BEYOND STABLE CONVERGENCE: NOISE COVARIANCE DESIGN FOR EFFICIENT TEST-TIME SCALING IN LOOPED MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Looped models scale test-time computation through recurrent refinement, but stable convergence alone does not ensure that additional iterations improve predictions. Even near an attracting fixed point, the fixed-point residual can decrease while the predictive loss increases. We formalize this discrepancy as misalignment between the attractor and task landscapes and develop a noise covariance learning method for efficient test-time depth scaling. Through a local analysis, we establish a noise covariance design framework that accounts for task-geometry adaptation under noisy training and optimizes a directional task-loss descent bound. Building on this framework, we propose Noise Designed Looped Models (NDLM), which employs an equivalent matrix-residual objective to train a lightweight network to generate structured perturbations efficiently. Experiments across multiple reasoning tasks demonstrate NDLM's competitive performance. Remarkably, NDLM enables test-time depth scaling with no observed overthinking and matches the isotropic-noise baseline's reference accuracy with an estimated 82.7% reduction in recurrent depth on Maze-Unique.

1 INTRODUCTION

Looped models offer a natural approach to test-time scaling by repeatedly applying a shared computational block to refine a latent state (Dehghani et al., 2019; Geiping et al., 2026). This recurrent structure increases the effective depth without increasing the parameter count, enabling models to allocate more computation to difficult inputs and solve challenging reasoning problems with remarkably few trainable parameters (Wang et al., 2025; Jolicoeur-Martineau, 2025; Movahedi et al., 2026). To support reliable test-time scaling, recent work has sought to stabilize recurrent dynamics and promote convergence to fixed-point attractors through constrained state-transition parameterizations (Prairie et al., 2026), Jacobian spectral-radius regularization (Yang et al., 2026), and equilibrium-based training (Fein-Ashley & Rashidinejad, 2026).

However, convergence alone does not guarantee that further computation improves predictions, as a trajectory may converge to a spurious attractor with a small fixed-point residual but high task error (Huang et al., 2026). Recent methods introduce stochastic latent updates to explore multiple reasoning trajectories and improve test-time scaling (Huang et al., 2026; Baek et al., 2026). Despite these empirical gains, the mechanism by which noisy training makes further recurrent computation task-beneficial remains unclear, and a principled basis for selecting the noise covariance to improve depth efficiency remains to be established. This motivates two questions:

- *Why and under what conditions can noisy training promote test-time depth scaling in attractor-based models?*
- *How should noise covariance be designed to improve inference-depth efficiency?*

We address these questions through a local analysis near an attracting fixed point, taking dynamical stability as given. As recurrent states approach the attractor, local dynamics increasingly govern whether additional iterations improve predictions or merely reinforce a suboptimal outcome. We first formalize misalignment between the attractor and task landscapes, showing that residual contraction can coexist with increasing task loss. We then establish how noise covariance shapes stationary

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

exploration and, through noisy training, the learned task geometry. Under a fixed-volume surrogate, this adaptation yields a task metric for which the deterministic contribution to local task-loss change is nonpositive, providing a mechanism for task-aligned recurrent refinement. To improve inference-depth efficiency, we optimize a lower bound on the deterministic task-loss descent rate under a fixed noise budget and characterize the optimal covariance through an equivalent matrix-residual minimization problem. This characterization leads to *Noise Designed Looped Models* (NDLM), which learns structured perturbations through a lightweight conditional network without explicitly constructing the full covariance matrix.

Experiments on different reasoning tasks demonstrate competitive reasoning performance. Notably, NDLM improves accuracy over EqR (Huang et al., 2026) by 1.9% point on Maze-Unique (Huang et al., 2026) and 2.9% point on Maze-Hard (Jolicoeur-Martineau, 2025). More importantly, NDLM enables efficient test-time depth scaling. On Maze-Unique, its accuracy increases from 65.6% at depth 8 to 92.9% at depth 128, while reaching EqR's reference accuracy of 89.8% with an estimated 82.7% reduction in recurrent depth. Furthermore, with breadth scaling enabled, both TRM (Jolicoeur-Martineau, 2025) and EqR exhibit overthinking, whereas NDLM maintains non-decreasing accuracy across the evaluated depths. These results demonstrate that structured noise supports effective test-time depth scaling while improving inference-depth efficiency.

Our contributions are summarized as follows.

- We formalize attractor–task misalignment by showing that stable residual contraction need not imply task-loss descent. Through a local analysis, we further establish how noisy training reshapes task geometry to support predictive improvement during recurrent refinement.
- We propose Noise Designed Looped Models (NDLM), which designs noise covariance by optimizing a lower bound on deterministic task-loss descent. We derive an equivalent matrix-residual characterization and translate it into a tractable objective for learning perturbations through a lightweight network.
- Experiments on Sudoku-Extreme, Maze-Unique, and Maze-Hard demonstrate competitive reasoning performance. NDLM enables test-time depth scaling with no observed overthinking over the evaluated range and reaches EqR's reference accuracy with an estimated 82.7% reduction in recurrent depth on Maze-Unique.

1.1 Related Work

Looped models. Looped models reuse parameters across successive updates, allowing computational depth to increase independently of parameter count. Some studies connect looped Transformers to programmable computation and iterative algorithms (Giannou et al., 2023; Yang et al., 2024), while HRM and TRM demonstrate compact recurrent reasoning through hierarchical and recursive latent refinement (Wang et al., 2025; Jolicoeur-Martineau, 2025). Recent work extends these ideas to language modeling at scale through looped pretraining (Saunshi et al., 2025), recurrent-depth architectures (Geiping et al., 2026; Zhu et al., 2025), and recurrence retrofitted into pretrained language models (McLeish et al., 2026). To enable reliable test-time depth scaling, a related line of work investigates the stability, convergence, and attractor structure of recurrent computation. DEQs define representations implicitly as fixed points (Bai et al., 2019), while more recent approaches promote stable recurrent dynamics through structured state transitions (Prairie et al., 2026) and regularization of the estimated recurrent Jacobian spectral radius (Yang et al., 2026). Mechanistic analyses characterize how architectural choices shape the emergence of fixed-point attractors (Blayney et al., 2026). Building on the attractor-based perspective, Attractor Models formulate prediction refinement as fixed-point solving (Fein-Ashley & Rashidinejad, 2026). To further improve inference-depth efficiency, recent approaches move beyond fixed-depth recurrence by adapting computation through convergence-based halting (Movahedi et al., 2026), budget-conditioned recurrent trajectories (Jeddi et al., 2026), and selective latent iterations (Fu et al., 2025).

Stochastic looped models. Stochasticity offers several mechanisms for improving iterative prediction. Diffusion and flow models learn transformations from noise to data (Ho et al., 2020; Lipman et al., 2023), motivating iterative methods based on annealed energy landscapes (Du et al., 2024), revision of discrete predictions (Wang et al., 2026), and recursive denoising. FRM combines flow modeling with self-conditioned recurrent refinement (Helbling et al., 2026), while Looped Flows

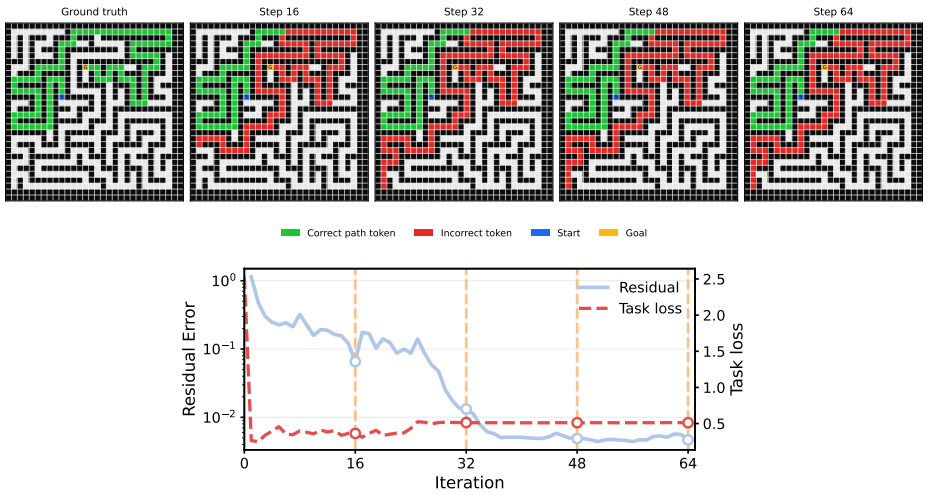


Figure 1: **Local convergence does not guarantee task improvement.** We evaluate TRM (Jolicoeur-Martineau, 2025) on a Maze-Unique (Huang et al., 2026) instance. **Top:** The ground-truth solution and predictions at different iterations. Green and red indicate correct path tokens and incorrect tokens, respectively. The model fails to return to the right path as iterations grow. **Bottom:** The corresponding dynamics of residual and task loss. Residual norm decreases substantially, while the task loss plateaus over later iterations.

couples recurrent states with temporally coordinated denoising objectives (Suleymanzade et al., 2026). Complementary approaches introduce stochasticity into recurrent latent dynamics. PTRM injects Gaussian noise into pretrained TRM rollouts to explore alternative solution basins and uses the existing Q head to select among candidate answers (Sghaier et al., 2026). Similarly, EqR combines randomized initialization with noise along recurrent trajectories to facilitate exploration of the attractor landscape and promote consistent predictions across trajectories (Huang et al., 2026). GRAM explores stochastic recurrent reasoning by learning latent transition distributions through a variational formulation, allowing the model to represent diverse reasoning paths (Baek et al., 2026).

2 PRELIMINARIES

Looped Models and Test-time depth scaling. For an input $\boldsymbol{x}$, a looped model iteratively refines a latent state $\boldsymbol{z} \in \mathbb{R}^d$ with a weight-tied network $F_{\boldsymbol{\theta}}$. Starting from an initial state $\boldsymbol{z}_0$, the recurrence is

$$\boldsymbol{z}_{k+1} = F_{\boldsymbol{\theta}}(\boldsymbol{z}_k; \boldsymbol{x}), \qquad k = 0, 1, \ldots, K-1.$$

A task head h_w maps the final state to a prediction, $\hat{\boldsymbol{y}}_K = h_w(\boldsymbol{z}_K)$. For a target y, we write the predictive loss as $L(\boldsymbol{z}; \boldsymbol{x}, \boldsymbol{y}) = \ell(h_w(\boldsymbol{z}), \boldsymbol{y})$. Parameter sharing allows the inference depth K to vary without changing the number of model parameters, making recurrent depth an axis for test-time computation (Dehghani et al., 2019; Yang et al., 2024; Geiping et al., 2026).

Looped models support *test-time depth scaling* by increasing the number of recurrent iterations at inference. The objective is to improve predictions through additional computation, while *test-time depth efficiency* concerns attaining a given performance level with fewer iterations. Recent work (Prairie et al., 2026; Yang et al., 2026) pursues these goals through stable recurrent dynamics and attractors. However, as illustrated in Figure 1, additional iterations may degrade predictions even as the dynamics converge, a phenomenon commonly termed *overthinking*. In this work, we adopt the attractor-based setting and study recurrent refinement near an attracting fixed point, examining the relationship between residual contraction and task-loss descent.

Stochastic recurrence and noise covariance. To explore alternative reasoning trajectories, recent methods such as EqR (Huang et al., 2026) and GRAM (Baek et al., 2026) introduce stochastic

perturbations into recurrent computation. We express a stochastic update as

$$\boldsymbol{z}_{k+1} = F_{\boldsymbol{\theta}}(\boldsymbol{z}_k; \boldsymbol{x}) + \boldsymbol{\Sigma}^{1/2}(\boldsymbol{z}_k; \boldsymbol{x})\epsilon_k, \qquad \epsilon_k \sim \mathcal{N}(0, \boldsymbol{I}_d),$$

where $\Sigma(z_k; x)$denotes the perturbation covariance. EqR uses isotropic Gaussian noise with covariance $\Sigma_{\mathrm{EqR}} = \beta^2 I_d$. GRAM learns a state-dependent perturbation mean and scalar variance, retaining an isotropic conditional covariance. These approaches motivate investigating the role of noise covariance in task-geometry adaptation, particularly whether its directional structure can improve alignment between the attractor and task landscapes.

Continuous-time formulation. To analyze task-loss evolution under deterministic and stochastic recurrence, we adopt a continuous-time formulation. Define the residual vector field as $R_{\boldsymbol{\theta}}(\boldsymbol{z}; \boldsymbol{x}) = F_{\boldsymbol{\theta}}(\boldsymbol{z}; \boldsymbol{x}) - \boldsymbol{z}$. The associated deterministic dynamics are

$$\mathrm{d}\boldsymbol{z}_t = R_{\boldsymbol{\theta}}(\boldsymbol{z}_t; \boldsymbol{x})\mathrm{d}t. \tag{1}$$

The stochastic recurrence has the continuous-time counterpart

$$\mathrm{d}\boldsymbol{z}_t = R_{\boldsymbol{\theta}}(\boldsymbol{z}_t; \boldsymbol{x})\,\mathrm{d}t + \boldsymbol{\Sigma}^{1/2}(\boldsymbol{z}_t; \boldsymbol{x})\,\mathrm{d}\boldsymbol{W}_t, \tag{2}$$

where $\boldsymbol{W}_t$ is a standard Wiener process. The deterministic and stochastic recurrences correspond to unit-step forward Euler and Euler–Maruyama updates, respectively. An attracting fixed point z^* of the deterministic recurrence satisfies $R_\theta(z^*; x) = 0$ and is therefore an equilibrium. We use this formulation to characterize local residual and task-loss dynamics, and subsequently the effect of noise covariance on task-geometry adaptation.

3 Main Results

In Section 3.1, we formulate this question as a mismatch between the dynamics of the attractor landscape and those of the task landscape. Section 3.2 characterizes how noise covariance shapes task geometry and optimizes a lower bound on the deterministic task-loss descent rate. Section 3.3 turns the optimality condition into a trainable objective for a small noise-prediction network.

3.1 Misalignment between the attractor and task landscapes

To understand when additional recurrent computation improves predictions, we study the local dynamics near an attracting fixed point $\boldsymbol{z}^\star$, taking stability as given. This regime captures a central bottleneck in test-time scaling, where further iterations reduce the fixed point residual without necessarily improving task performance. We first formalize this misalignment between the attractor and task landscapes by showing that residual contraction need not imply task-loss descent. Such misalignment can limit test-time depth scaling and reduce inference-depth efficiency, as additional iterations incur computational costs without corresponding gains in predictive performance. Throughout this section, the input $\boldsymbol{x}$ is fixed, and expectations are taken over the stochastic dynamics. We consider perturbation distributions centered at $\boldsymbol{z}^\star$.

Attractor dynamics. We quantify convergence toward the attractor using the squared-residual objective $\mathcal{R}(\boldsymbol{z}) := \frac{1}{2}\left\|R_{\boldsymbol{\theta}}(\boldsymbol{z}; \boldsymbol{x})\right\|_2^2$. Let $\boldsymbol{J} := \left.\frac{\partial R_{\boldsymbol{\theta}}(\boldsymbol{z}; \boldsymbol{x})}{\partial \boldsymbol{z}}\right|_{\boldsymbol{z}=\boldsymbol{z}^\star}$ denote the residual Jacobian at the fixed point. Assuming sufficient local smoothness, the resulting local dynamic of the expected residual error is:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[\mathcal{R}(\boldsymbol{\xi}_t)] = \frac{1}{2}\mathbb{E}\big[\boldsymbol{\xi}_t^\top \boldsymbol{H}_R \boldsymbol{\xi}_t\big] + \mathcal{O}\big(\mathbb{E}\|\boldsymbol{\xi}_t\|_2^3\big), \qquad \boldsymbol{H}_R = \boldsymbol{J}^\top(\boldsymbol{J} + \boldsymbol{J}^\top)\boldsymbol{J}, \tag{3}$$

where $\boldsymbol{\xi}_t = \boldsymbol{z}_t - \boldsymbol{z}^\star$ denotes the perturbation at time t and $\boldsymbol{H}_R$ characterizes the direction-dependent contribution of the deterministic drift to the residual error. To formalize our focus on the stable regime, we assume that the deterministic drift satisfies a local contraction condition.

Assumption 1 (Local Euclidean contraction)**.** *We assume that the symmetric part of the residual Jacobian is strictly negative definite, i.e.,* $\boldsymbol{J} + \boldsymbol{J}^\top \prec \mathbf{0}$.

Assumption 1 implies that $\boldsymbol{J}$ is nonsingular and hence $\boldsymbol{H}_R \prec \mathbf{0}$. Thus, the deterministic drift strictly decreases the local residual error at every nonzero displacement, whereas noise contributes a nonnegative term that sustains fluctuations around the attractor.

Task-loss dynamics. Let $L(\boldsymbol{z}) := \ell(\boldsymbol{z}; \boldsymbol{x}, y)$ denote the predictive loss and $\Delta L(\boldsymbol{\xi}_t) = L(\boldsymbol{z}^\star + \boldsymbol{\xi}_t) - L(\boldsymbol{z}^\star)$ denote the loss change caused by $\boldsymbol{\xi}_t$. Under the same local dynamics, the resulting dynamic of the expected task-loss drift is

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[\Delta L(\boldsymbol{\xi}_t)] = \frac{1}{2}\mathbb{E}\big[\boldsymbol{\xi}_t^\top \boldsymbol{H}_L \boldsymbol{\xi}_t\big] + \mathcal{O}\big(\mathbb{E}\|\boldsymbol{\xi}_t\|_2^3\big), \qquad \boldsymbol{H}_L = \boldsymbol{J}^\top \boldsymbol{C} + \boldsymbol{C}\boldsymbol{J}, \tag{4}$$

where $\boldsymbol{C} = \nabla_{\boldsymbol{z}}^2 L(\boldsymbol{z}^\star)$ is the task hessian. Similar to $\boldsymbol{H}_R$, $\boldsymbol{H}_L$ characterizes the direction-dependent contribution of the deterministic drift to task-loss change. Unlike $\boldsymbol{H}_R$, $\boldsymbol{H}_L$ is not guaranteed to be negative semidefinite under Assumption 1, even when the task Hessian $\boldsymbol{C}$ is positive semidefinite. The derivations of Eq. (3) and Eq. (4) are deferred to Appendix A.

Proposition 1 (Attractor–task misalignment)**.** *Under Assumption 1, the residual-dynamics operator satisfies $\boldsymbol{H}_R \prec \boldsymbol{0}$, whereas the task-dynamics operator $\boldsymbol{H}_L$ can be indefinite even when $\boldsymbol{C} \succeq \boldsymbol{0}$. Consequently, there can exist arbitrarily small perturbations for which the deterministic drift decreases the local squared residual but increases the local task loss.*

This distinction arises because residual contraction is governed by the Euclidean geometry of the drift, whereas task-loss change depends on how the drift interacts with the task metric $\boldsymbol{C}$. Thus, residual contraction alone does not guarantee a negative expected task-loss drift rate. This motivates examining how noisy training reshapes the task metric and aligns it with the residual dynamics.

3.2 Noise covariance as landscape design

We next examine how the injected noise covariance shapes local task geometry and, through this adaptation, influences task-loss dynamics. Throughout this subsection, we fix the input $\boldsymbol{x}$, the recurrent dynamics $F_{\boldsymbol{\theta}}$, and the attracting fixed point $\boldsymbol{z}^\star$, so that the residual Jacobian $\boldsymbol{J}$ remains fixed. We study stationary exploration as an idealized description of the states visited during late-iteration noisy training.

Lemma 1 (Stationary exploration covariance (Särkkä & Solin, 2019))**.** *Consider the local linear dynamics $\mathrm{d}\boldsymbol{\xi}_t = \boldsymbol{J}\boldsymbol{\xi}_t\,\mathrm{d}t + \boldsymbol{\Sigma}^{1/2}\mathrm{d}\boldsymbol{W}_t$, with constant $\boldsymbol{\Sigma} \succeq \boldsymbol{0}$ and centered initial perturbations of finite second moment. Define $\boldsymbol{P}_t := \mathbb{E}[\boldsymbol{\xi}_t\boldsymbol{\xi}_t^\top]$. Under Assumption 1, $\boldsymbol{P}_t$ converges to the unique stationary covariance $\boldsymbol{P}_\infty$ satisfying*

$$\boldsymbol{J}\boldsymbol{P}_\infty + \boldsymbol{P}_\infty\boldsymbol{J}^\top + \boldsymbol{\Sigma} = \boldsymbol{0}. \tag{5}$$

Lemma 1 characterizes how the recurrent dynamics transform the injected noise covariance $\boldsymbol{\Sigma}$ into the stationary exploration covariance $\boldsymbol{P}_\infty$. During noisy training, the task head adapts to this exploration distribution. Characterizing this adaptation directly is challenging because the expected task loss depends on both the explored states and the head parameters. To expose how exploration enters the training objective, we use the local task-loss expansion

$$\mathbb{E}\left[L(\boldsymbol{z}^\star + \boldsymbol{\xi}_\infty)\right] = L(\boldsymbol{z}^\star) + \frac{1}{2}\operatorname{Tr}(\boldsymbol{C}\boldsymbol{P}_\infty) + \mathcal{O}\left(\mathbb{E}\|\boldsymbol{\xi}_\infty\|_2^3\right). \tag{6}$$

Together, Eq. (5) and Eq. (6) reveal a local mechanism linking noise injection to task geometry: $\boldsymbol{\Sigma}$ determines $\boldsymbol{P}_\infty$ through the recurrent dynamics, while $\boldsymbol{P}_\infty$ weights the curvature penalty imposed on $\boldsymbol{C}$ during training. The resulting adaptation of $\boldsymbol{C}$ changes the task-dynamics operator $\boldsymbol{H}_L$ in Eq. (4). We therefore formulate covariance design by accounting for how the task head adapts to the induced stationary exploration distribution.

Covariance design through task adaptation. To investigate task-geometry alignment under noisy training, we model the task head's adaptation to the injected noise. Let w denote the task-head parameters, write L_w for the corresponding task loss, and define $\boldsymbol{C}_w := \nabla_{\boldsymbol{z}}^2 L_w(\boldsymbol{z}^\star)$. We model the adapted task-head parameters as a function of the noise covariance $\boldsymbol{\Sigma}$:

$$w^\star(\boldsymbol{\Sigma}) \in \arg\min_w \ \mathbb{E}_{\boldsymbol{\xi}}\left[L_w(\boldsymbol{z}^\star + \boldsymbol{\xi})\right], \tag{7}$$

where the expectation is over stationary perturbations induced by $\boldsymbol{\Sigma}$.

To improve inference-depth efficiency, we seek a noise covariance whose induced task geometry supports greater task-loss reduction during subsequent recurrent computation. As a local design criterion, we consider the drift contribution to expected quadratic task-loss decrease:

$$-\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[\Delta L_{w^\star(\boldsymbol{\Sigma})}] = \frac{1}{2}\operatorname{Tr}\big(-\boldsymbol{H}_{L,w^\star(\boldsymbol{\Sigma})}\boldsymbol{P}_t\big) \geq \frac{\lambda_{\min}\big(-\boldsymbol{H}_{L,w^\star(\boldsymbol{\Sigma})}\big)}{2}\mathbb{E}\|\boldsymbol{\xi}_t\|_2^2. \tag{8}$$

Maximizing the lower-bound coefficient improves the weakest directional descent contribution. Under a prescribed noise budget $\tau > 0$, this yields the bi-level formulation

$$\begin{aligned} \max_{\boldsymbol{\Sigma} \succeq \mathbf{0}} \quad & \lambda_{\min}\big(-\boldsymbol{H}_{L,w^\star(\boldsymbol{\Sigma})}\big) \\ \text{s.t.} \quad & \operatorname{Tr}(\boldsymbol{\Sigma}) = \tau, \\ & w^\star(\boldsymbol{\Sigma}) \in \arg\min_{w} \; \mathbb{E}_{\boldsymbol{\xi}} \left[L_w(\boldsymbol{z}^\star + \boldsymbol{\xi}) \right]. \end{aligned} \tag{9}$$

The inner problem studies noise-induced task alignment to promote test-time depth scaling, while the outer problem designs the covariance to improve inference-depth efficiency. Directly solving this bi-level problem is challenging as evaluating each covariance requires determining the corresponding adapted head. To make this formulation tractable, we therefore take two steps: (1) introduce a metric-level surrogate for the inner problem and (2) use its solution to characterize the covariance.

Inner problem: task-geometry adaptation. Using Eq. (6), the inner training objective has the second-order approximation

$$L_w(\boldsymbol{z}^\star) + \frac{1}{2} \operatorname{Tr}(\boldsymbol{C}_w \boldsymbol{P}_\infty(\boldsymbol{\Sigma})). \tag{10}$$

Both terms depend on the head parameters, coupling prediction at the attractor with sensitivity to the explored perturbations. We approximate the inner problem by minimizing the curvature penalty over positive-definite metrics with a common determinant. This fixed-volume constraint serves as a scale normalization, giving the metric-level surrogate

$$\boldsymbol{C}^\star(\boldsymbol{\Sigma}) := \arg\min_{\boldsymbol{C} \succ \mathbf{0}} \operatorname{Tr}(\boldsymbol{C} \boldsymbol{P}_\infty(\boldsymbol{\Sigma})) \quad \text{s.t.} \quad \log\det \boldsymbol{C} = \rho. \tag{11}$$

Theorem 1 (Task-geometry alignment under noisy adaptation)**.** *Let $\boldsymbol{P}_\infty(\boldsymbol{\Sigma}) \succ \mathbf{0}$. The unique solution of Eq. (11) satisfies $\boldsymbol{C}^\star(\boldsymbol{\Sigma}) \propto \boldsymbol{P}_\infty(\boldsymbol{\Sigma})^{-1}$. The induced task-dynamics operator $\boldsymbol{H}_L^\star(\boldsymbol{\Sigma}) := \boldsymbol{J}^\top \boldsymbol{C}^\star(\boldsymbol{\Sigma}) + \boldsymbol{C}^\star(\boldsymbol{\Sigma}) \boldsymbol{J}$ satisfies*

$$\boldsymbol{\Sigma} \succeq \mathbf{0} \implies \boldsymbol{H}_L^\star(\boldsymbol{\Sigma}) \preceq \mathbf{0}, \tag{12}$$

with strict inequality whenever $\boldsymbol{\Sigma} \succ \mathbf{0}$.

Theorem 1 characterizes the inner response in the bi-level problem. It reveals that adaptation to noise-induced perturbations in the stationary regime yields a task metric aligned with the recurrent dynamics. This result suggests that noisy training promotes test-time depth scaling by reshaping the task geometry to support additional recurrent computation. Proofs are deferred to Appendix B.

Outer problem: noise covariance design. Theorem 1 provides the metric response $\boldsymbol{C}^\star(\boldsymbol{\Sigma})$ for the surrogate inner problem. Substituting this response into the outer objective of Eq. (9) gives

$$\max_{\boldsymbol{\Sigma} \succeq \mathbf{0}} \quad \lambda_{\min}(-\boldsymbol{H}_L^\star(\boldsymbol{\Sigma})) \quad \text{s.t.} \quad \operatorname{Tr}(\boldsymbol{\Sigma}) = \tau, \; \boldsymbol{P}_\infty(\boldsymbol{\Sigma}) \succ \mathbf{0}. \tag{13}$$

Theorem 2 (Covariance design for task-loss descent)**.** *Under Assumption 1, the covariance-design problem in Eq. (13) has the same unique optimal covariance as*

$$\min_{\boldsymbol{\Sigma} \succ \mathbf{0}, \, \gamma > 0} \quad \left\| \boldsymbol{J} \boldsymbol{\Sigma}^{1/2} + \boldsymbol{\Sigma}^{1/2} \boldsymbol{J}^\top + \gamma \boldsymbol{\Sigma} \right\|_{\mathrm{F}}^2 \quad \text{s.t.} \quad \operatorname{Tr}(\boldsymbol{\Sigma}) = \tau. \tag{14}$$

Theorem 2 characterizes the solution to the surrogate outer problem through a matrix-residual objective. It reveals that the covariance geometry is determined by the local residual Jacobian, establishing a connection between the directional structure of the injected noise and task-loss descent through the adapted task metric. Proofs are deferred to Appendix C. Together with Theorem 1, this suggests that structured noise can both promote test-time depth scaling and improve inference-depth efficiency, allowing additional recurrent computation to yield greater task benefit. Moreover, this matrix-residual characterization enables efficient covariance learning without explicitly solving the inner metric problem, providing the basis for the noise-network objective in the next subsection.

Algorithm 1 Training

Input: Dataset $\mathcal{D}$, models $F_{\boldsymbol{\theta}}, g_{\boldsymbol{\phi}}$, iterations n, threshold K, budget τ

repeat
  Sample a minibatch $\mathcal{B} = \{(\boldsymbol{x}_i, y_i)\}$
  Run the first K iterations to obtain $\boldsymbol{z}_{K,i}$
  for $t = K, \ldots, n-1$ **do**
    Sample $\mathbf{v}_{t,i} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$
    $\boldsymbol{e}_{t,i} \leftarrow g_{\boldsymbol{\phi}}(\boldsymbol{x}_i, \boldsymbol{z}_{t,i}, \mathbf{v}_{t,i})$
    $\boldsymbol{z}_{t+1,i} \leftarrow F_{\boldsymbol{\theta}}(\boldsymbol{z}_{t,i}; \boldsymbol{x}_i) + \mathrm{sg}(\boldsymbol{e}_{t,i})$
  end for
  Compute γ, $\mathcal{L}_{\text{task}}$ and $\mathcal{L}_{\text{cov}}$
  Update $\boldsymbol{\theta}$ using $\mathcal{L}_{\text{task}}$ and $\boldsymbol{\phi}$ using $\mathcal{L}_{\text{cov}}$
until converged
return $F_{\boldsymbol{\theta}}, g_{\boldsymbol{\phi}}$

Algorithm 2 Inference with Breadth Scaling

Input: Input $\boldsymbol{x}$, trained models $F_{\boldsymbol{\theta}}, g_{\boldsymbol{\phi}}$, iterations n, threshold K, budget τ, trajectories B

Sample B independent initial states $\{\boldsymbol{z}_0^{(b)}\}_{b=1}^B$
Run the first K iterations to obtain $\{\boldsymbol{z}_K^{(b)}\}_{b=1}^B$
for $t = K, \ldots, n-1$ **do**
  Sample independent $\mathbf{v}_t^{(b)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ for all b
  $\boldsymbol{e}_t^{(b)} \leftarrow g_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}_t^{(b)}, \mathbf{v}_t^{(b)})$
  $\boldsymbol{z}_{t+1}^{(b)} \leftarrow F_{\boldsymbol{\theta}}(\boldsymbol{z}_t^{(b)}; \boldsymbol{x}) + \boldsymbol{e}_t^{(b)}$
end for
Decode $\{\boldsymbol{z}_n^{(b)}\}_{b=1}^B$ into predictions $\{\hat{y}^{(b)}\}_{b=1}^B$
$\hat{y} \leftarrow \mathrm{majority_voting}(\{\hat{y}^{(b)}\}_{b=1}^B)$
return $\hat{y}$

3.3 Efficient learning of structured noise

Theorem 2 characterizes the optimal covariance through a matrix-residual minimization problem. However, directly evaluating this objective requires explicitly forming $\boldsymbol{J}$, which is impractical for high-dimensional looped models. Moreover, the analysis assumes a fixed input $\boldsymbol{x}$, so applying the result directly would require solving a separate covariance-design problem for each input. To make covariance design practical and scalable, we propose *Noise Designed Looped Models* (NDLM), which uses a lightweight network to generate input-dependent noise perturbations without explicitly constructing a covariance matrix for each input.

Training target. Given an input $\boldsymbol{x}$, its corresponding latent state $\boldsymbol{z}$, and a standard Gaussian sample $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$, we aim to learn a network $g_{\boldsymbol{\phi}}$ that generates noise perturbations satisfying

$$g_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}, \mathbf{v}) = \boldsymbol{\Sigma}^\star(\boldsymbol{x})^{1/2}\mathbf{v}, \tag{15}$$

where $\boldsymbol{\Sigma}^\star(\boldsymbol{x})$ denotes the optimal covariance characterized by Theorem 2 for input $\boldsymbol{x}$. Rather than explicitly computing this target, we train the network through the optimality condition in Eq. (14). For brevity, write $g_{\boldsymbol{\phi}}^{\boldsymbol{x}}(\mathbf{v}) := g_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}, \mathbf{v})$. Under the symmetric parameterization described below, this yields the training objective

$$\mathcal{L}_{\text{cov}}(\boldsymbol{\phi}) := \mathbb{E}_{\boldsymbol{x},\mathbf{v}}\left[\left(2\left\langle \boldsymbol{J}(\boldsymbol{x})^\top \mathbf{v}, g_{\boldsymbol{\phi}}^{\boldsymbol{x}}(\mathbf{v})\right\rangle + \gamma \left\|g_{\boldsymbol{\phi}}^{\boldsymbol{x}}(\mathbf{v})\right\|_2^2\right)^2\right], \tag{16}$$

where the expectation is over training inputs and independent Gaussian probes, and $\boldsymbol{J}(\boldsymbol{x})^\top \mathbf{v}$ can be efficiently computed using a reverse-mode vector–Jacobian product. We adaptively compute a batch-specific γ by solving a scalar least-squares problem to minimize Eq. (16). Motivated by the local analysis, we learn the perturbations only during the final $n - K$ of the n total iterations, using a predefined threshold K to target later recurrent states. This schedule is a practical approximation to selecting states near an attractor.

Practical Implementation. To facilitate learning, we parameterize the noise map using a conditioned symmetric-attention network whose action on the Gaussian input is linear, symmetric, and positive definite. The network generates the perturbation

$$g_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}, \mathbf{v}) = \left(\boldsymbol{I} + h_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}) h_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})^\top\right)\mathbf{v}, \tag{17}$$

where $h_{\boldsymbol{\phi}}$ is a learnable network. This shared projection produces a symmetric positive-semidefinite Gram matrix, while the identity term ensures positive definiteness. Furthermore, to approximately enforce the trace budget $\mathrm{Tr}(\boldsymbol{\Sigma}) = \tau$, we explicitly normalize the generated noise samples. More implementation details are deferred to Appendix D. We summarize the training and inference procedures in Algorithms 1 and 2, respectively.

Table 1: Peak exact accuracy and model size on Sudoku-Extreme, Maze-Hard, and Maze-Unique. Results marked with (*) are obtained from our reimplementation. All other results are sourced from prior works (Helbling et al., 2026; Suleymanzade et al., 2026; Huang et al., 2026). "–" indicates an unreported result.

	Sudoku-Extreme		Maze-Unique		Maze-Hard	
Method	**Acc. (%)**	**Size**	**Acc. (%)**	**Size**	**Acc. (%)**	**Size**
Flow-based						
MDLM	3.9	30M	89.0	8M	–	–
FLM	13.9	7M	93.3	8M	–	–
Adaptive MDLM	19.1	30M	100.0	8M	–	–
Flow Reasoning Model	99.5	7M	99.9	8M	–	–
Looped Flows	99.3	5M	–	–	86.9	7M
Attractor-based						
HRM	55.0	27M	0.3	27M	74.5	27M
TRM	87.4	5M	86.5*	0.26M	85.3	7M
FPRM	94.2	7M	100.0	7M	87.0	7M
GRAM	97.0	10M	–	–	–	–
EqR	99.8	5M	97.1*	0.26M	83.6*	7M
NDLM	99.4	5.55M	99.0	0.30M	86.5	7.35M

4 EXPERIMENTS

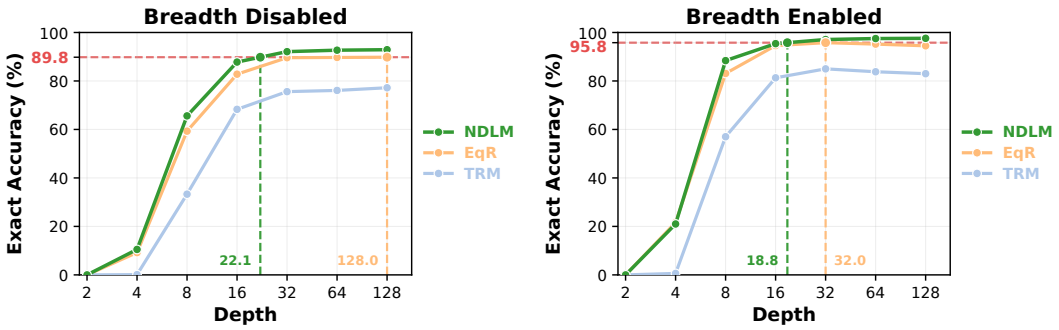
In this section, we evaluate NDLM on different reasoning datasets. We consider three structured reasoning benchmarks. Sudoku-Extreme (Wang et al., 2025) contains challenging 9×9 Sudoku puzzles. Maze-Hard (Wang et al., 2025) contains 30×30 mazes with long solution paths, while Maze-Unique (Huang et al., 2026) contains 30×30 mazes with unique solution paths. We report exact accuracy on all benchmarks. Our experimental setup follows TRM (Jolicoeur-Martineau, 2025) and EqR (Huang et al., 2026). Further details are provided in Appendix D.

Experimental results. We compare NDLM against a broad range of reasoning baselines from two families. For flow-based methods, we consider MDLM (Sahoo et al., 2024), FLM (Lee et al., 2026), Adaptive MDLM (Kim et al., 2025), Flow Reasoning Model (FRM) (Helbling et al., 2026), and Looped Flows (Suleymanzade et al., 2026). For attractor-based methods, we consider HRM (Wang et al., 2025), TRM (Jolicoeur-Martineau, 2025), FPRM (Movahedi et al., 2026), GRAM (Baek et al., 2026), and EqR (Huang et al., 2026). We adopt the same iterations n and breadth B as EqR to ensure a fair comparison. The detailed results are presented in Table 1.

As shown in Table 1, NDLM achieve the second-best results across all three datasets among attractor-based models. Remarkably, compared to EqR, NDLM achieve an 1.9% point performance gain on Maze-Unique and a 2.9% point gain on Maze-Hard, while maintaining competitive performance on Sudoku-Extreme. These results suggest that our designed noise can provide greater predictive benefits than isotropic noise by adapting the perturbation structure to the local dynamics.

NDLM enables efficient test-time depth scaling. We investigate on Maze-Unique how structured perturbations affect the relationship between recurrent convergence and prediction, and whether these effects translate into improved depth scaling. Our controlled comparisons include TRM (Jolicoeur-Martineau, 2025) and EqR (Huang et al., 2026), which correspond to training without injected noise and training with isotropic Gaussian noise, respectively.

Figure 2 shows that NDLM benefits from additional recurrent computation both with and without breadth scaling. Without breadth scaling, its exact accuracy increases from 65.6% at depth 8 to 92.9% at depth 128. With breadth scaling enabled, accuracy rises from 88.4% to 97.6% over the same depth range. These results demonstrate that the benefits of structured perturbations extend to increasing recurrent depth, including when multiple trajectories are available.



(a) Breadth scaling disabled

(b) Breadth scaling enabled

Figure 2: **NDLM enables efficient test-time depth scaling without observed overthinking.** Exact accuracy as a function of recurrent depth for the Maze-Unique dataset, with breadth scaling disabled (left) and enabled (right) with $B = 8$.

NDLM also achieves these gains with substantially fewer iterations than EqR. As shown in Figure 2a, NDLM reaches the 89.8% accuracy attained by EqR at depth 128 with an estimated depth of only 22.1, reducing the required depth by approximately 82.7%. With breadth scaling enabled, NDLM reaches EqR's peak accuracy of 95.8% at an estimated depth of 18.8, compared with 32 for EqR, a reduction of approximately 41.3%. Thus, structured noise improves inference-depth efficiency as well as the accuracy attainable with additional computation.

Moreover, Figure 2b reveals overthinking in both EqR and TRM, whose accuracy peaks at depth 32 and subsequently declines. In particular, EqR's accuracy decreases from 95.8% to 94.5% as depth increases from 32 to 128. In contrast, NDLM maintains non-decreasing accuracy across the evaluated depths, with no observed overthinking in this range. Together, these results suggest that our designed perturbations support more effective recurrent refinement than isotropic noise, yielding predictive gains with fewer iterations and preserving those gains at greater depths.

Breadth scaling for NDLM. We evaluate NDLM on Sudoku-Lite (a subset of 2,048 puzzles sampled from Sudoku-Extreme (Huang et al., 2026)), Maze-Unique, and Maze-Hard at different breadth scales, keeping recurrent depth fixed. We aggregate predictions from B independent trajectories through majority voting. As shown in Table 2, accuracy improves with increasing breadth across all three datasets. These results indicate that NDLM benefits from additional parallel trajectories, providing a complementary axis for scaling test-time computation beyond recurrent depth.

Table 2: NDLM exact accuracy (%) at different breadth scales with recurrent depth fixed.

B	Sudoku-Lite	Maze-Unique	Maze-Hard
1	84.8	87.8	85.5
4	87.9	92.2	86.0
16	94.5	97.6	86.3
64	97.4	98.4	86.5

5 Conclusion

In this paper, we study how additional recurrent computation improves predictions in attractor-based models, taking local stability as given. We formalize attractor–task misalignment, showing that residual contraction need not imply task-loss descent. Under a local surrogate, we characterize how noise covariance shapes stationary exploration and task-geometry adaptation to support predictive improvement. We propose Noise Designed Looped Models (NDLM), which optimize a lower bound on deterministic task-loss descent through an equivalent matrix-residual objective learned by a lightweight network. Experiments on Sudoku-Extreme, Maze-Unique, and Maze-Hard demonstrate competitive performance and efficient test-time depth scaling. On Maze-Unique, NDLM matches the isotropic-noise baseline's reference accuracy with an estimated 82.7% reduction in recurrent depth, while retaining gains from additional iterations with no observed overthinking across the evaluated depths, including with breadth scaling. These findings highlight noise covariance design as a means of improving the predictive benefits and efficiency of recurrent computation.

AI USE STATEMENT

Generative AI tools were used to assist with the initial drafting and LaTeX editing of this manuscript. The authors reviewed the mathematical derivations, citations, claims, and generated text and take responsibility for the final content of the work.

REFERENCES

Junyeob Baek, Mingyu Jo, Minsu Kim, Mengye Ren, Yoshua Bengio, and Sungjin Ahn. Generative recursive reasoning. *arXiv preprint arXiv:2605.19376*, 2026.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in neural information processing systems*, 32, 2019.

Hugh Blayney, Álvaro Arroyo, Johan Obando-Ceron, Pablo Samuel Castro, Aaron Courville, Michael M Bronstein, and Xiaowen Dong. A mechanistic analysis of looped reasoning language models. *arXiv preprint arXiv:2604.11791*, 2026.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyzdRiR9Y7.

Yilun Du, Jiayuan Mao, and Joshua B. Tenenbaum. Learning iterative reasoning through energy diffusion. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=CduFAALvGe.

Jacob Fein-Ashley and Paria Rashidinejad. Solve the loop: Attractor models for language and reasoning. *arXiv preprint arXiv:2605.12466*, 2026.

Tianyu Fu, Yichen You, Zekai Chen, Guohao Dai, Huazhong Yang, and Yu Wang. Think-at-hard: Selective latent iterations to improve reasoning language models. *arXiv preprint arXiv:2511.08577*, 2025.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *Advances in Neural Information Processing Systems*, 38: 41340–41391, 2026.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *International Conference on Machine Learning*, pp. 11398–11442. PMLR, 2023.

Alec Helbling, Andrey Bryutkin, Mauro Martino, Nima Dehmamy, and Hendrik Strobelt. Flow reasoning models: Scaling reasoning through iterative self-refinement. *arXiv preprint arXiv:2606.29150*, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Benhao Huang, Zhengyang Geng, and J Zico Kolter. Equilibrium reasoners: Learning attractors enables scalable reasoning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=lh95PnOlpM.

Ahmadreza Jeddi, Marco Ciccone, and Babak Taati. Loopformer: Elastic-depth looped transformers for latent reasoning via shortcut modulation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RzYXb5YWBs.

Alexia Jolicoeur-Martineau. Less is more: Recursive reasoning with tiny networks. *arXiv preprint arXiv:2510.04871*, 2025.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M. Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DjJmre5IkP.

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M Boffi, and Jinwoo Kim. Flow map language models: One-step language modeling via continuous denoising. *arXiv preprint arXiv:2602.16813*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Sean Michael McLeish, Ang Li, John Kirchenbauer, Dayal Singh Kalra, Brian R. Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Jonas Geiping, Tom Goldstein, and Micah Goldblum. Teaching pretrained language models to think deeper with retrofitted recurrence. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=PXVQTHYwgt.

Sajad Movahedi, Vera Milovanović, Shlomo Libo Feigin, Alexander Theus, Thomas Hofmann, Valentina Boeva, T Konstantin Rusch, and Antonio Orvieto. Fixed-point reasoners: Stable and adaptive deep looped transformers. *arXiv preprint arXiv:2606.18206*, 2026.

Hayden Prairie, Zachary Novack, Taylor Berg-Kirkpatrick, and Daniel Y Fu. Parcae: Scaling laws for stable looped language models. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=pIBqtrqFeP.

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

Simo Särkkä and Arno Solin. Applied stochastic differential equations. *Institute of mathematical statistics textbooks*, 2019.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *International Conference on Learning Representations*, volume 2025, pp. 14855–14881, 2025.

Amin Sghaier, Ali Parviz, and Alexia Jolicoeur-Martineau. Probabilistic tiny recursive model. *arXiv preprint arXiv:2605.19943*, 2026.

Ayhan Suleymanzade, Chanhyuk Lee, Floor Eijkelboom, Nicholas M Boffi, İsmail İlkan Ceylan, and Jinwoo Kim. Thinking with looped flows. *arXiv preprint arXiv:2609.11801*, 2026.

Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical reasoning model. *arXiv preprint arXiv:2506.21734*, 2025.

Guanghan Wang, Yair Schiff, Subham Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *Advances in Neural Information Processing Systems*, 38:147282–147339, 2026.

Liu Yang, Kangwook Lee, Robert Nowak, and Dimitris Papailiopoulos. Looped transformers are better at learning learning algorithms. In *International conference on learning representations*, volume 2024, pp. 42195–42214, 2024.

Xiao-Wen Yang, Ziyu Han, Xi-Hua Zhang, Wen-Da Wei, Jie-Jing Shao, Lan-Zhe Guo, and Yu-Feng Li. Stabilizing recurrent dynamics for test-time scalable latent reasoning in looped language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=tbdSfBMlbq.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, Fan Yin, He Xing, et al. Scaling latent reasoning via looped language models. *arXiv preprint arXiv:2510.25741*, 2025.

A Derivations of Local Dynamics

We derive the local residual and task-loss dynamics used in Eq. (3) and Eq. (4). Throughout this section, we focus on the contribution of the deterministic drift under the local linearization around an attracting fixed point $\boldsymbol{z}^\star$ which satisfies $\boldsymbol{R}_\theta(\boldsymbol{z}^\star; \boldsymbol{x}) = 0$.

The residual vector field admits the local expansion

$$\boldsymbol{R}_\theta(\boldsymbol{z}^\star + \boldsymbol{\xi}; \boldsymbol{x}) = \boldsymbol{J}\boldsymbol{\xi} + \mathcal{O}(\|\boldsymbol{\xi}\|_2^2). \tag{18}$$

Accordingly, the deterministic perturbation dynamics satisfy

$$\dot{\boldsymbol{\xi}}_t = \boldsymbol{J}\boldsymbol{\xi}_t + \mathcal{O}(\|\boldsymbol{\xi}_t\|_2^2).$$

Residual dynamics. Recall the squared-residual objective

$$\mathcal{R}(\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{R}_\theta(\boldsymbol{z}; \boldsymbol{x})\|_2^2.$$

Using Eq. (18), its local expansion around $\boldsymbol{z}^\star$ is

$$\mathcal{R}(\boldsymbol{z}^\star + \boldsymbol{\xi}) = \frac{1}{2}\boldsymbol{\xi}^\top \boldsymbol{J}^\top \boldsymbol{J}\boldsymbol{\xi} + \mathcal{O}(\|\boldsymbol{\xi}\|_2^3).$$

Therefore,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{R}(\boldsymbol{z}^\star + \boldsymbol{\xi}_t) &= \boldsymbol{\xi}_t^\top \boldsymbol{J}^\top \boldsymbol{J}\dot{\boldsymbol{\xi}}_t + \mathcal{O}(\|\boldsymbol{\xi}_t\|_2^3) \\ &= \boldsymbol{\xi}_t^\top \boldsymbol{J}^\top \boldsymbol{J}\boldsymbol{J}\boldsymbol{\xi}_t + \mathcal{O}(\|\boldsymbol{\xi}_t\|_2^3). \end{aligned} \tag{19}$$

Since the quantity in Eq. (19) is scalar, it can be symmetrized as

$$\begin{aligned} \boldsymbol{\xi}_t^\top \boldsymbol{J}^\top \boldsymbol{J}\boldsymbol{J}\boldsymbol{\xi}_t &= \frac{1}{2}\boldsymbol{\xi}_t^\top \left[\boldsymbol{J}^\top \boldsymbol{J}^2 + (\boldsymbol{J}^\top \boldsymbol{J}^2)^\top\right] \boldsymbol{\xi}_t \\ &= \frac{1}{2}\boldsymbol{\xi}_t^\top \boldsymbol{J}^\top (\boldsymbol{J} + \boldsymbol{J}^\top)\boldsymbol{J}\boldsymbol{\xi}_t. \end{aligned}$$

Defining

$$\boldsymbol{H}_R := \boldsymbol{J}^\top (\boldsymbol{J} + \boldsymbol{J}^\top)\boldsymbol{J},$$

and taking expectations gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[\mathcal{R}(\boldsymbol{\xi}_t)] = \frac{1}{2}\mathbb{E}\big[\boldsymbol{\xi}_t^\top \boldsymbol{H}_R \boldsymbol{\xi}_t\big] + \mathcal{O}\big(\mathbb{E}\|\boldsymbol{\xi}_t\|_2^3\big)\,,$$

which yields Eq. (3).

Task-loss dynamics. Let

$$L(\boldsymbol{z}) := \ell(\boldsymbol{z}; \boldsymbol{x}, \boldsymbol{y}), \qquad \Delta L(\boldsymbol{\xi}) := L(\boldsymbol{z}^\star + \boldsymbol{\xi}) - L(\boldsymbol{z}^\star),$$

and define

$$\boldsymbol{g} := \nabla_{\boldsymbol{z}} L(\boldsymbol{z}^\star), \qquad \boldsymbol{C} := \nabla_{\boldsymbol{z}}^2 L(\boldsymbol{z}^\star).$$

A second-order Taylor expansion gives

$$\Delta L(\boldsymbol{\xi}) = \boldsymbol{g}^\top \boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\xi}^\top \boldsymbol{C}\boldsymbol{\xi} + \mathcal{O}(\|\boldsymbol{\xi}\|_2^3).$$

Recall that we consider centered perturbations $\mathbb{E}[\boldsymbol{\xi}_t] = 0$, which are preserved by the locally linearized dynamics under zero-mean additive perturbations. Hence the deterministic drift contribution to the expected task-loss dynamics is therefore

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[\Delta L(\boldsymbol{\xi}_t)] = \mathbb{E}\big[\boldsymbol{\xi}_t^\top \boldsymbol{C}\boldsymbol{J}\boldsymbol{\xi}_t\big] + \mathcal{O}\big(\mathbb{E}\|\boldsymbol{\xi}_t\|_2^3\big)\,.$$

Again using the fact that the quadratic form is scalar,

$$\boldsymbol{\xi}_t^\top \boldsymbol{C}\boldsymbol{J}\boldsymbol{\xi}_t = \frac{1}{2}\boldsymbol{\xi}_t^\top \left(\boldsymbol{C}\boldsymbol{J} + \boldsymbol{J}^\top \boldsymbol{C}\right) \boldsymbol{\xi}_t.$$

Defining

$$\boldsymbol{H}_L := \boldsymbol{J}^\top \boldsymbol{C} + \boldsymbol{C}\boldsymbol{J},$$

we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[\Delta L(\boldsymbol{\xi}_t)] = \frac{1}{2}\mathbb{E}\big[\boldsymbol{\xi}_t^\top \boldsymbol{H}_L \boldsymbol{\xi}_t\big] + \mathcal{O}\big(\mathbb{E}\|\boldsymbol{\xi}_t\|_2^3\big)\,,$$

which gives Eq. (4)

B Proof of Theorem 1

Proof. Let $\boldsymbol{P} := \boldsymbol{P}_\infty \succ \mathbf{0}$. For any feasible $\boldsymbol{C} \succ \mathbf{0}$ satisfying $\det \boldsymbol{C} = e^\rho$, the arithmetic–geometric mean inequality applied to the eigenvalues of $\boldsymbol{P}^{1/2}\boldsymbol{C}\boldsymbol{P}^{1/2}$ gives

$$\begin{aligned}\operatorname{Tr}(\boldsymbol{C}\boldsymbol{P}) &= \operatorname{Tr}\Big(\boldsymbol{P}^{1/2}\boldsymbol{C}\boldsymbol{P}^{1/2}\Big) \\ &\geq d\left[\det\Big(\boldsymbol{P}^{1/2}\boldsymbol{C}\boldsymbol{P}^{1/2}\Big)\right]^{1/d} \\ &= d\, e^{\rho/d}(\det \boldsymbol{P})^{1/d}.\end{aligned}$$

Equality holds if and only if all eigenvalues of $\boldsymbol{P}^{1/2}\boldsymbol{C}\boldsymbol{P}^{1/2}$ are equal, or equivalently,

$$\boldsymbol{P}^{1/2}\boldsymbol{C}\boldsymbol{P}^{1/2} = \alpha \boldsymbol{I}_d, \qquad \alpha := e^{\rho/d}(\det \boldsymbol{P})^{1/d} > 0.$$

Consequently, the unique global minimizer of Eq. (11) is

$$\boldsymbol{C}^\star = \alpha \boldsymbol{P}^{-1}. \tag{20}$$

This matrix is positive definite and satisfies $\det \boldsymbol{C}^\star = e^\rho$, establishing both feasibility and uniqueness.

It remains to characterize the induced task-dynamics operator. The stationary covariance satisfies Eq. (5),

$$\boldsymbol{J}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{J}^\top + \boldsymbol{\Sigma} = \mathbf{0}.$$

Multiplying on the left and right by $\boldsymbol{P}^{-1}$ yields

$$\boldsymbol{P}^{-1}\boldsymbol{J} + \boldsymbol{J}^\top \boldsymbol{P}^{-1} = -\boldsymbol{P}^{-1}\boldsymbol{\Sigma}\boldsymbol{P}^{-1}.$$

Substituting Eq. (20) therefore gives

$$\boldsymbol{H}_L^\star = \boldsymbol{J}^\top \boldsymbol{C}^\star + \boldsymbol{C}^\star \boldsymbol{J} = -\alpha \boldsymbol{P}^{-1}\boldsymbol{\Sigma}\boldsymbol{P}^{-1}.$$

For any $\boldsymbol{v} \in \mathbb{R}^d$,

$$\boldsymbol{v}^\top \boldsymbol{H}_L^\star \boldsymbol{v} = -\alpha(\boldsymbol{P}^{-1}\boldsymbol{v})^\top \boldsymbol{\Sigma}(\boldsymbol{P}^{-1}\boldsymbol{v}).$$

Thus, $\boldsymbol{\Sigma} \succeq \mathbf{0}$ implies $\boldsymbol{H}_L^\star \preceq \mathbf{0}$. If $\boldsymbol{\Sigma} \succ \mathbf{0}$, then the right-hand side is strictly negative for every nonzero $\boldsymbol{v}$, since $\boldsymbol{P}^{-1}$ is invertible. Hence $\boldsymbol{H}_L^\star \prec \mathbf{0}$. □

C Proof of Theorem 2

Proof. Assumption 1 implies that $\boldsymbol{J}$ is Hurwitz. Therefore, the Lyapunov equation

$$\boldsymbol{J}^\top \boldsymbol{M} + \boldsymbol{M}\boldsymbol{J} = -\boldsymbol{I}_d \tag{21}$$

has a unique positive-definite solution

$$\boldsymbol{M} = \int_0^\infty e^{\boldsymbol{J}^\top t} e^{\boldsymbol{J}t}\, \mathrm{d}t \succ \mathbf{0}.$$

Define

$$\kappa^\star := \left(\frac{e^\rho}{\det \boldsymbol{M}}\right)^{1/d}, \qquad s := \frac{\tau}{\operatorname{Tr}(\boldsymbol{M}^{-2})}.$$

Both scalars are strictly positive.

We first establish an upper bound on the worst-direction objective. By Theorem 1, every metric induced by a feasible covariance is positive definite and satisfies $\det \boldsymbol{C} = e^\rho$. We therefore first upper-bound the objective over this larger class of admissible metrics.

For any $\boldsymbol{C} \succ \mathbf{0}$ satisfying $\det \boldsymbol{C} = e^\rho$, define

$$\boldsymbol{Q} := -\boldsymbol{J}^\top \boldsymbol{C} - \boldsymbol{C}\boldsymbol{J}, \qquad \kappa := \lambda_{\min}(\boldsymbol{Q}).$$

If $\kappa \leq 0$, then $\kappa < \kappa^\star$. If $\kappa > 0$, the inequality $\boldsymbol{Q} \succeq \kappa \boldsymbol{I}_d$ and the Lyapunov integral representation imply

$$\begin{aligned}\boldsymbol{C} &= \int_0^\infty e^{\boldsymbol{J}^\top t} \boldsymbol{Q} e^{\boldsymbol{J} t}\, \mathrm{d}t \\ &\succeq \kappa \int_0^\infty e^{\boldsymbol{J}^\top t} e^{\boldsymbol{J} t}\, \mathrm{d}t \\ &= \kappa \boldsymbol{M}.\end{aligned}$$

Monotonicity of the determinant on the positive-definite cone then gives

$$e^\rho = \det \boldsymbol{C} \geq \kappa^d \det \boldsymbol{M}, \qquad \kappa \leq \kappa^\star. \tag{22}$$

Equality can hold only when $\boldsymbol{C} = \kappa^\star \boldsymbol{M}$: indeed, $\boldsymbol{C} \succeq \kappa^\star \boldsymbol{M}$ together with equality of their determinants forces equality of the matrices. Conversely, $\boldsymbol{C} = \kappa^\star \boldsymbol{M}$ satisfies the determinant constraint and, by Eq. (21), yields $\boldsymbol{Q} = \kappa^\star \boldsymbol{I}_d$. Thus, $\kappa^\star$ is the largest achievable value over the admissible metric class, attained by the unique metric $\boldsymbol{C} = \kappa^\star \boldsymbol{M}$.

We next show that this upper bound is attained by a feasible covariance. Set

$$\boldsymbol{\Sigma}^\star := s\boldsymbol{M}^{-2}, \qquad \boldsymbol{P}^\star := s\boldsymbol{M}^{-1}.$$

Both matrices are positive definite, and $\mathrm{Tr}(\boldsymbol{\Sigma}^\star) = \tau$. Multiplying Eq. (21) on the left and right by $\boldsymbol{M}^{-1}$ gives

$$\boldsymbol{J}\boldsymbol{M}^{-1} + \boldsymbol{M}^{-1}\boldsymbol{J}^\top = -\boldsymbol{M}^{-2}. \tag{23}$$

Hence

$$\boldsymbol{J}\boldsymbol{P}^\star + \boldsymbol{P}^\star \boldsymbol{J}^\top + \boldsymbol{\Sigma}^\star = \boldsymbol{0},$$

so $\boldsymbol{P}^\star$ is the stationary covariance induced by $\boldsymbol{\Sigma}^\star$.

By Theorem 1, the selected metric is

$$\begin{aligned}\boldsymbol{C}^\star(\boldsymbol{\Sigma}^\star) &= e^{\rho/d} (\det \boldsymbol{P}^\star)^{1/d} (\boldsymbol{P}^\star)^{-1} \\ &= \left(\frac{e^\rho}{\det \boldsymbol{M}}\right)^{1/d} \boldsymbol{M} \\ &= \kappa^\star \boldsymbol{M}.\end{aligned}$$

It follows that

$$-\boldsymbol{H}_L^\star(\boldsymbol{\Sigma}^\star) = \kappa^\star \boldsymbol{I}_d.$$

Thus, $\boldsymbol{\Sigma}^\star$ attains the upper bound in Eq. (22) and is therefore a global solution of Eq. (13).

To establish uniqueness, suppose that another feasible covariance $\boldsymbol{\Sigma}$ attains the same objective value. The equality condition above forces its selected metric to satisfy

$$\boldsymbol{C}^\star(\boldsymbol{\Sigma}) = \kappa^\star \boldsymbol{M}.$$

Since Theorem 1 gives $\boldsymbol{C}^\star(\boldsymbol{\Sigma}) \propto \boldsymbol{P}_\infty(\boldsymbol{\Sigma})^{-1}$, there exists $b > 0$ such that

$$\boldsymbol{P}_\infty(\boldsymbol{\Sigma}) = b\boldsymbol{M}^{-1}.$$

The stationary Lyapunov equation and Eq. (23) then imply

$$\begin{aligned}\boldsymbol{\Sigma} &= -\boldsymbol{J}\boldsymbol{P}_\infty(\boldsymbol{\Sigma}) - \boldsymbol{P}_\infty(\boldsymbol{\Sigma})\boldsymbol{J}^\top \\ &= b\boldsymbol{M}^{-2}.\end{aligned}$$

The trace constraint forces $b = s$, proving that

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^\star.$$

In particular, although Eq. (13) permits positive-semidefinite covariances, its unique optimum is positive definite.

Finally, we consider the residual-minimization problem in Eq. (14), where $\boldsymbol{\Sigma}^{1/2}$ denotes the principal positive-definite square root. For $\boldsymbol{\Sigma}^\star = s\boldsymbol{M}^{-2}$,

$$(\boldsymbol{\Sigma}^\star)^{1/2} = \sqrt{s}\, \boldsymbol{M}^{-1}.$$

Choosing

$$\gamma^\star = s^{-1/2}$$

and applying Eq. (23) gives

$$\begin{aligned}
&\boldsymbol{J}(\boldsymbol{\Sigma}^\star)^{1/2} + (\boldsymbol{\Sigma}^\star)^{1/2}\boldsymbol{J}^\top + \gamma^\star\boldsymbol{\Sigma}^\star \\
&= \sqrt{s}\left(\boldsymbol{J}\boldsymbol{M}^{-1} + \boldsymbol{M}^{-1}\boldsymbol{J}^\top + \boldsymbol{M}^{-2}\right) \\
&= \mathbf{0}.
\end{aligned}$$

Hence the residual objective attains its global minimum value of zero.

Since the residual objective is nonnegative and the construction above attains zero, any global minimizer $(\boldsymbol{\Sigma}, \gamma)$ of Eq. (14) must satisfy

$$\boldsymbol{J}\boldsymbol{\Sigma}^{1/2} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{J}^\top + \gamma\boldsymbol{\Sigma} = \mathbf{0}.$$

Because $\boldsymbol{\Sigma} \succ \mathbf{0}$, multiplying on the left and right by $\boldsymbol{\Sigma}^{-1/2}$ yields

$$\boldsymbol{J}^\top\boldsymbol{\Sigma}^{-1/2} + \boldsymbol{\Sigma}^{-1/2}\boldsymbol{J} = -\gamma\boldsymbol{I}_d.$$

By uniqueness of the Lyapunov solution,

$$\boldsymbol{\Sigma}^{-1/2} = \gamma\boldsymbol{M},$$

and hence

$$\boldsymbol{\Sigma} = \gamma^{-2}\boldsymbol{M}^{-2}.$$

The trace constraint determines

$$\gamma^{-2} = \frac{\tau}{\mathrm{Tr}(\boldsymbol{M}^{-2})} = s.$$

Therefore,

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^\star, \qquad \gamma = \gamma^\star.$$

Thus, the residual-minimization problem and the original covariance-design problem have the same unique optimal covariance. □

D More Implementation Details

This section provides additional implementation details for NDLM. We describe the parameterization of the noise generator, the normalization used to control the perturbation budget, and the hyperparameter settings used in our experiments.

D.1 Noise-Network Parameterization

Recall that the noise network receives the current recurrent state $\boldsymbol{z}$, the input $\boldsymbol{x}$, and a Gaussian probe $\boldsymbol{v} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$, and outputs a structured perturbation $\boldsymbol{e} = g_\phi(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{v})$. We parameterize g_ϕ using a lightweight conditional network whose action on $\boldsymbol{v}$ is linear, while its coefficients depend on the current input and latent state. The parameterization is designed such that the induced linear operator is symmetric and positive definite, consistent with the covariance characterization in Theorem 2. Specifically, we use the following attention-like parameterization:

$$g_\phi(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{v}) = \left(\boldsymbol{I}_d + \frac{\boldsymbol{Q}_\phi(\widehat{\boldsymbol{z}})\boldsymbol{Q}_\phi(\widehat{\boldsymbol{z}})^\top}{\sqrt{d_h}}\right)\boldsymbol{v}, \qquad \widehat{\boldsymbol{z}} = \mathrm{RMSNorm}(\boldsymbol{z}) + \boldsymbol{W}_c\,\mathrm{RMSNorm}(\boldsymbol{x}), \quad (24)$$

where $\boldsymbol{Q}_\phi(\widehat{\boldsymbol{z}})$ is a learned multi-head linear projection of the conditioned state and d_h denotes the dimension of the attention head. The identity component ensures a non-degenerate perturbation in every latent direction, while the learned component captures anisotropic directions that depend on the local recurrent dynamics. The perturbations are detached from the task-model update, so that the backbone parameters are optimized by the task objective while the noise-network parameters are optimized by $\mathcal{L}_{\mathrm{cov}}$. During training, structured perturbations are introduced only in the final 2 recurrent steps, where the latent states are expected to lie closer to an attractor.

D.2 PERTURBATION-BUDGET REGULARIZATION

The covariance-design problem constrains the perturbation budget through $\mathrm{Tr}(\mathbf{\Sigma}) = \tau$. In practice, explicitly constructing the full covariance matrix is unnecessary, since the network directly generates samples $\boldsymbol{e} = g_\phi(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{v})$. We therefore impose the noise budget by normalizing the generated perturbations. We consider two normalization strategies.

Per-sample normalization. The per-sample variant independently normalizes each generated perturbation to the prescribed perturbation budget. For a sample i with raw perturbation $\tilde{\boldsymbol{e}}_i$, the normalized perturbation is defined as

$$\boldsymbol{e}_i = \sqrt{\tau}\,\frac{\tilde{\boldsymbol{e}}_i}{\sqrt{\|\tilde{\boldsymbol{e}}_i\|_2^2 + \varepsilon}} \tag{25}$$

where $\varepsilon > 0$ is a small constant for numerical stability.

Per-batch normalization. The per-batch variant instead computes a shared normalization factor from all perturbations in a minibatch. Given a minibatch $\mathcal{B} = \{\tilde{\boldsymbol{e}}_i\}_{i=1}^{|\mathcal{B}|}$, we normalize the perturbations according to

$$\boldsymbol{e}_i = \sqrt{\frac{\tau}{\frac{1}{|\mathcal{B}|}\sum_{j\in\mathcal{B}}\|\tilde{\boldsymbol{e}}_j\|_2^2 + \varepsilon}}\,\tilde{\boldsymbol{e}}_i, \qquad i \in \mathcal{B} \tag{26}$$

Both strategies provide sample-based approximations to the trace constraint in the covariance-design problem without explicitly materializing $\mathbf{\Sigma}$. Unless otherwise specified, we use the same perturbation budget τ when comparing EqR (Huang et al., 2026) and NDLM, ensuring that performance differences arise from covariance structure rather than from a larger overall noise magnitude.

D.3 HYPERPARAMETER SETTINGS

Our backbone configurations follow the corresponding TRM (Jolicoeur-Martineau, 2025) and EqR (Huang et al., 2026) settings for each benchmark. We keep the recurrent architecture, training budget, and batch size fixed when comparing different noise designs, with the additional parameters in our model arising only from the structured noise network. Table 3 summarizes the configurations used for Sudoku-Extreme, Maze-Unique, and Maze-Hard.

Table 3: Hyper-parameter settings for different models on Sudoku-Extreme, Maze-Unique, and Maze-Hard. "–" denotes fields that are not applicable.

Hyper-parameter	Model								
	TRM			EqR			NDLM		
	Sudoku-Extreme	Maze-Unique	Maze-Hard	Sudoku-Extreme	Maze-Unique	Maze-Hard	Sudoku-Extreme	Maze-Unique	Maze-Hard
H-Layer	–	–	–	–	–	–	–	–	–
L-Layer	2	1	2	2	1	2	2	1	2
H-Cycle	3	3	3	3	3	3	3	3	3
L-Cycle	6	4	4	6	4	4	6	4	4
Outer Loop	16	16	16	16	16	16	16	16	16
Hidden Size	512	128	512	512	128	512	512	128	512
#Param (M)	5.03	0.264	6.82	5.03	0.264	6.82	5.55	0.297	7.35
Global Batch Size	768	768	768	768	768	768	768	768	768
Training Steps	50k	100k	50k	50k	100k	50k	50k	100k	50k