research, pages 4035–4047. PMLR, 2026. URL https://proceedings.mlr.press/v306/asadulaev26a.html. arXiv:2511.16886.

S. Bae, Y. Kim, R. Bayat, S. Kim, J. Ha, T. Schuster, A. Fisch, H. Harutyunyan, Z. Ji, A. Courville, and S. Yun. Mixture-of-recursions: Learning dynamic recursive depths for adaptive token-level computation. In Advances in Neural Information Processing Systems (NeurIPS 2025), 2025. URL https://openreview.net/forum?id=QuqsEIVWIG.

S. Bai, J. Z. Kolter, and V. Koltun. Deep equilibrium models. In Advances in Neural Information Processing Systems, 2019.

B. Behzadian, B. Daley, G. Subbaraj, and H. Nassif. Unrolled policy iteration for tiny recursive models. ICLR 2026 Workshop on AI with Recursive Self-Improvement (poster), OpenReview, 2026. URL https://openreview.net/forum?id=rIzREKws05.

P. G. Bissiri, C. C. Holmes, and S. G. Walker. A general framework for updating belief distributions. Journal of the Royal Statistical Society: Series B, 78(5):1103–1130, 2016. doi: 10.1111/rssb.12158.

J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. IEE Proceedings - Radar, Sonar and Navigation, 146(1):2–7, 1999. doi: 10.1049/ip-rsn:19990255.

P. Clark, O. Tafjord, and K. Richardson. Transformers as soft reasoners over language. In Christian Bessiere, editor, Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, pages 3882–3890. International Joint Conferences on Artificial Intelligence Organization, July 2020. doi: 10.24963/ijcai.2020/537. URL https://doi.org/10.24963/ijcai.2020/537. Main track.

P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. Journal of the Royal Statistical Society Series B: Statistical Methodology, 68(3):411–436, 2006. doi: 10.1111/j.1467-9868.2006.00553.x.

J. V. Dillon. Speed is confidence, 2026. URL https://arxiv.org/abs/2601.19085.

A. Efstathiou and A. Balwani. Recursive reasoning as attractor landscape search: Mechanistic dynamics of the tiny recursive model. In ICLR 2026 Workshop on Latent & Implicit Thinking: Going Beyond CoT Reasoning, 2026. URL https://openreview.net/forum?id=kKps9W1K7n.

S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal. Detecting hallucinations in large language models using semantic entropy. Nature, 630(8017):625–630, 2024. doi: 10.1038/s41586-024-07421-0.

A. Gelman and X.-L. Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. Statistical Science, 13(2):163–185, 1998. doi: 10.1214/ss/1028905934.

W. R. Gilks and C. Berzuini. Following a moving target — Monte Carlo inference for dynamic Bayesian models. Journal of the Royal Statistical Society: Series B, 63(1):127–146, 2001. doi: 10.1111/1467-9868.00280.

A. Graves. Adaptive computation time for recurrent neural networks. arXiv preprint arXiv:1603.08983, 2016.

A. Jolicoeur-Martineau. Less is more: Recursive reasoning with tiny networks. arXiv preprint arXiv:2510.04871, 2025. doi: 10.48550/arXiv.2510.04871.

G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. Journal of Computational and Graphical Statistics, 5(1):1–25, 1996. doi: 10.1080/10618600.1996.10474692.

O. Koch. olivkoch/nano-trm: An implementation of tiny recursive models (TRM). GitHub, 2025. URL https://github.com/olivkoch/nano-trm.

Y. Koishekenov, A. Lipani, and N. Cancedda. Encode, think, decode: Scaling test-time reasoning with recursive latent thoughts. arXiv preprint arXiv:2510.07358, 2025. doi: 10.48550/arXiv.2510.07358. URL https://arxiv.org/abs/2510.07358.

L. Kuhn, Y. Gal, and S. Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In International Conference on Learning Representations (ICLR), 2023.

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

J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. Journal of the American Statistical Association, 93(443):1032–1044, 1998. doi: 10.1080/01621459.1998.10473765.

R. McGovern. Test-time adaptation of tiny recursive models. arXiv preprint arXiv:2511.02886, 2025. doi: 10.48550/arXiv.2511.02886. URL https://arxiv.org/abs/2511.02886.

R. Palm, U. Paquet, and O. Winther. Recurrent relational networks. In Advances in Neural Information Processing Systems (NeurIPS), 2018. URL https://papers.nips.cc/paper/2018/hash/b9f94c77652c9a76fc8a442748cd54bd-Abstract.html.

N. Rimsky, N. Gabrieli, J. Schulz, M. Tong, E. Hubinger, and A. Turner. Steering Llama 2 via contrastive activation addition. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

A. Roye-Azar, S. Vargas-Naranjo, D. Ghai, N. Balamurugan, and R. Amir. Tiny recursive models on ARC-AGI-1: Inductive biases, identity conditioning, and test-time compute. arXiv preprint arXiv:2512.11847, 2025. doi: 10.48550/arXiv.2512.11847. URL https://arxiv.org/abs/2512.11847.

A. Sghaier, A. Parviz, and A. Jolicoeur-Martineau. Probabilistic tiny recursive model. arXiv preprint arXiv:2605.19943, 2026. doi: 10.48550/arXiv.2605.19943. URL https://arxiv.org/abs/2605.19943.

K. Sinha, S. Sodhani, J. Dong, J. Pineau, and W. L. Hamilton. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4506–4515, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1458. URL https://aclanthology.org/D19-1458/.

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

O. Tafjord, B. Dalvi, and P. Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3621–3634, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.317. URL https://aclanthology.org/2021.findings-acl.317/.

A. M. Turner, L. Thiergart, G. Leech, D. Udell, J. J. Vazquez, U. Mini, and M. MacDiarmid. Steering language models with activation engineering, 2023. URL https://arxiv.org/abs/2308.10248.

G. Wang, J. Li, Y. Sun, X. Chen, C. Liu, Y. Wu, M. Lu, S. Song, and Y. Abbasi Yadkori. Hierarchical reasoning model, 2025. URL https://arxiv.org/abs/2506.21734.

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

J. Waugh. Pencil puzzle bench: A benchmark for multi-step verifiable reasoning. arXiv preprint arXiv:2603.02119, 2026. doi: 10.48550/arXiv.2603.02119. URL https://arxiv.org/abs/2603.02119.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Advances in Neural Information Processing Systems, volume 35, 2022.

S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In Advances in Neural Information Processing Systems, volume 36, 2023.

S. Zhao, R. Brekelmans, A. Makhzani, and R. Grosse. Probabilistic inference in language models via twisted sequential Monte Carlo. In Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pages 60704–60748. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhao24c.html.

R.-J. Zhu, Z. Wang, K. Hua, T. Zhang, Z. Li, H. Que, B. Wei, Z. Wen, F. Yin, H. Xing, L. Li, J. Shi, K. Ma, S. Li, T. Kergan, A. Smith, X. Qu, M. Hui, B. Wu, Q. Min, H. Huang, X. Zhou, W. Ye, J. Liu, J. Yang, Y. Shi, C. Lin, E. Zhao, T. Cai, G. Zhang, W. Huang, Y. Bengio, and J. Eshraghian. Scaling latent reasoning via looped language models. arXiv preprint arXiv:2510.25741, 2025. doi: 10.48550/arXiv.2510.25741. URL https://arxiv.org/abs/2510.25741.

A Evaluation protocol

Models and data. We consider two frozen TRM models trained with the Nano-TRM framework (Koch, 2025) on Sudoku-Extreme and Maze-Hard (Wang et al., 2025), using Nano-TRM's default training hyperparameters, and a third that we train on PPBench (Appendix I). Following the data pipeline released with Wang et al. (2025), we apply $1{,}000$ shuffling augmentations to Sudoku-Extreme training data and no augmentation to Maze-Hard. The model, dataset, and outer horizon $N = 48$ are held fixed throughout, matching Jolicoeur-Martineau (2025), whose 16 supervision steps of 3 deep recursions give the same 48 outer steps, so only the inference procedure (§4) varies. Figure 5 (§B) shows the contrasting training trajectories for the two Q-heads.

Splits and statistics. For Sudoku-Extreme, we hold out a 500-puzzle validation subset for hyperparameter selection, disjoint from both the $10{,}020$-puzzle Test split and the $1{,}427$-puzzle deterministic-failure (DF) split (DF puzzles are those on which the deterministic TRM baseline fails). The seed-fold runs of Table 1 evaluate $9{,}984$ Test and $1{,}408$ DF puzzles; the full-set trace analyses in the appendices evaluate every puzzle of both splits. For Maze-Hard, the $1{,}000$-puzzle Test split (of which the seed-fold runs evaluate 992 puzzles and the trace analyses all $1{,}000$) is too small to subdivide; no validation subset is held out, and hyperparameters are transferred from the Sudoku-Extreme sweep (discussion below). Headline experiments are run over 5 data folds (disjoint partition of Test, each fold 20% of Test; see Table 4) $\times\, 5$ random seeds (initialisation of stochastic variables), giving 25 independent runs; reported $\pm$ values are seed-fold standard deviations. Hyperparameter ablations on Sudoku-Extreme (Figure 6) use a $2{,}000$-puzzle subset of Test for compute reasons.

Puzzles evaluated and call budgets. Table 2 lists, for each benchmark split, how many puzzles each procedure evaluated and its backbone calls per puzzle. The core runs evaluated slightly fewer puzzles than the splits contain; the stored-trace analyses of Appendices H to K evaluate every puzzle. The PPBench core runs are 5 seeds of one fold.

Table 2: **Puzzles evaluated and per-puzzle call budgets.** Backbone calls are $S \times N \times (M+1)$ for the particle filter and $N \times (M+1)$ for the deterministic TRM, both at $N = 48$.

Benchmark	Split	In split	Core runs	Stored-trace analyses	Filter calls	Deterministic calls
Sudoku-Extreme	Test	10,020	9,984	10,020	5,376	336
Sudoku-Extreme	DF	1,427	1,408	1,427	5,376	336
Maze-Hard	Test	1,000	992	1,000	3,840	240
Maze-Hard	DF	134	128	134	3,840	240
PPBench	Validation	253	253	253	3,840	240
PPBench	Golden	42	42	42	3,840	240

Compute resources. All training and inference runs use a single NVIDIA H100 GPU. The full set of reported experiments (headline 5×5 seed-fold sweeps on Sudoku-Extreme and Maze-Hard, hyperparameter ablations, the checkpoint scan, and figure-data generation) consumes approximately 20–30 GPU-hours in total. Preliminary or discarded runs that did not make it into the final paper are not included in this estimate but are of comparable order. The PPBench backbone training, its five-seed inference runs and the stored-trace analyses of Appendices H to K were run afterwards on the same hardware.

Implementation and cost. Weights are carried in log space and normalised by log-sum-exp; the ESS and the systematic resampling are computed from the normalised weights. Table 3 reports batched throughput and peak memory on one NVIDIA H100 at $N = 48$: 64 puzzles in four batches of 16 per setting, timed end to end, so the figures are amortised seconds per puzzle rather than single-query latency. Memory grows linearly in S (the state is $2LD$ floats per particle); wall-clock is nearly flat to $S = 8$ and roughly four times the single-particle cost at $S = 16$.

Table 3: **Measured inference cost against particle count** ($N = 48$, one NVIDIA H100, 64 puzzles per setting).

S	1	2	4	8	16
Seconds per puzzle	0.014	0.013	0.018	0.031	0.058
Peak GPU memory (MB)	84	111	185	307	554

Table 4: **Hyperparameters used for main results.** Common across all three benchmarks: $S = 16$, $N = 48$; $\tau_{\mathrm{ESS}} = 0.3$ on Sudoku-Extreme and Maze-Hard and 0.5 on PPBench, fixed a priori. The noise scale σ, inverse temperature β and trained inner depth M vary by benchmark; selection methodology in 'Hyperparameter selection', §A.

Model	σ	β	M	Splits	Design	Reported as
Sudoku-Extreme	0.3	0.25	6	Test (9,984), DF (1,408)	5×5 seed-fold	$\pm$ seed–fold s.d.
Maze-Hard	0.5	1	4	Test (992)	5×5 seed-fold	$\pm$ seed–fold s.d.
PPBench	0.3	0.25	4	Val (253), Golden (42)	5 seeds, one fold	$\pm$ seed s.d.

Hyperparameter selection. Three hyperparameters are common to Sudoku-Extreme and Maze-Hard: $S = 16$ particles, $N = 48$ outer steps, $\tau_{\mathrm{ESS}} = 0.3$; PPBench reuses S, N, σ and β from the Sudoku-Extreme sweep with $\tau_{\mathrm{ESS}} = 0.5$ fixed a priori (Appendix I). Only the noise scale σ and inverse temperature β vary by benchmark (Table 4); the inner depth M is the checkpoint's trained value (6 and 4) and is not a free parameter. For Sudoku-Extreme, we sweep $\mathfrak{H} := (\sigma, \beta, S, \tau_{\mathrm{ESS}})$ on the validation subset and select $(\sigma, \beta) = (0.3, 0.25)$; reported Test and DF numbers are then a single-shot evaluation of this configuration. For Maze-Hard, no validation set is available, and we

deliberately do not tune on Maze-Hard data: S, N, τ_{ESS} are transferred from the Sudoku-Extreme sweep, $\sigma = 0.5$ follows the architectural prior (Maze-Hard's normalisation scale is larger than Sudoku-Extreme's), and $\beta = 1$ is a canonical default for non-trivial tempering, rather than the more conservative $\beta = 0.25$ used for Sudoku-Extreme. The role of Maze-Hard here is to test whether the diagnostics of §5 correctly predict failure on a configuration chosen without Maze-Hard data, which they do without observing ground truth data (Appendix B).

Ablation scope. The hyperparameter sweeps in §C are performed on the held-out Test sets and serve as post-tuning robustness checks for the validated (Sudoku-Extreme) and diagnostic-selected (Maze-Hard) configurations. Consistent with the diagnostic prediction, the Maze-Hard Q-head induces a constant response to β (Figure 6), confirming that no β sweep recovers the headroom.

B Maze-Hard: further analysis

Figure 5 places the Maze-Hard tube-stability evidence, the analogue of Figure 2 for Sudoku-Extreme, alongside the Q-head training diagnostics for both models. Reading the *left* panel shows that, much like Figure 2, the stochastic trajectories of the Maze-Hard model are stable in the sense of Proposition 1 for $\alpha \geq 0.5$ and $\sigma \leq 0.3$; the range of stable σ values conforms to the in-tube stability bound (14). However, the *right* panel shows what we believe is a justification for the poor performance of the in-built guide, observed in Figure 3. We display the training curves for the Q-head on both models, showing extremely high accuracy in the Sudoku-Extreme case, and poor, highly unstable convergence on Maze-Hard. We conjecture this is in part due to the sensitivity of the Maze-Hard structure, in that minor deviations from the solutions lead to poor results, given the objective is to find the 'shortest' path to the goal, not just an admissible one; this bimodal objective is not well supported by the BCE objective. A constructive use of the diagnostics for hyperparameter selection (rather than verification) is a natural follow-up; in this paper they serve only as a label-free verifier.

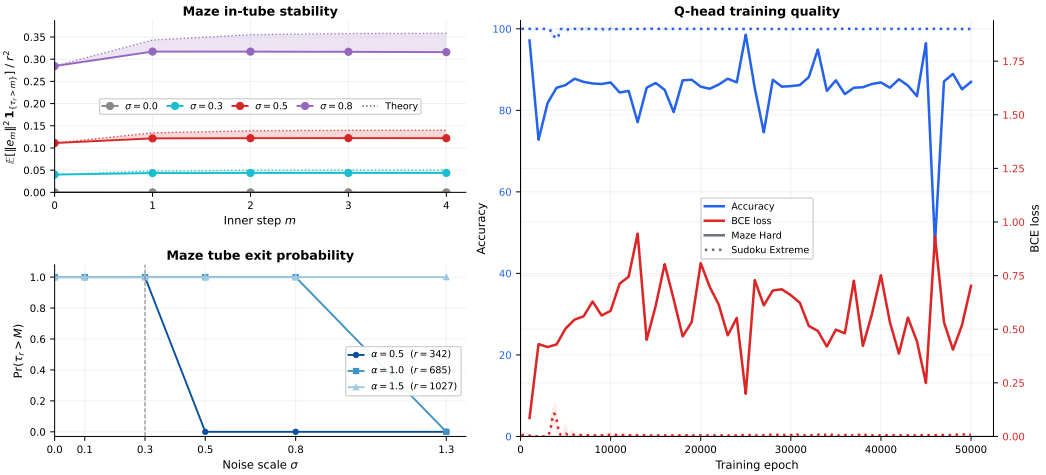

Figure 5: **Maze-Hard case study.** *Top, left*: In-tube deviations $\mathbb{E}[\|e_m\|^2 \mathbf{1}_{\{\tau_r > m\}}]/r^2$ per inner step m at $\alpha = 1.5$, with upper bounds (dotted) from the empirical local-Lipschitz estimate $\hat{\rho}_r$ defined in Figure 2. *Bottom, left*: Tube-survival probability $\Pr(\tau_r > M)$ vs. σ for $r = \alpha\sqrt{LD}$; stable for $\alpha \geq 0.5$, $\sigma \leq 0.3$, matching (14). *Right*: Q-head halting accuracy and BCE loss on the training framework's evaluation split, as logged during training. Sudoku-Extreme saturates near zero BCE and 100% accuracy, while Maze-Hard fails to converge.

C Hyperparameter ablation analysis

We report hyperparameter sweeps that complement the headline results of §6; Figure 6 summarises the observations across both benchmarks.

Sudoku-Extreme (top). The β sweep shows that even small amounts of Q-guidance recover the oracle particle, consistent with the high training-time accuracy of the Q-head (Figure 5, *right*). Once the oracle is recovered, larger β has no further upside; resampling at high β only distorts the cloud weighting. Terminal accuracy increases with the particle count S (*middle*). This is encouraging for scaling, and motivates us to consider adaptive particle-count schemes as a follow-up.

Maze-Hard (bottom). The sweeps tell quite a clear story: the guide is ineffective at identifying successful trajectories in any regime. The oracle gap is consistently visible, i.e. the cloud contains successful trajectories for every setting, but none of β, S, or τ_{ESS} has any effect on MAP performance. It is noteworthy that the normalised effective sample size holds at $\text{ESS}_n/S \approx 1$ throughout the sweeps (meaning weights stay uniform; no reweighting takes place) so the collapse is a property of the trained Q-head, not of the inference procedure. This makes sense given the near-zero variance in q_θ values (Figure 3, *bottom*).

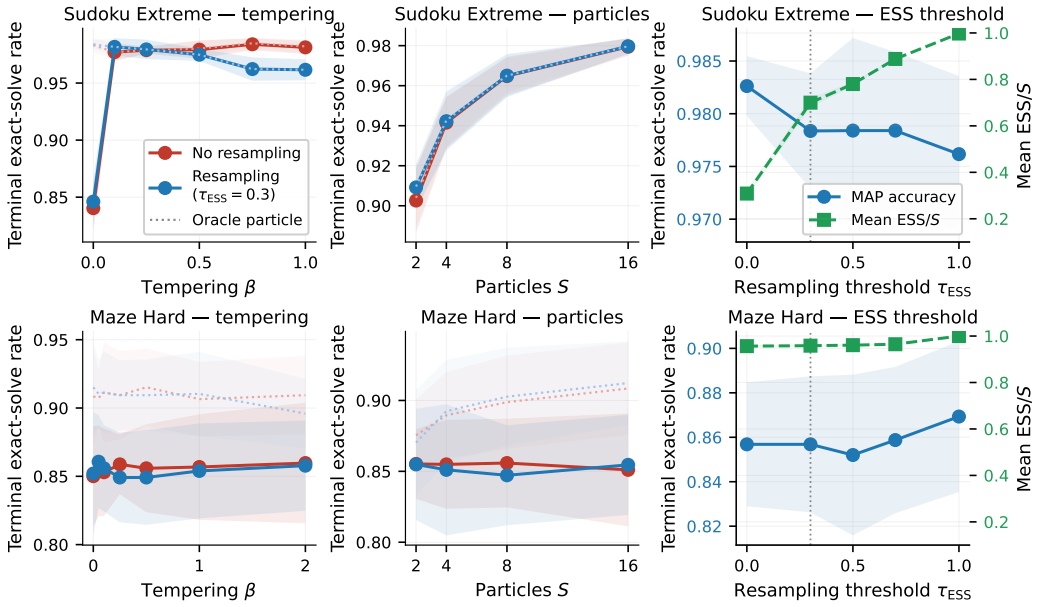

Figure 6: **Hyperparameter sensitivity sweeps.** We report sweeps for Sudoku-Extreme (*top*) and Maze-Hard (*bottom*). *Left*: we vary β across scales, with and without adaptive resampling. *Middle*: we vary the number of particles in the cloud, $S \in \{2, 4, 8, 16\}$. *Right*: we vary the resampling threshold τ_{ESS}. All other hyperparameters are fixed to the values in Table 4.

Inner depth and noise family. Two further sweeps, each on 5 folds of one seed, test the two choices that the main results hold fixed. Table 5 varies the inner depth M at inference on a 1,008-puzzle Sudoku-Extreme Test subset with $S = 16$ and $N = 48$: cutting the trained depth to $M = 2$ costs 17 points and the oracle falls with it, so the loss is the backbone's rather than the guide's, while $M \geq 4$ is flat within a fold s.d. We therefore keep the trained M (6 on Sudoku-Extreme, 4 on Maze-Hard and PPBench) and control exploration through σ. Table 6 replaces the Gaussian proposal noise by Laplace or Student-t noise, each scaled to unit variance per coordinate so that ν^2 in Proposition 1 is unchanged (Remark 3). Accuracy and oracle move by less than a fold s.d. on both benchmarks; the guide spread is unchanged on Sudoku-Extreme and rises from 0.23 to 0.33 under Student-t noise on Maze-Hard, still five times below the Sudoku-Extreme value, so the Maze-Hard guide stays flat under every family.

D Trajectory contraction in token entropy

To understand the capacity for calibrated uncertainty quantification a little further, we ask the question: 'how does accumulated entropy measure up against final-prediction entropy?' The answer, we propose, sheds light on the evolution of uncertainty along the reasoning trajectories.

Table 5: **Inner depth at inference** on a 1,008-puzzle Sudoku-Extreme Test subset ($S = 16$, $N = 48$; exact-solve %, mean $\pm$ s.d. over 5 folds of one seed). The trained depth is $M = 6$; its full-split values are in Table 1.

	$M = 2$	$M = 4$	$M = 8$	$M = 12$
Sequential guiding (ours)	81.1 ± 3.5	97.9 ± 1.3	97.7 ± 0.8	98.4 ± 0.7
Oracle (guided cloud)	84.0 ± 1.9	98.0 ± 1.1	97.7 ± 0.8	98.4 ± 0.7

Table 6: **Proposal noise family** (exact-solve %, mean $\pm$ s.d. over 5 folds of one seed; Sudoku-Extreme on a 1,008-puzzle Test subset, Maze-Hard on the full 1,000-puzzle Test split). Laplace noise has scale $1/\sqrt{2}$ and the Student-t noise has three degrees of freedom scaled by $1/\sqrt{3}$, so every family has unit variance per coordinate. Guide spread is the s.d. of the terminal raw logit over particles. The Gaussian configuration was not rerun on the Sudoku-Extreme subset; its full-split values are in Table 1.

	Family	Sequential guiding (ours)	Oracle (guided cloud)	Guide spread
Sudoku-Extreme	Laplace	98.0 ± 1.4	98.0 ± 1.4	1.737 ± 0.052
	Student-t	98.2 ± 0.7	98.3 ± 0.9	1.757 ± 0.164
Maze-Hard	Gaussian	85.6 ± 4.4	90.9 ± 3.3	0.234 ± 0.012
	Laplace	85.5 ± 4.1	91.3 ± 3.1	0.231 ± 0.016
	Student-t	84.8 ± 4.6	91.7 ± 2.8	0.327 ± 0.016

Specifically, we introduce a blunt measure of the relationship between the two statistics introduced in §5.3: the terminal H_ℓ (19) and the path-integrated $\bar{H}_\ell$ (20). Define

$$C_\ell := \bar{H}_\ell - H_\ell, \tag{21}$$

the per-token *contraction* between the trajectory mean and the terminal snapshot. Positive C_ℓ therefore implies that the token marginal sharpened towards the final prediction.

Figure 7 reports the per-puzzle average of C_ℓ. The histograms (*left*), separated by solve outcome, reinforce the utility of the diagnostic. Correctly solved puzzles, i.e. every token is correct, admit a clear positive contraction, as expected.

The more remarkable feature is the negative skew of the unsolved-puzzle histogram. With mean contraction near zero, the long left tail represents deepening entropy with the token marginal becoming less concentrated towards the terminal step. This implies that the trajectories continue to explore rather than settle on a stable incorrect solution.

The scatter plot (*right*) gives a bird's-eye view of the same data, exposing the magnitudes of $\bar{H}_\ell$ and H_ℓ rather than only their difference. As expected from Figure 4 (*right*), the highest entropy clusters among unsolved puzzles. More striking is the threshold at roughly $(0.1, 0.1)$: below it, every prediction in the cloud is correct. Could these entropy statistics serve as calibrated guides in their own right?

E Sudoku-Extreme: train-time efficiency

In this section we ask whether guided reasoning acts as a train-time accelerator: how much earlier in training can it reach the accuracy achieved by the deterministic readout at convergence? Figure 8 applies Algorithm 1 to 19 stored checkpoints from the Sudoku-Extreme training run (epochs 2,499–49,999), under the same five-fold evaluation as the main results; the training loss is shown on the *right*.

The *left* panel reveals two observations. First, guided MAP reaches the deterministic terminal accuracy approximately $3.1\times$ earlier in training than the deterministic readout itself. We take this as a meaningful training-economics signal. Second, the method is stable across the full checkpoint range, with no fragility around any particular epoch; this also serves as a secondary ablation on the checkpoint stability of the deterministic TRM core itself.

Remark 2. The Q-head is far cheaper to train than the TRM backbone: it is a classifier whose loss converges quickly (Figure 5, *right*), whereas the backbone is the expensive object (Figure 8, *right*).

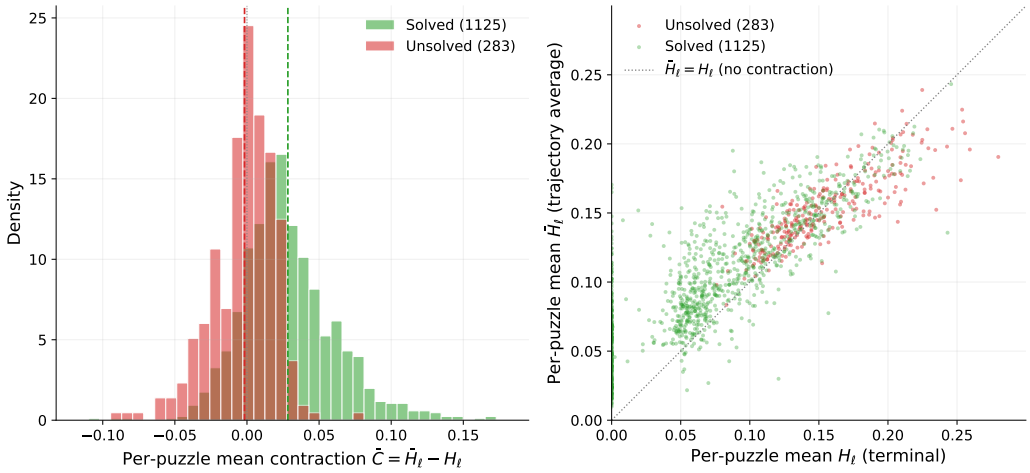

Figure 7: **Trajectory contraction on the deterministic-failure split.** *Left*: histogram of per-puzzle mean contraction $\bar{C} := \text{mean}_\ell C_\ell$ over unmasked tokens, indexed by guided MAP solve outcome. Solved puzzles have mean 0.028 whilst unsolved puzzles have mean -0.002. *Right*: per-puzzle scatter of trajectory-averaged mean entropy $\bar{H}_\ell$ against terminal mean entropy H_ℓ, with the identity, $C_\ell = 0$, shown.

This asymmetry supports the bespoke-guide direction in §7: a separately trained guide can ride on top of an existing TRM without re-touching the backbone.

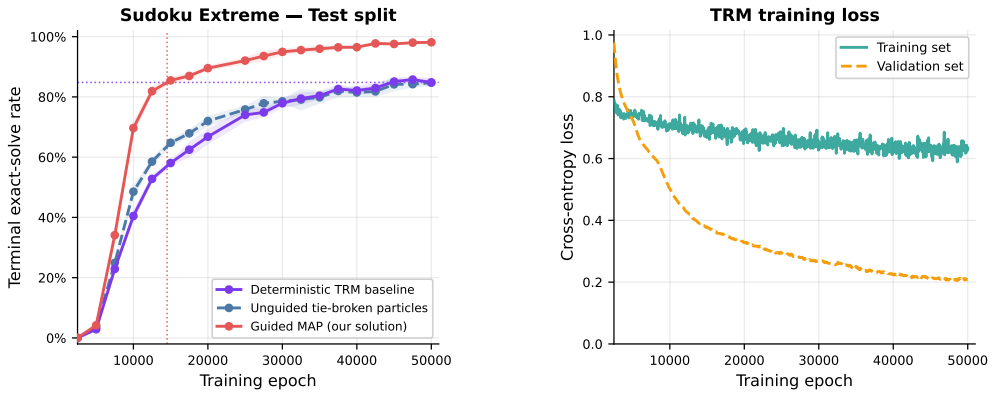

Figure 8: **Per-checkpoint scan over the TRM training run.** *Left*: terminal exact-solve rate of guided MAP (red), unguided MAP (blue), and deterministic TRM (purple) across 19 training checkpoints (epochs 2,499–49,999) on the Sudoku-Extreme test split; bands are five-fold standard deviations. Dotted horizontal: deterministic terminal accuracy at epoch 49,999. Dotted vertical: epoch at which guided MAP first crosses this threshold ($\approx 3.1\times$ earlier than the deterministic readout). *Right*: TRM training-loss trace (train and validation token cross-entropy) over the same epochs.

F Analysis of the class-conditional gap

In this section we dig deeper into the guide alignment diagnostic of §5.2. We specialise the Gibbs covariance identity (16) to the binary-success case, yielding a log-odds derivative that directly links the class-conditional gap Δ_n to the mass shift on the success set.

Fix an outer depth n and a measurable success set $A_n \subseteq \mathcal{H}$; in our experiments, A_n is the set of states that solve the puzzle by the terminal horizon N. Write $\varphi := \mathbf{1}_{A_n}$ and $p_n(\beta) := \mathbb{P}_{\pi_{n+1,\beta}}(H \in A_n)$.

Applying (16) with $\varphi = \mathbf{1}_{A_n}$ and expanding the covariance gives

$$\frac{\mathrm{d}}{\mathrm{d}\beta}\, p_n(\beta) = \mathrm{Cov}_{\pi_{n+1,\beta}}\big(\mathbf{1}_{A_n},\, q_\theta\big) = p_n(\beta)\big(1 - p_n(\beta)\big)\, \Delta_n(\beta), \tag{22}$$

where (cf. (17))

$$\Delta_n(\beta) = \mathbb{E}_{\pi_{n+1,\beta}}\big[q_\theta(H) \mid H \in A_n\big] - \mathbb{E}_{\pi_{n+1,\beta}}\big[q_\theta(H) \mid H \in A_n^c\big] \tag{23}$$

is the class-conditional mean gap in guide score between success and failure states. For $0 < p_n(\beta) < 1$, applying the logit chain rule $\frac{\mathrm{d}}{\mathrm{d}\beta} \log \frac{p}{1-p} = \frac{1}{p(1-p)} \frac{\mathrm{d}p}{\mathrm{d}\beta}$ to (22) cancels the variance factor and yields

$$\frac{\mathrm{d}}{\mathrm{d}\beta} \log \frac{p_n(\beta)}{1 - p_n(\beta)} = \Delta_n(\beta). \tag{24}$$

Equation (24) is the labelled form of the one-number test in §5.2: tempering strictly improves the log-odds of success whenever $\Delta_n(\beta) > 0$, and cannot help when the two classes are indistinguishable under q_θ. The general alignment notion of §5.2 reduces here to $\Delta_n(\beta) \geq 0$ along the tilting path. The binary-success case is the natural setting for future custom guides; for example, classifiers identifying answers that lie outside a pre-trained constraint set (§7), or those that forecast future success (via a delayed training reward).

G Proofs supporting §5

Here we record the proofs of the theoretical results stated in §5: the BCE-alignment lemma (Lemma 1, §5.2) for any strictly increasing transformation, the tube-stability bound (Proposition 1, §5.1), and the resulting tube-exit probability (Corollary 1).

G.1 Proof of Lemma 1

The tilt $\exp(t\,\psi(\eta))$ depends on h only through η, so conditioning on η removes it: the conditional law of Y given η is the same under every $\pi_{n+1,t}$, $t \in [0, \beta]$, and the hypothesis $\mathbb{E}[Y \mid \eta] = \eta$ holds under each. By the tower property it gives $\mathbb{E}[Y\,\psi(\eta)] = \mathbb{E}[\eta\,\psi(\eta)]$ and $\mathbb{E}[Y] = \mathbb{E}[\eta]$ under $\pi_{n+1,t}$, hence

$$\mathrm{Cov}_{\pi_{n+1,t}}\big(Y,\, \psi(\eta)\big) \;=\; \mathrm{Cov}_{\pi_{n+1,t}}\big(\eta,\, \psi(\eta)\big). \tag{25}$$

Strict monotonicity of ψ gives $(\eta - \eta')(\psi(\eta) - \psi(\eta')) \geq 0$ pointwise for an i.i.d. pair (η, η'), and the standard identity $2\,\mathrm{Cov}(U, V) = \mathbb{E}[(U - U')(V - V')]$ for i.i.d. copies gives

$$2\,\mathrm{Cov}\big(\eta,\, \psi(\eta)\big) = \mathbb{E}\big[(\eta - \eta')(\psi(\eta) - \psi(\eta'))\big] \;\geq\; 0, \tag{26}$$

with equality iff η is constant on the support of $\pi_{n+1,t}$. The case $\psi = \log$ gives the BCE-population statement. □

G.2 Proof of Proposition 1

Fix $h_n = (y_n, z_n)$ and suppress n, x, h_n from the notation, setting

$$F_\theta(z) := f_\theta(x + y_n + z). \tag{27}$$

Then the deterministic and stochastic inner trajectories satisfy $z_1 := z_{n,1} = f_\theta(x + y_n + z_n)$, $z_{m+1} = F_\theta(z_m)$, as well as $\zeta_1 = z_1 + \sigma\xi_0$, $\zeta_{m+1} = F_\theta(\zeta_m) + \sigma\xi_m$ for $m = 1, \ldots, M-1$.

Write $e_m := \zeta_m - z_m$ as in (13), and for $m \geq 0$ let

$$E_m := \{\|e_1\| \leq r, \ldots, \|e_m\| \leq r\}, \tag{28}$$

with the convention $E_0 := \Omega$ (the empty intersection); equivalently, $E_m = \{\tau_r > m\}$ for $m \geq 0$.

Base case. Since $e_1 = \sigma\xi_0$ and $\mathbf{1}_{E_1} \leq 1$,

$$\mathbb{E}\big[\|e_1\|^2 \mathbf{1}_{E_1}\big] \leq \sigma^2 \mathbb{E}\|\xi_0\|^2 \leq \sigma^2 \nu^2. \tag{29}$$

One-step recursion. Fix $m \in \{1, \ldots, M-1\}$, and write

$$D_m := F_\theta(\zeta_m) - F_\theta(z_m), \tag{30}$$

so that $e_{m+1} = D_m + \sigma\xi_m$. Condition on $\mathcal{F}_m := \sigma(\xi_0, \ldots, \xi_{m-1})$: both D_m and $\mathbf{1}_{E_m}$ are $\mathcal{F}_m$-measurable, while $\xi_m \perp \mathcal{F}_m$ has mean zero. Expanding the square and taking expectations, the cross-term vanishes and the noise term satisfies $\mathbb{E}[\|\xi_m\|^2 \mathbf{1}_{E_m}] \le \mathbb{E}\|\xi_m\|^2 \le \nu^2$, so

$$\mathbb{E}\left[\|e_{m+1}\|^2 \mathbf{1}_{E_m}\right] \le \mathbb{E}\left[\|D_m\|^2 \mathbf{1}_{E_m}\right] + \sigma^2\nu^2. \tag{31}$$

On E_m, $\|e_m\| \le r$, so $\zeta_m \in \{z : \|z - z_m\| \le r\}$; both z_m and ζ_m therefore lie in $\bigcup_{j=1}^{M}\{z : \|z - z_j\| \le r\}$, and the local Lipschitz hypothesis gives $\|D_m\| \le \rho_r\|e_m\|$ on E_m. Substituting into (31) yields the one-step bound

$$\mathbb{E}\left[\|e_{m+1}\|^2 \mathbf{1}_{E_m}\right] \le \rho_r^2\, \mathbb{E}\left[\|e_m\|^2 \mathbf{1}_{E_m}\right] + \sigma^2\nu^2. \tag{32}$$

Since $E_{m+1} \subseteq E_m$, in particular

$$\mathbb{E}\left[\|e_{m+1}\|^2 \mathbf{1}_{E_{m+1}}\right] \le \rho_r^2\, \mathbb{E}\left[\|e_m\|^2 \mathbf{1}_{E_m}\right] + \sigma^2\nu^2. \tag{33}$$

Iterating the recursion. Define $a_m := \mathbb{E}\left[\|e_m\|^2 \mathbf{1}_{E_m}\right]$ for $m = 1, \ldots, M$. By (29) and (33),

$$a_1 \le \sigma^2\nu^2, \qquad a_{m+1} \le \rho_r^2 a_m + \sigma^2\nu^2. \tag{34}$$

Iterating gives

$$a_m \le \sigma^2\nu^2 \sum_{j=0}^{m-1} \rho_r^{2j} \le \frac{\sigma^2\nu^2}{1 - \rho_r^2} \tag{35}$$

for $m = 1, \ldots, M$, which recovers (14). □

G.3 Proof of Corollary 1

Using the notation and conventions from the previous proof ($E_m = \{\tau_r > m\}$ for $m \ge 0$, with $E_0 = \Omega$), define

$$b_m := \mathbb{E}\left[\|e_m\|^2 \mathbf{1}_{E_{m-1}}\right] \qquad (m = 1, \ldots, M). \tag{36}$$

We first show that

$$b_m \le \sigma^2\nu^2 \sum_{j=0}^{m-1} \rho_r^{2j} \le \frac{\sigma^2\nu^2}{1 - \rho_r^2}. \tag{37}$$

For $m = 1$, since $E_0 = \Omega$ and $e_1 = \sigma\xi_0$, $b_1 = \mathbb{E}\|e_1\|^2 \le \sigma^2\nu^2$. For $m \ge 1$, the one-step bound (32) gives

$$b_{m+1} = \mathbb{E}\left[\|e_{m+1}\|^2 \mathbf{1}_{E_m}\right] \le \rho_r^2\, \mathbb{E}\left[\|e_m\|^2 \mathbf{1}_{E_m}\right] + \sigma^2\nu^2 \le \rho_r^2\, b_m + \sigma^2\nu^2, \tag{38}$$

where the last inequality uses $E_m \subseteq E_{m-1}$ so that $\mathbb{E}[\|e_m\|^2 \mathbf{1}_{E_m}] \le b_m$. The bound (37) follows by induction.

Next, $\{\tau_r = m\} = E_{m-1} \cap \{\|e_m\| > r\}$, on which $\|e_m\|^2 \mathbf{1}_{E_{m-1}} \ge r^2$. Markov's inequality therefore gives

$$\mathbb{P}(\tau_r = m) \le \frac{1}{r^2}\, \mathbb{E}\left[\|e_m\|^2 \mathbf{1}_{E_{m-1}}\right] = \frac{b_m}{r^2}. \tag{39}$$

Summing over $m = 1, \ldots, M$,

$$\mathbb{P}(\tau_r \le M) = \sum_{m=1}^{M} \mathbb{P}(\tau_r = m) \le \frac{1}{r^2} \sum_{m=1}^{M} b_m \le \frac{M}{r^2} \frac{\sigma^2\nu^2}{1 - \rho_r^2}, \tag{40}$$

which recovers (15). □

Remark 3 (Noise assumptions)**.** Proposition 1 uses the noise only through independence, zero mean and the second-moment bound $\mathbb{E}\|\xi_m\|^2 \le \nu^2$. Any such family qualifies: isotropic Gaussian noise gives $\nu^2 = LD$; Laplace or Student-t noise with more than two degrees of freedom, scaled to unit variance per coordinate, gives the same ν^2 and hence the same bound; anisotropic noise with covariance Σ enters through $\nu^2 = \operatorname{tr}\Sigma$. Families without a second moment, such as Cauchy noise, make the bound vacuous. Appendix C reports the matching experiment.

Remark 4 (Dependence on the inner depth)**.** The in-tube bound (14) is uniform in M: the geometric sum is bounded by $\sigma^2\nu^2/(1 - \rho_r^2)$ for every m, so a deeper inner recursion does not loosen the control on the deviation while the rollout stays in the tube. Only the exit probability (15) grows, linearly in M, through the union over steps: at fixed σ the price of depth is paid in how often the control applies, not in how tight it is.

H Compute-matched selection rules

The rescoring rules of Table 1 re-compute statistics in a stored particle cloud of $S = 16$ unguided ($\beta = 0$) trajectories with no further backbone calls. These are: *plurality vote*, the most frequent decoded answer, ties to the lowest index; *terminal-Q*, the particle with the largest raw guide logit at step N, which is Algorithm 1 without resampling and with the tilt applied once at the final step, the best-of-K rule of PTRM (Sghaier et al., 2026); *path-sum tilt*, $\arg\max_s \sum_{n=1}^{N} \log \mathrm{sigmoid}(\tilde{q}_n^{(s)})$, the Feynman–Kac tilt (8) without resampling; and *oracle*, which counts a puzzle solved if any particle is correct. Table 7 compares the rules on the main-results runs, so the differences are paired; Table 8 degrades the guide with Gaussian noise. The path-sum tilt is the more robust rule: at $\sigma_q = 16$ it loses four points on Sudoku DF and under five on PPBench golden, where terminal-Q loses 23 and 14. On Maze-Hard, where the guide is flat, neither rule moves by more than two points at any noise level.

Table 7: **Paired selection rules on the main-results runs** (5 seeds $\times$ 5 folds; exact-solve %, mean $\pm$ s.d. over the 25 cells; differences are paired within cells and carry 95% confidence intervals). Guided and unguided clouds share seed and fold. Puzzles evaluated: Sudoku-Extreme Test 9,984, DF 1,408; Maze-Hard Test 992, DF 128.

	Sudoku Test	Sudoku DF	Maze Test	Maze DF
Single rollout ($\beta = 0$, first particle)	84.3 ± 0.7	35.7 ± 2.8	85.2 ± 2.5	13.1 ± 6.5
Sequential guiding (ours)	98.0 ± 0.2	85.9 ± 1.5	85.6 ± 2.8	12.5 ± 5.5
Terminal-Q, guided cloud	98.0 ± 0.2	85.9 ± 1.5	86.8 ± 2.4	19.1 ± 9.1
Terminal-Q, unguided cloud	98.2 ± 0.3	86.8 ± 1.8	86.8 ± 2.5	18.1 ± 8.3
Oracle, guided cloud	98.0 ± 0.2	86.0 ± 1.5	90.7 ± 2.2	37.1 ± 8.1
Oracle, unguided cloud	98.2 ± 0.3	86.8 ± 1.8	91.1 ± 2.2	37.4 ± 8.7
Terminal-Q (unguided) minus ours	$+0.2 \pm 0.1$	$+0.9 \pm 0.7$	$+1.2 \pm 0.3$	$+5.6 \pm 2.8$
Ours minus single rollout	$+13.7 \pm 0.3$	$+50.2 \pm 1.0$	$+0.4 \pm 0.6$	-0.6 ± 2.6
Oracle guided minus oracle unguided	-0.2 ± 0.1	-0.8 ± 0.7	-0.4 ± 0.4	-0.3 ± 3.1

Table 8: **Selection under a corrupted guide.** I.i.d. Gaussian noise of s.d. σ_q is added to every raw guide logit of the stored unguided clouds and the rules are re-read (8 draws per cell); exact-solve %, mean $\pm$ s.d. over 5 folds (Sudoku-Extreme and Maze-Hard, full splits of 10,020, 1,427 and 1,000 puzzles, one seed) or over 5 seeds (PPBench golden, 42 puzzles). T is terminal-Q, P the path-sum tilt. The online filter was not rerun under corruption.

	Sudoku Test		Sudoku DF		Maze Test		PPBench golden	
σ_q	T	P	T	P	T	P	T	P
0	98.1 ± 0.5	98.1 ± 0.5	88.4 ± 0.6	88.1 ± 0.9	86.9 ± 4.0	85.7 ± 3.8	75.7 ± 2.0	75.7 ± 3.1
4	98.1 ± 0.5	97.9 ± 0.5	88.2 ± 0.7	87.2 ± 1.3	85.8 ± 3.7	86.0 ± 4.3	73.4 ± 1.2	74.4 ± 2.1
8	96.9 ± 0.5	97.8 ± 0.5	81.9 ± 1.0	86.2 ± 1.5	85.8 ± 3.8	85.8 ± 4.0	67.2 ± 0.8	74.2 ± 1.9
16	93.4 ± 0.6	97.4 ± 0.6	65.3 ± 0.9	83.9 ± 1.6	85.3 ± 4.0	85.9 ± 3.7	62.1 ± 1.7	71.2 ± 2.7
32	89.6 ± 0.7	95.8 ± 0.8	51.4 ± 1.1	75.6 ± 1.5	85.7 ± 3.9	85.5 ± 3.9	59.8 ± 1.4	66.9 ± 1.8

I PPBench

PPBench (Waugh, 2026) is a pencil-puzzle benchmark of 300 golden puzzles across 20 varieties. PTRM (Sghaier et al., 2026) trains a TRM on six of them. We train one with attention on the four whose answers are orientation-free with a single clue layer, Sudoku, Light Up, Nurikabe and Tapa, so that dihedral augmentation needs no token remapping (33,460 training examples; $H_{\text{cycles}} = 3$, $L_{\text{cycles}} = M = 4$). Heyawake and Shakashaka are excluded, so our aggregate is not comparable with PTRM's. We evaluate on the 42 golden puzzles of these types and on a held-out 253-puzzle validation split from the public pool, disjoint from training and golden. Inference reuses $S = 16$, $N = 48$, $\sigma = 0.3$ and $\beta = 0.25$ from the Sudoku-Extreme sweep and fixes $\tau_{\text{ESS}} = 0.5$ a priori; nothing is tuned on PPBench. Table 9 gives every rule by type. Over 5 seeds, sequential guiding lifts validation exact-solve from 62.5% (deterministic) to $77.9 \pm 0.6\%$, ahead on every type, and golden from 59.5% to $74.8 \pm 2.1\%$. The gain concentrates on Sudoku and Tapa, the types with the most headroom; on golden Sudoku the plurality vote falls below a single rollout while every guided rule

roughly doubles it. Guide spread and class gap Δ_N are large on every type, as on Sudoku-Extreme and unlike Maze-Hard. The deterministic depth sweep is in Table 11.

Table 9: **PPBench by puzzle type** (exact-solve %, mean $\pm$ s.d. over 5 seeds; the deterministic TRM is one run). Rescoring rules and the oracle read the stored unguided clouds, and the single rollout here is the mean per-particle rate of those clouds. The last two rows of each block give the guide spread (s.d. of the terminal raw logit over particles, averaged over puzzles) and the class gap Δ_N, both in raw-logit units.

	All	Sudoku	Light Up	Nurikabe	Tapa
Validation (253 puzzles: 95, 63, 67, 28)					
Deterministic TRM	62.5	47.4	93.7	68.7	28.6
Single rollout (mean per particle)	63.8±0.5	48.9±0.8	89.9±0.4	71.1±1.1	37.9±1.0
Plurality vote	67.7±1.4	51.8±2.2	93.3±0.6	76.1±1.6	43.6±2.7
Terminal-Q	79.5±0.8	79.2±1.5	95.6±1.6	78.2±1.5	47.9±3.6
Path-sum tilt	78.7±0.7	77.5±2.5	95.6±1.6	77.6±1.9	47.9±4.8
Sequential guiding (ours)	77.9±0.6	74.3±1.2	94.6±0.9	78.8±1.9	50.7±3.0
Oracle	82.8±0.7	82.3±1.7	95.9±1.6	83.6±2.1	53.6±2.3
Guide spread	1.770±0.035	2.708±0.089	1.177±0.043	1.123±0.039	1.467±0.060
Class gap Δ_N	8.2±0.1	9.3±0.1	12.2±0.1	4.2±0.1	5.2±0.1
Golden (42 puzzles: 15, 8, 9, 10)					
Deterministic TRM	59.5	46.7	87.5	88.9	30.0
Single rollout (mean per particle)	57.0±0.8	31.4±3.2	96.6±1.4	81.1±0.5	41.9±2.4
Plurality vote	61.0±3.2	28.0±10.7	100.0±0.0	88.9±0.0	54.0±4.9
Terminal-Q	75.7±1.8	69.3±3.3	100.0±0.0	88.9±0.0	54.0±4.9
Path-sum tilt	75.7±2.8	66.7±4.2	100.0±0.0	88.9±0.0	58.0±7.5
Sequential guiding (ours)	74.8±2.1	62.7±3.7	100.0±0.0	88.9±0.0	60.0±7.1
Oracle	80.5±1.0	69.3±3.3	100.0±0.0	88.9±0.0	74.0±4.9
Guide spread	1.790±0.083	2.631±0.132	1.111±0.262	1.072±0.097	1.719±0.116
Class gap Δ_N	8.5±0.1	9.8±0.1	11.7±0.6	2.8±0.4	6.2±0.1

J Guide quality: calibration and discrimination

For each benchmark we score every particle of the stored unguided clouds by $\mathrm{sigmoid}(\tilde{q})$ at every third outer step and compute, per fold, the ten-bin expected calibration error (ECE) of that score against per-particle exact correctness, and its AUROC as a classifier of correctness, pooling particles over puzzles (Table 10). Sudoku-Extreme is calibrated and discriminating from step 6 onwards. Maze-Hard is neither: on Test the AUROC falls from 0.89 at step 3 to 0.68 at step 48 while the ECE rises to 0.14, and the terminal mean confidence of 0.995 sits against a per-particle success rate of 0.856 (Sudoku-Extreme: 0.961 against 0.844); on DF, where few particles are correct, the guide is no better than chance (AUROC 0.46 to 0.58). PPBench golden sits between, with AUROC 0.99 and ECE 0.03 to 0.08.

Table 10: **Guide calibration and discrimination by outer step** on the stored unguided clouds (mean $\pm$ s.d. over 5 folds, or over 5 seeds for PPBench golden). ECE: ten-bin expected calibration error of $\mathrm{sigmoid}(\tilde{q})$ against per-particle correctness; AUROC: pooled over particles and puzzles. A dash marks a step at which the AUROC is undefined.

		$n=3$	$n=6$	$n=12$	$n=24$	$n=48$
Sudoku Test	ECE	0.004±0.001	0.002±0.001	0.001±0.000	0.001±0.000	0.000±0.000
	AUROC	0.994±0.001	1.000±0.000	1.000±0.000	1.000±0.000	1.000±0.000
Sudoku DF	ECE	0.001±0.000	0.001±0.000	0.001±0.000	0.001±0.000	0.001±0.000
	AUROC	–	1.000±0.000	1.000±0.000	1.000±0.000	1.000±0.000
Maze Test	ECE	0.087±0.027	0.133±0.036	0.138±0.036	0.138±0.036	0.139±0.036
	AUROC	0.894±0.026	0.768±0.044	0.696±0.037	0.697±0.037	0.684±0.033
Maze DF	ECE	0.571±0.030	0.846±0.052	0.864±0.046	0.869±0.045	0.875±0.041
	AUROC	0.485±0.088	0.464±0.127	0.563±0.158	0.563±0.163	0.547±0.181
PPBench golden	ECE	0.063±0.005	0.079±0.004	0.059±0.007	0.042±0.008	0.033±0.002
	AUROC	0.986±0.004	0.993±0.003	0.995±0.002	0.993±0.002	0.989±0.001

Intervening on the guide. Alignment in the sense of Lemma 1 depends only on the ranking, which suggests three interventions of increasing cost, none demonstrated here. If calibration fails but ranking survives, a monotone recalibration (temperature or isotonic) repairs the ECE and leaves the lemma intact. If discrimination fails, as on Maze-Hard, no recalibration can help and the guide must be refitted, which is cheap relative to the backbone (Appendix E); this is why §7 sends Maze-Hard to a refitted guide rather than to recalibration. Where neither suffices, the guide is built bespoke for the task (§7).

K Depth, particles and cost

Tables 11 to 14 collect the depth, particle-count and cost evidence behind §6. Deterministic recursion improves slowly with depth: at $N = 768$, where its 5,376 backbone calls match the $S = 16$, $N = 48$ cloud, it reaches 95.7% on a 256-puzzle Sudoku-Extreme Test subset, still short of the cloud; on PPBench validation it moves from 62.5% to 63.6% and on golden it declines. Sequential guiding passes the full-depth deterministic baseline at step 9 on Sudoku-Extreme Test, reaches $96.3\pm0.4\%$ at $N = 24$ and is still rising at $N = 48$; on Maze-Hard the curve is flat from $N = 6$. Over particle count on a 512-puzzle subset (three folds, hence wide intervals) the guided rate tracks the unguided oracle at every S; Figure 6 has the sweep on 2,000 puzzles. Table 3 (Appendix A) gives wall-clock and memory against S on one H100, measured end to end on 64 puzzles in four batches of 16 at $N = 48$, so it reports batched throughput rather than single-query latency. Finally, halting the stored unguided cloud when $\max_s \mathrm{sigmoid}(\tilde{q}_n^{(s)})$ first exceeds a threshold, checked every third step, matches the full-horizon terminal-Q rate at a fraction of the budget wherever the guide is calibrated; on Maze-Hard it stops after three steps and gives up around two points.

Table 11: **Deterministic recursion against outer depth** (exact-solve %, one run; $N = 48$ is the trained horizon and $N = 768$ matches the backbone calls of the $S = 16$, $N = 48$ cloud). Sudoku-Extreme: a 256-puzzle Test subset; PPBench: validation (253 puzzles) and golden (42).

N	Sudoku Test (subset)	PPBench validation	PPBench golden
48	86.7	62.5	59.5
96	90.2	62.5	59.5
192	92.6	62.8	59.5
384	93.8	63.2	57.1
768	95.7	63.6	57.1

Table 12: **Sequential guiding against outer depth** (5 seeds $\times$ 5 folds; exact-solve % of the highest-weight particle at step N, mean $\pm$ s.d. over cells). The deterministic TRM at $N = 48$ scores 85.9, 0.0, 86.6 and 0.0 on the four splits.

N	Sudoku Test	Sudoku DF	Maze Test	Maze DF
6	75.0 ± 0.8	22.0 ± 2.4	85.0 ± 2.3	10.4 ± 5.1
12	91.7 ± 0.6	58.6 ± 3.2	85.2 ± 2.5	11.1 ± 5.5
24	96.3 ± 0.4	77.4 ± 2.8	85.4 ± 2.6	11.3 ± 5.9
36	97.5 ± 0.3	83.1 ± 1.8	85.5 ± 2.7	12.0 ± 5.5
48	98.0 ± 0.2	85.9 ± 1.5	85.6 ± 2.8	12.5 ± 5.5

Table 13: **Particle count at fixed depth** ($N = 48$; a 512-puzzle Sudoku-Extreme Test subset, 3 folds of one seed, $\pm$ 95% confidence interval over folds). The deterministic TRM scores 87.3 ± 6.9 on the same subset, and the $S = 1$ guided run is a single stochastic rollout.

	$S = 1$	$S = 2$	$S = 4$	$S = 8$	$S = 16$
Sequential guiding (ours)	83.5 ± 7.4	90.8 ± 10.1	94.3 ± 7.6	96.0 ± 6.2	98.5 ± 2.9
Oracle (unguided cloud)	86.2 ± 9.1	90.1 ± 9.1	94.7 ± 5.6	97.0 ± 4.1	97.7 ± 3.7

Table 14: **Anytime halting on the stored unguided clouds.** Halt at the first step $n \in \{3, 6, \ldots, 48\}$ with $\max_s \mathrm{sigmoid}(\tilde{q}_n^{(s)}) \geq \tau$ and return the particle with the largest score. Exact-solve % at the stop (mean $\pm$ s.d. over 5 folds, or 5 seeds for PPBench golden), mean stop step, share of the $N = 48$ budget used, and terminal-Q at the full horizon.

	τ	Exact at stop	Mean stop step	Budget used	Full horizon
Sudoku Test	0.90	97.4±0.6	6.9	14.4%	98.1
	0.95	98.0±0.5	7.0	14.5%	
	0.99	98.1±0.5	7.1	14.7%	
Sudoku DF	0.90	88.2±0.7	16.8	35.0%	88.4
	0.95	88.3±0.7	16.8	35.1%	
	0.99	88.4±0.6	16.9	35.2%	
Maze Test	0.90	84.6±3.8	3.1	6.4%	86.9
	0.95	84.8±3.9	3.1	6.4%	
	0.99	85.3±3.6	3.3	6.9%	
PPBench golden	0.90	68.6±2.6	12.7	26.4%	75.7
	0.95	71.9±1.1	13.8	28.8%	
	0.99	75.2±2.1	16.8	35.0%	

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and §1 state three operational claims (Sudoku-Extreme $85.9\% \rightarrow 98.0\%$ exact-solve, PPBench $62.5\% \rightarrow 77.9\%$ with the transferred configuration, and a label-free Maze-Hard non-improvement prediction), a comparative claim (the gain comes from the guide rather than the sampling, and sequential reweighting holds under a degraded guide where terminal selection does not), and three diagnostic claims (tube stability, guide alignment, token-marginal entropy). The Sudoku and PPBench claims are verified in Table 1 and Appendix I; the comparative claim in Table 1 and Appendix H. The Maze prediction is verified in Figure 3 and the Maze-Hard row of Table 1. The diagnostics are stated in Proposition 1, Lemma 1, and §5.3, with empirical evaluation in Figures 2, 3, 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A 'Limitations and scope' details our assessment, found at the end of §6. This includes: (i) empirical scope (Sudoku-Extreme, Maze-Hard and PPBench, all structured puzzles solved by a backbone with a trained halting head; language tasks and ARC-AGI are not evaluated), (ii) inference-only scope over three fixed backbones and outer horizon N, with weight-updating test-time adaptation noted as complementary and not compared, (iii) the theoretical assumptions (f_θ locally Lipschitz on the r-tube, Proposition 1; a Q-head aligned in the sense of Lemma 1, which BCE training motivates but does not guarantee), (iv) the necessary-but-not-sufficient nature of the label-free spread bound (18), (v) the methodological caveat that our Q-head is trained for early stopping rather than trajectory ranking, and (vi) that the recovery gates of §7 are proposals, and that the comparison with PTRM's selection rule rescores our stored clouds rather than rerunning that method. The Maze-Hard case study is itself a non-improvement-result example (§B), and the directions in §7 make the open follow-ups explicit.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.

- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Proposition 1 states a local Lipschitz hypothesis on the r-tube and a finite second-moment hypothesis on the noise; Corollary 1 inherits these, and Remarks 3 and 4 state what the proof uses of the noise and of the inner depth. Lemma 1 states a tower-property hypothesis on η, which holds at the population BCE optimum and is tested empirically in Appendix J. Proofs are given in Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Algorithm 1 states the inference procedure end-to-end. Appendix A ('Models and data', 'Splits and statistics', 'Hyperparameter selection') gives the dataset versions, splits, 5×5 seed-fold protocol, error-bar definition, and validation methodology. Table 4 pins the five inference hyperparameters (σ, β, S, N, τ_{ESS}) for all three benchmarks, Table 2 the puzzle counts and call budgets, Appendix I the PPBench dataset construction and training recipe, and Appendix H the definition of every rescoring rule. The full code, configuration files, and cached run outputs are provided in the public repository (`https://github.com/digiLab-ai/bigr`), with default arguments in every `run_*.py` script as used in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The codebase is public at `https://github.com/digiLab-ai/bigr`. Its README documents (a) a no-GPU figure-regeneration path that uses cached run outputs, and (b) a full GPU re-run path with exact commands, expected wall-clock per experiment, and a reference table. Sudoku-Extreme and Maze-Hard are HRM-released datasets and are not redistributed; instructions to retrieve them are provided. The PPBench training, validation and golden splits are derived from the public PPBench release (Waugh, 2026) by the recipe in Appendix I. TRM checkpoints are not bundled (size); retraining instructions and configs are included. The stored particle traces and the machine-generated evidence register from which every table in the paper is built are included.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Inference hyperparameters are in Table 4 with selection methodology in Appendix A. Training follows Nano-TRM defaults, cited in Appendix A, 'Models and data'. Splits, validation subset, fold sizes, and seed protocol are given in Appendix A, 'Splits and statistics', and the puzzle counts in Table 2. PPBench (puzzle types, splits, training recipe, transferred inference configuration with $\tau_{\mathrm{ESS}} = 0.5$) is in Appendix I; the rescoring rules, the corruption protocol and the anytime rule are in Appendices H and K.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The core runs of Table 1 and bands in Figures 1 are 1-sigma standard deviations over 5 random-noise seeds $\times$ 5 disjoint Test folds (25 runs); the rescoring rules of Table 1 and the appendices re-read stored clouds of the full splits over 5 folds of one seed, PPBench over 5 seeds, as each caption states; paired comparisons between selection rules on the same runs carry 95% confidence intervals (Table 7); the remaining presented results are computed on a 5-fold single-seed protocol only due to compute constraints. (Note that in some plots errors are not visible due to the scale of the y-axis.) Specifics in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).

- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A, 'Compute resources', states that all training and inference runs use a single NVIDIA H100 GPU and reports approximately 20–30 GPU-hours total for the full set of reported experiments. Note that preliminary or discarded runs are of comparable order. The PPBench backbone training, its inference runs and the stored-trace analyses of Appendices H to K were run afterwards on the same hardware and are not included in that figure; Table 3 reports measured inference cost against particle count. The repository README.md provides per-experiment wall-clock estimates within the same budget.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses only public structured-reasoning benchmarks (Sudoku-Extreme, Maze-Hard, PPBench); no human subjects, personal data, or sensitive content involved.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: §7 and §1 discuss positive impacts (compact, auditable reasoning models suited to latency- and privacy-constrained deployment), and in particular we highlight future work intended for AI safety and 'steering' AI models with safety constraints. While not the focus of the paper we note that the method modifies inference rather than introducing new capabilities, limiting direct paths to misuse.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released code targets three structured-reasoning benchmarks with $\sim$7M-parameter solvers. We do not release pretrained generative models, large language models, or scraped datasets; the work poses no foreseeable misuse risk that warrants safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All upstream assets are cited at first use and used under their original licenses, listed alongside their upstream URLs in the released repository: the deterministic TRM backbone via Nano-TRM (`https://github.com/olivkoch/nano-trm`, MIT licence) (Koch, 2025); the Sudoku-Extreme dataset (`https://huggingface.co/datasets/sapientinc/sudoku-extreme`) and the Maze-Hard dataset (`https://huggingface.co/datasets/sapientinc/maze-30x30-hard-1k`), both released by the HRM authors (Wang et al., 2025); neither dataset page states a licence, and we use the data for research as the HRM and TRM papers do; PPBench (Waugh, 2026) via its public package and repository (`https://github.com/approximatelabs/pencil-puzzle-bench`, MIT licence).

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released repository contains a README with installation, run, and figure-regeneration instructions, and is licensed for open use under the MIT LICENCE; it includes the stored particle traces, the evidence register that generates every table, and the PPBench splits as used.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the core methodology of the work; the work studies tiny recursive models trained without LLM components. Whilst LLMs can be benchmarked against on the Sudoku and Maze datasets (Jolicoeur-Martineau, 2025), we reference against recursive model baselines only. LLMs have been used for editorial assistance during writing and code review, including the camera-ready revision.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.