TRAJECTORY-ALIGNED GRADIENT SUPPRESSION FOR TEST-TIME GENERALIZATION IN RECURRENT TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time compute scaling has emerged as a powerful paradigm for improving reasoning in large language models. Recurrent transformers offer a natural mechanism for such scaling by repeatedly applying a shared computation block, producing a sequence of latent states that can be unrolled for additional steps at inference. However, additional recurrence often struggles to improve performance beyond the training depth. We study this limitation through a stability–progress trade-off in the latent state dynamics: the model must learn a recurrent operator that remains stable when extrapolated to greater depths, while still making step-by-step progress along the latent trajectory required for multi-step reasoning. We examine this trade-off through the geometry of training gradients. A first-order expansion of the loss under additional recurrence identifies the component of the backpropagated state gradient aligned with the local tangent of the latent-state trajectory as a local channel of depth sensitivity. This trajectory-aligned gradient has a dual role: it can support sequential progress along the recurrent path, but it also determines how the loss changes when computation is extended beyond the training depth. We formulate a curvature-weighted local signal filtering objective for this component and derive a corresponding closed-form suppression rule. Motivated by this analysis, we propose **Trajectory-Aligned Gradient Suppression (TAGS)**, a backward-pass intervention that selectively attenuates tangent-aligned state gradients while preserving transverse gradients for iterative refinement. TAGS adds no parameters, modifies only training-time gradients, and leaves inference unchanged. Across structural reasoning benchmarks (maze navigation, Sudoku, graph shortest path) and symbolic reasoning tasks, TAGS improves test-time compute generalization over backpropagation-through-time (BPTT), truncated BPTT, and fixed-point baselines. The code will be available at https://anonymous.4open.science/r/TAGS.

1 INTRODUCTION

Scaling test-time computation has emerged as a powerful paradigm for improving reasoning in large language models, from chain-of-thought prompting (Wei et al., 2022) to inference-time search and verification (Guo et al., 2025). Recurrent transformers provide a natural mechanism for such scaling (Geiping et al., 2025; Koishekenov et al., 2025). By iteratively applying a shared computation block to refine latent states, they allow the same learned operator to be unrolled to greater depth at test time, in principle enabling harder problems to be solved through additional latent computation. This capacity for **test-time compute generalization**, i.e., generalizing a learned iterative procedure beyond its training depth, is the central promise of weight-shared recurrence, and underlies architectures like Universal Transformers (Dehghani et al., 2019), Looped Transformers (Giannou et al., 2023) and Huginn (Geiping et al., 2025).

Realizing this promise requires the shared recurrent operator f_θ, which iteratively updates the latent state across recurrence steps, to balance a stability–progress tradeoff. On one hand, the operator must exhibit *depth-invariant* behavior: stable refinement dynamics that remain both well-behaved and productive as recursion depth increases (Bai et al., 2019; Dehghani et al., 2019). On the other hand, it must support *trajectory-dependent* computation: the ability to carry out sequential multi-step reasoning where intermediate states encode meaningful progress (Giannou et al., 2023; Marro et al., 2025). In practice, existing training regimes struggle to maintain this trade-off, often forcing a compromise. Prior architectural interventions, including adaptive halting (Graves, 2017; Dehghani

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.

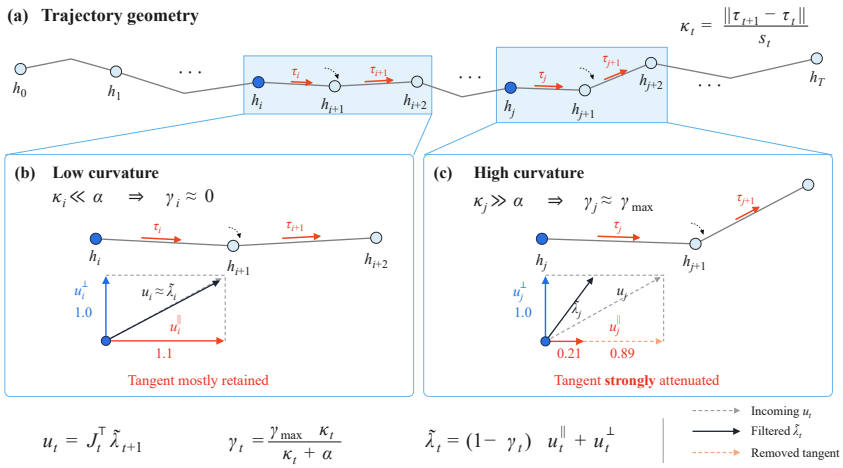


Figure 1: **Trajectory-Aligned Gradient Suppression (TAGS).** A recurrent transformer traces a trajectory through latent state space. A first-order loss expansion identifies the tangent-aligned state gradient as a local channel of sensitivity to additional recurrence. TAGS applies curvature-dependent suppression to this component while preserving the transverse component of each local backward signal. The intervention operates during training; inference uses frozen parameters.

et al., 2019) and depth routing (Bae et al., 2025; Zhang et al., 2026), improve computational flexibility by deciding how much recurrence to allocate, but they do not directly address whether the gradients used to train the recurrent operator make additional steps useful beyond the training depth. This suggests that a key limitation lies in the training dynamics rather than only in the architecture itself.

In Section 3, we analyze the gradients induced by finite-depth recurrent training and examine this stability–progress trade-off through the geometry of the latent-state trajectory. A first-order expansion of the loss under an additional recurrence identifies the component of the backpropagated state gradient aligned with the local tangent of the latent-state trajectory as the local carrier of first-order depth sensitivity. Yet this same tangent-aligned component can also reflect useful progress along the recurrent path, so the two signals are entangled and cannot be separated by uniformly weakening or truncating recurrent gradients. We further show that this credit propagates backward through local smoothness (Theorem 1), and that the change in this depth sensitivity across neighboring horizons decomposes into speed-change, turning, and loss-nonlinearity terms, with curvature entering through the turning term (Proposition 1). This analysis identifies candidate signals for localizing an intervention but does not by itself determine its strength. To design a selective intervention, we formulate a curvature-weighted penalty on the squared magnitude of this component while limiting distortion of the incoming backward signal. The resulting objective admits a closed-form minimizer whose under-relaxed form corrects only along the tangent, exactly preserving the transverse component of each local backward signal.

Building on these insights, we propose **Trajectory-Aligned Gradient Suppression** (TAGS), a geometry-aware training-time backward-pass intervention for recurrent transformers. As shown in Figure 1, TAGS targets the training-depth-sensitive component of gradient propagation. At each recurrence step during training, TAGS decomposes the backpropagated state gradient into tangent-aligned and transverse components. It then suppresses the tangent-aligned component with a curvature-dependent strength, while preserving the transverse component. Thus, TAGS interpolates between standard backpropagation through time (BPTT), which propagates gradients through all recurrent computation steps, and hard trajectory-orthogonal projection, which removes the tangent-aligned gradient component entirely. This allows TAGS to retain gradient signal needed for step-by-step reasoning while reducing the training depth sensitive component.

Our key contributions are:

1. We formalize test-time compute generalization in recurrent transformers as a stability–progress trade-off in latent state dynamics, and establish a first-order connection between tangent-aligned state gradients and loss extrapolation beyond the training depth.

2. We introduce **TAGS**, a training-time backward-pass intervention whose suppression strength is derived from the closed-form solution of a curvature-weighted signal-filtering objective, attenuating

only the tangent-aligned component while exactly preserving the transverse component of each local backward signal.

3. Across maze navigation, Sudoku, graph shortest path, and mathematical reasoning benchmarks, TAGS improves test-time compute generalization over BPTT, truncated BPTT, and fixed-point baselines in both training-from-scratch and fine-tuning settings.

2 Related Work

Test-time compute scaling and recurrent reasoning. Scaling computation at inference time has emerged as a central axis for improving reasoning, spanning explicit chain-of-thought prompting (Wei et al., 2022), reinforcement-learned reasoning traces (OpenAI et al., 2024; Guo et al., 2025), inference-time search and verification (Yao et al., 2023; Dhuliawala et al., 2024), and latent-space recurrence (Geiping et al., 2025; Koishekenov et al., 2025). Weight-shared recurrent transformers, including Universal Transformers (Dehghani et al., 2019), Looped Transformers (Giannou et al., 2023), and recent depth-recurrent language models (Geiping et al., 2025; Koishekenov et al., 2025), instantiate test-time compute scaling inside the model by repeatedly applying a shared operator to refine latent states. However, prior work on algorithmic and recurrent reasoning shows that deeper rollouts can suffer from saturation, degradation, or "overthinking" when models are evaluated beyond the depths seen during training (Schwarzschild et al., 2021; Bansal et al., 2022; He, 2025). Our work studies this failure mode directly: how should recurrent models be trained so that additional test-time steps remain productive rather than merely increasing the length of the rollout?

Training recurrent models for depth extrapolation. We view existing training methods as different compromises in the stability–progress trade-off of recurrent latent dynamics. Standard backpropagation through time (BPTT) (Werbos, 1990) propagates gradients through the full finite recurrent trajectory. This preserves gradient signal for step-by-step computation, but can make the learned operator overly specialized to the recurrence depth used during training. Truncated BPTT (Williams & Zipser, 1995) weakens this dependence by limiting how far gradients propagate backward through the recurrent trajectory, but it removes gradient signal according to temporal distance rather than according to the geometric source of training-depth sensitivity. Fixed-point and deep-equilibrium methods define gradients implicitly at a solved equilibrium rather than along a finite recurrent trajectory (Bai et al., 2019). This changes what the gradient describes: an equilibrium property rather than the path taken to reach it. TAGS takes a different approach: it retains finite-trajectory gradient propagation, but selectively attenuates the tangent-aligned, state-gradient component that our analysis identifies as a leading source of training-depth sensitivity.

Adaptive computation and dynamic depth routing. A complementary line of work learns how much computation to allocate per input. Adaptive Computation Time (Graves, 2017) and PonderNet (Banino et al., 2021) learn halting policies, while Mixture-of-Recursions (Bae et al., 2025) and Mixture-of-Depth-Recurrent Transformers (Zhang et al., 2026) route tokens or examples through variable recurrence depths. These methods primarily address the allocation problem: *when* should the model stop, and *where* should additional recurrence be spent? TAGS addresses a complementary training problem: whether the recurrent operator has been trained so that spending additional computation remains beneficial beyond the training depth.

3 Background and Geometric Analysis

Finite-depth training must support step-by-step progress, yet additional recurrence can degrade performance beyond the training depth (Figure 2(a)). The analysis in this section starts from a simple observation. Under BPTT, the gradient that reaches a latent state has a component along the direction in which the trajectory is moving. That component is the sensitivity of the loss to the state having advanced slightly farther along its path, so it measures whether one more unit of latent motion would help or hurt at the training depth. Training on it teaches the operator to calibrate how much progress it makes per step to the depth T it is trained at. This is useful for moving along the path, but it is also the channel through which the operator specializes to T. We make this precise in three steps: the tangent-aligned state gradient is the first-order loss change under one more recurrence (Eq. 3); it is carried backward from the end of the rollout (Theorem 1); and how it transfers from one horizon to the next depends on whether the trajectory turns (Proposition 1). The section closes by stating what this analysis does and does not determine about the intervention in Section 4.

Setup. Consider a shared-weight recurrence $h_{t+1} = f_\theta(h_t, c)$ for $t = 0, \ldots, T-1$, with latent state h_t and fixed context c, as in Universal Transformers (Dehghani et al., 2019) and Huginn (Geiping

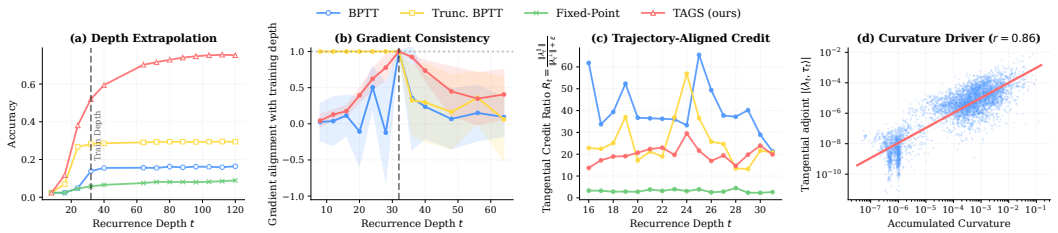


Figure 2: **Diagnosing the extrapolation trade-off. (a)** Additional recurrence beyond the training depth yields diminishing gains or degradation. **(b)** Gradient similarity decreases as the evaluation recurrence depth moves farther from the training depth. **(c)** Ratio of tangent-aligned to transverse BPTT state-gradient magnitudes. **(d)** Association between accumulated trajectory curvature and tangent-aligned state-gradient magnitude.

et al., 2025). Training minimizes a terminal loss $\mathcal{L}^{(T)} = \ell(q_\theta(h_T), y)$ with readout q_θ; we call the depth at which the loss is evaluated the *horizon*. Under BPTT,

$$\lambda_T^{(T)} = \nabla_{h_T}\mathcal{L}^{(T)}, \qquad \lambda_t^{(T)} = J_t^\top \lambda_{t+1}^{(T)}, \qquad g_T^{\mathrm{rec}}(\theta) = \sum_{t=0}^{T-1} B_t^\top \lambda_{t+1}^{(T)}, \tag{1}$$

where J_t and B_t are the state and parameter Jacobians of f_θ. The trajectory traces a discrete curve in $\mathbb{R}^d$, described by discrete Frenet–Serret quantities (Do Carmo, 1976):

$$v_t = h_{t+1} - h_t, \qquad s_t = \|v_t\|, \qquad \tau_t = \frac{v_t}{s_t}, \qquad \kappa_t = \frac{\|\tau_{t+1} - \tau_t\|}{s_t}, \tag{2}$$

namely the displacement, step size, unit tangent, and curvature (the change in direction per unit displacement), defined where $s_t > 0$. State gradients decompose into tangent-aligned and transverse parts, $\lambda_t^{\parallel} = \langle \lambda_t, \tau_t \rangle \tau_t$ and $\lambda_t^{\perp} = \lambda_t - \lambda_t^{\parallel}$.

Tangent-aligned gradients and depth sensitivity. Write $F(h) = f_\theta(h, c)$ and let $C_r(h) = \ell(q_\theta(F^r(h)), y)$ be the loss after r remaining steps. Since the operator is shared, $C_{T-t}(h_{t+i}) = \mathcal{L}^{(T+i)}$: starting one state farther along the trajectory and applying the same $T - t$ remaining steps lands at h_{T+i}. Taylor expansion along v_t therefore gives, for $0 \le t < T$ with $s_t > 0$,

$$D_T := \mathcal{L}^{(T+1)} - \mathcal{L}^{(T)} = s_t \langle \lambda_t^{(T)}, \tau_t \rangle + R_{t,T}, \qquad |R_{t,T}| \le \tfrac{H_{t,T}}{2} s_t^2, \tag{3}$$

where $H_{t,T}$ bounds the Hessian of C_{T-t} near $[h_t, h_{t+1}]$. Up to the remainder, the tangent-aligned BPTT gradient at any interior step is the loss change from one more recurrence. This is the dual role described in the introduction: a negative value says that further motion along the path would help, a positive value that it would hurt, and in either case the same scalar is what the operator is trained to adjust. Both roles occupy a single direction, so weakening or truncating the whole gradient cannot separate them; an intervention aimed at depth sensitivity has to act on this component specifically. Appendix A.3 shows how the component is carried backward (Theorem 1): tangent-aligned credit at step t is inherited from the end of the rollout and injected from transverse components at a rate bounded by the smoothness of f_θ, so it originates at the horizon or in transverse-to-tangent mixing along the way rather than in any single step, which is what makes it a natural target for a backward-pass intervention.

Horizon transfer. Holding the gradient $\lambda_t^{(T)}$ fixed, if the trajectory keeps its direction, the linearized increment at the next step is the current one rescaled by $s_{t+1}/s_t > 0$ and keeps its sign; if the trajectory turns, other components of the gradient contribute and the sign can change. Formally, $D_{T+1} - D_T$ is a second difference of C_{T-t} along h_t, h_{t+1}, h_{t+2}, driven by the change in displacement, which splits exactly as

$$v_{t+1} - v_t = (s_{t+1} - s_t)\,\tau_{t+1} + s_t(\tau_{t+1} - \tau_t), \qquad \|\tau_{t+1} - \tau_t\| = \kappa_t s_t. \tag{4}$$

Proposition 1 (Horizon transfer of depth sensitivity) *For $0 \le t \le T-2$ with $s_t, s_{t+1} > 0$, if $\|D^2 C_{T-t}\| \le H$ near $[h_t, h_{t+1}]$ and $[h_t, h_{t+2}]$, then*

$$D_{T+1} - D_T = \underbrace{(s_{t+1} - s_t)\langle \lambda_t^{(T)}, \tau_{t+1} \rangle}_{\textit{speed change}} + \underbrace{s_t \langle \lambda_t^{(T)}, \tau_{t+1} - \tau_t \rangle}_{\textit{turning},\ |\cdot| \le \kappa_t s_t^2 \|\lambda_t^{(T)}\|} + R'_t, \tag{5}$$

with $|R'_t| \le \frac{H}{2}\left[(s_t + s_{t+1})^2 + 2s_t^2\right]$.

The proof is a first-order Taylor expansion of C_{T-t} around h_t with a Hessian-controlled remainder (Appendix A.4). Two features of the decomposition matter for what follows. First, speed change rescales the linearized increment by a positive factor and preserves its sign, whereas turning lets other components of the gradient contribute and can reverse it; curvature enters only through the turning term, jointly with step size and the state gradient. Second, for small turning the turning term is first order in the transverse component of $\lambda_t^{(T)}$ and only second order in its tangent-aligned component, and the remainder, which captures loss nonlinearity, is not determined by the trajectory geometry and can be comparable to the geometric terms.

What the analysis determines, and what it does not. The results above fix the *direction* of an intervention but not its *strength*. Equation 3 and Theorem 1 identify the tangent-aligned state gradient as the component that carries first-order depth sensitivity and show that it is propagated along the whole rollout; this is a principled reason to act on that component and to leave the transverse component, which carries the remaining refinement signal, untouched. Proposition 1 then makes directional change $\omega_t = \|\tau_{t+1} - \tau_t\| = \kappa_t s_t$ a natural signal for *where* the first-order estimate in Eq. 3 is least reliable as a guide to the next horizon, since turning is what can reverse the sign of the linearized increment. Curvature and the tangent direction therefore play different roles in TAGS: the tangent direction is *what* carries depth-T credit, and curvature marks *where* that credit is least likely to transfer, so credit is down-weighted there. This is a trust-weighting rationale, not a claim that suppression shrinks the turning term: by the second feature above, the leading turning contribution is transverse and is preserved, and attenuating the tangent-aligned component need not reduce the term's magnitude (Appendix A.4). The proposition also does not rank candidate signals, and a curvature-only bound cannot replace the smoothness control in Theorem 1 (Appendix A.3). We therefore treat the gate as an inductive bias and select it empirically. Within the same tangent-suppression rule we compare curvature, step size, and downstream gradient norm ($\|\lambda_{t+1}\|$) as gating signals, with each signal's scale chosen on the same in-distribution validation budget: on maze navigation, curvature reaches 79.44% average OOD accuracy against 57.78% for step size and 46.81% for gradient norm (Table 4), and the association in Figure 2(d) ($r = 0.86$) is consistent with this choice; speed-change and unnormalized-turning (ω_t) gates are not evaluated. Section 4 builds on exactly this division: the direction comes from the analysis, the curvature weighting is the empirically selected bias, and the local objective there is a way to read the resulting rule, not a further derivation.

4 Method: Trajectory-Aligned Gradient Suppression

Section 3 identifies the tangent-aligned state gradient as the component to act on and curvature as the empirically selected signal for how strongly to act. *Trajectory-Aligned Gradient Suppression* (TAGS) combines the two in the backward pass: at each interior recurrence step it attenuates the tangent-aligned part of the local backward signal by a curvature-dependent factor and passes the transverse part through unchanged.

Backward update. Let $u_t = J_t^\top \tilde{\lambda}_{t+1}$ be the local backward proposal at step t, with $\tilde{\lambda}_T = \nabla_{h_T} \mathcal{L}^{(T)}$; unlike $\lambda_t^{(T)}$, it incorporates any downstream TAGS modifications and is generally not the true BPTT state gradient. For $0 \le t \le T-2$, a unit tangent τ_t, and constants $\gamma_{\max} \in [0, 1]$ and $\alpha > 0$, TAGS replaces u_t by

$$\tilde{\lambda}_t = u_t - \gamma_t \langle u_t, \tau_t \rangle \tau_t, \qquad \gamma_t = \gamma_{\max} \frac{\kappa_t}{\kappa_t + \alpha}. \tag{6}$$

Writing $u_t^{\parallel} = \langle u_t, \tau_t \rangle \tau_t$ and $u_t^{\perp} = u_t - u_t^{\parallel}$, this is

$$\tilde{\lambda}_t^{\parallel} = \left(1 - \gamma_{\max} \frac{\kappa_t}{\kappa_t + \alpha}\right) u_t^{\parallel}, \qquad \tilde{\lambda}_t^{\perp} = u_t^{\perp}. \tag{7}$$

The gate γ_t is 0 at zero curvature and rises toward $\gamma_{\max}$ as $\kappa_t/\alpha \to \infty$, so $\gamma_{\max}$ bounds the suppression strength and α sets the curvature scale at which half of that strength is reached. The recurrent parameter pseudogradient is

$$g_T^{\mathrm{TAGS}}(\theta) = \sum_{t=0}^{T-1} B_t^\top \tilde{\lambda}_{t+1}. \tag{8}$$

Both the state vector-Jacobian product (VJP) $J_t^\top \tilde{\lambda}_{t+1}$ and the parameter VJP $B_t^\top \tilde{\lambda}_{t+1}$ use the same incoming signal; TAGS filters only the state-VJP output before passing it to earlier steps, and filtering the incoming signal before both VJPs would define a different method. Contributions through a

trainable readout, initial state, or context are added separately, using the modified state signals where applicable.

Reading the rule as a local objective. The gate in Eq. 6 has a compact interpretation that fixes how α and $\gamma_{\max}$ relate to the suppression strength. For $\alpha > 0$, consider

$$\mathcal{E}_t(z) = \frac{\alpha}{2}\|z - u_t\|^2 + \frac{\kappa_t}{2}\langle z, \tau_t\rangle^2, \tag{9}$$

which trades fidelity to the local proposal against a penalty on the squared tangent-aligned component, weighted by curvature; the linear curvature weight corresponds to penalizing tangent-aligned credit per unit of trajectory turning (Appendix B.1). Its unique minimizer is $z_t^\star = u_t - \frac{\kappa_t}{\alpha+\kappa_t}\langle u_t, \tau_t\rangle\tau_t$, and the under-relaxed step $\tilde{\lambda}_t = (1-\gamma_{\max})u_t + \gamma_{\max} z_t^\star$ recovers Eq. 6 exactly. Writing $\Gamma = \gamma_{\max}$, quadraticity gives

$$\mathcal{E}_t(u_t) - \mathcal{E}_t(\tilde{\lambda}_t) = \Gamma(2-\Gamma)\big[\mathcal{E}_t(u_t) - \mathcal{E}_t(z_t^\star)\big], \tag{10}$$

so $\Gamma(2-\Gamma)$ is the fraction of the available local-objective reduction that the rule achieves. This reading is local to the proposal u_t at each step: transverse preservation holds relative to u_t rather than the unmodified BPTT gradient, and the objective does not make the full parameter update the gradient of a global scalar objective or guarantee descent in the task loss (Appendix B.1).

Numerical geometry. The derivation uses unit tangents. Algorithm 1 instead uses the stabilized direction $\bar{\tau}_t = v_t/(s_t+\epsilon)$ and the corresponding numerical curvature $\hat{\kappa}_t$, both held fixed during the modified backward pass; we use $\epsilon = 10^{-6}$ for the structural reasoning experiments. For $s_t > 0$, write $\rho_t = s_t/(s_t+\epsilon)$. The implemented filter has effective suppression $\gamma_t^{\text{eff}} = \hat{\gamma}_t\rho_t^2$ along the unit tangent and still preserves its true transverse component. For an active gate, it is the $\Gamma\rho_t^2$-relaxed minimizer of the unit-tangent objective with weight $\hat{\kappa}_t$, and the reduction factor in Eq. 10 becomes $\Gamma\rho_t^2(2-\Gamma\rho_t^2)$. The numerical direction is not unit length, so $(\bar{\tau}_t^\top u_t)\bar{\tau}_t$ is not an orthogonal projection. At zero displacement the correction is zero; a near-stationary safeguard also disables the gate when $s_t < \delta_{\text{num}}$.

Algorithm 1 One training iteration with Trajectory-Aligned Gradient Suppression (TAGS)

Input: θ, h_0, c, y^*, $T \geq 1$
Settings: $\gamma_{\max}$, α, ϵ, δ_{num}
Forward pass (same as BPTT)
for $t = 0, \ldots, T-1$ **do**
    $h_{t+1} \leftarrow f_\theta(h_t, c)$
$\mathcal{L} \leftarrow \ell(q_\theta(h_T), y^*)$
Cache geometry; hold fixed in backward pass
for $t = 0, \ldots, T-1$ **do**
    $v_t \leftarrow h_{t+1} - h_t$, $s_t \leftarrow \|v_t\|$
    $\bar{\tau}_t \leftarrow v_t/(s_t+\epsilon)$
for $t = 0, \ldots, T-2$ **do**
    $\hat{\kappa}_t \leftarrow \|\bar{\tau}_{t+1} - \bar{\tau}_t\|/(s_t+\epsilon)$
    $\hat{\gamma}_t \leftarrow \gamma_{\max}\hat{\kappa}_t/(\hat{\kappa}_t+\alpha)$
    If $s_t < \delta_{\text{num}}$, set $\hat{\gamma}_t \leftarrow 0$
Backward pass
$\tilde{\lambda}_T \leftarrow \nabla_{h_T}\mathcal{L}$
$u_{T-1} \leftarrow J_{T-1}^\top \tilde{\lambda}_T$
$g_{T-1} \leftarrow B_{T-1}^\top \tilde{\lambda}_T$
$\tilde{\lambda}_{T-1} \leftarrow u_{T-1}$ ▷ Ungated boundary
for $t = T-2, \ldots, 0$ **do**
    $u_t \leftarrow J_t^\top \tilde{\lambda}_{t+1}$
    $g_t \leftarrow B_t^\top \tilde{\lambda}_{t+1}$
    $\tilde{\lambda}_t \leftarrow u_t - \hat{\gamma}_t\langle u_t, \bar{\tau}_t\rangle\bar{\tau}_t$
Accumulate remaining trainable-module contributions into g_{other}
$g \leftarrow \sum_{t=0}^{T-1} g_t + g_{\text{other}}$
Update: $\theta \leftarrow \text{ADAM}(\theta, g)$

Boundary convention. All tangents are computed before curvature. With only $h_0, \ldots, h_T$ available, curvature is defined for $t = 0, \ldots, T-2$. The final state VJP is therefore left unfiltered, $\tilde{\lambda}_{T-1} = u_{T-1}$, and suppression is applied only at earlier steps. This requires no additional rollout; for $T = 1$, there are no gated steps.

Two limits. For the ideal unit-tangent rule, $\gamma_{\max} = 0$ recovers BPTT, and $\gamma_{\max} = 1$ approaches hard trajectory-orthogonal projection as $\kappa_t/\alpha \to \infty$; at finite α and finite curvature, suppression remains partial, and at $\kappa_t = 0$ the proposal is unchanged. TAGS therefore interpolates between full BPTT and hard projection, with the interpolation set by local curvature. The numerical implementation adds the damping described above.

Inference is unchanged. TAGS modifies only the training-time backward pass. Model parameters are optimized during training and frozen during inference, when the same forward recurrence is unrolled to the chosen evaluation depth. TAGS adds no parameters or inference-time filtering.

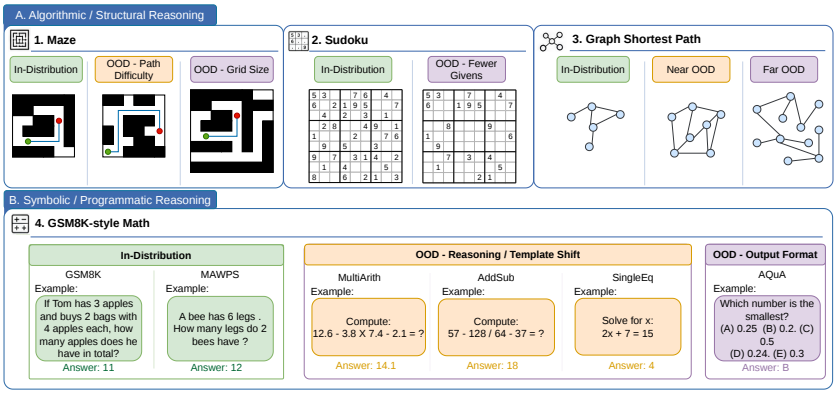


Figure 3: **Evaluation suite for test-time compute generalization.** We evaluate recurrent transformers on structural reasoning tasks (maze navigation, Sudoku, graph shortest path) and symbolic mathematical reasoning tasks under ID and OOD evaluation settings. OOD settings test harder paths or larger grids in mazes, fewer givens in Sudoku, more nodes in graphs, and cross-dataset shifts in mathematical reasoning, including changes in reasoning templates and output format. Examples are schematic illustrations only and are not actual dataset samples.

Table 1: **Maze navigation results.** Models trained on 20x20 grid with path difficulty between 20-50 and recurrence depth $T = 32$, evaluated at depth $T' = 64$. Near and Far OOD evaluate path difficulty and grid size at progressively larger distribution shifts (see column headers for ranges). OOD Avg. is the mean across all four OOD columns. Values are mean $\pm$ standard deviation over three seeds. Best results are highlighted in **bold** and second-best results are underlined.

Method	In Distribution	Near OOD		Far OOD		Statistics
		Path Difficulty (50-80)	Grid Size (25x25)	Path Difficulty (80-100)	Grid Size (30x30)	OOD Avg.
BPTT	97.5 $\pm$2.2	33.7 $\pm$5.1	16.3 $\pm$7.8	8.0 $\pm$3.1	4.6 $\pm$2.5	15.7 $\pm$2.5
Fixed Point	94.4 $\pm$3.7	20.3 $\pm$9.4	10.8 $\pm$4.1	2.34 $\pm$0.0	2.37 $\pm$0.0	9.0 $\pm$3.0
Trunc. BPTT	98.1 $\pm$1.9	**97.5** $\pm$2.5	22.0 $\pm$7.7	**93.3** $\pm$3.6	27.1 $\pm$4.4	60.0 $\pm$2.5
TAGS (Ours)	**100.0** $\pm$0.0	95.2 $\pm$2.5	**64.9** $\pm$7.3	92.3 $\pm$4.1	**48.6** $\pm$9.9	**75.3** $\pm$5.3

5 Experiments and Results

We evaluate TAGS on the test-time compute generalization problem motivated in Sections 1–4. Our experiments address three questions:

(A) Does TAGS improve OOD generalization while maintaining comparable in-distribution performance?

(B) Does TAGS improve accuracy with additional test-time recurrence beyond the training depth?

(C) Do gradient diagnostics and suppression ablations support the tangent direction and curvature-based gating used by TAGS?

Experimental setup. We evaluate TAGS across two suites: *algorithmic/structural reasoning* and *symbolic reasoning* (Figure 3). The structural suite includes maze navigation, Sudoku, and graph shortest path, with controlled OOD shifts in path length and grid size for mazes, clue density for Sudoku, and graph size for shortest path. These models are trained from scratch on task-specific training splits. The symbolic suite evaluates mathematical reasoning by fine-tuning a pretrained Huginn model (Geiping et al., 2025) following the evaluation setup of MoDR (Zhang et al., 2026). GSM8K and MAWPS provide ID evaluation, while SingleEq, MultiArith, AddSub, and AQuA test cross-dataset transfer, including reasoning-template and output-format shifts. Unless otherwise stated, structural models are trained with $T_{\text{train}} = 32$ and evaluated with $T_{\text{test}} = 64$, while mathematical reasoning models are trained with $T_{\text{train}} = 16$ and evaluated with $T_{\text{test}} = 32$. All parameter updates occur during training or fine-tuning; parameters remain frozen during evaluation and recurrence-depth sweeps.

Baselines and metrics. We compare TAGS against three gradient-propagation baselines: **BPTT**, which propagates gradients through the full recurrent trajectory; **truncated BPTT**, which limits

Table 2: **Graph shortest-path and Sudoku results. Left**: graph shortest-path models are trained from scratch with recurrence depth $T = 32$ and evaluated at increased depth $T' = 64$. **Right**: Sudoku models are trained on easy puzzles and evaluated on harder puzzles, using the same recurrence depth setup. For both tasks, OOD Avg. is the mean across OOD columns when multiple OOD splits are present. Values are mean ± standard deviation over three seeds. Best results are shown in **bold** and second-best results are underlined.

Method	In Distribution (n=10)	Near OOD (n=12)	Far OOD (n=14)	Statistics OOD Avg.
BPTT	96.4 ±3.1	15.1 ±5.5	3.3 ±1.2	9.2 ±2.8
Fixed Point	80.1 ±4.4	7.3 ±2.5	2.37 ±0.1	4.8 ±1.3
Trunc. BPTT	96.2 ±1.5	20.3 ±3.7	5.5 ±3.2	12.9 ±2.4
TAGS (Ours)	**97.6** ±1.3	**30.4** ±4.1	**14.1** ±2.8	**22.3** ±2.5

Method	ID (Easy)	OOD (Hard)
BPTT	94.1 ±4.3	35.3 ±5.6
Fixed Point	96.2 ±1.4	15.6 ±1.9
Truncated BPTT	96.2 ±1.5	39.9 ±3.7
TAGS (Ours)	**97.8** ±1.1	**45.2** ±3.0

Table 3: **Mathematical reasoning benchmark results.** All fine-tuned methods are trained with recurrence depth $T_{\text{train}} = 16$ and evaluated with increased test-time recurrence depth $T_{\text{test}} = 32$. Fine-tuned results are mean ± standard deviation over three seeds; the Huginn (No FT) row is a single evaluation of the released checkpoint. OOD Avg. is the mean across all four OOD columns. Best results are highlighted in **bold** and second-best results are underlined.

Method	In Distribution		Out Of Distribution				Statistics	
			Reasoning / Template Shift			Multiple Choice		
	GSM8K	MAWPS	SingleEq	MultiArith	AddSub	AQuA	ID Avg.	OOD Avg.
Huginn (No FT)	43.59	71.85	76.97	79.83	71.90	**27.95**	57.72	64.16
BPTT	49.2 ±1.3	76.8 ±2.1	80.5 ±1.2	88.3 ±1.5	68.1 ±2.7	23.7 ±1.6	63.0 ±1.2	65.2 ±0.9
Fixed Point	35.2 ±2.4	76.3 ±1.6	77.0 ±2.2	75.8 ±3.1	71.9 ±1.8	10.0 ±2.6	55.8 ±1.4	58.7 ±1.2
Trunc. BPTT	44.0 ±1.7	76.6 ±1.8	79.3 ±1.5	87.0 ±1.9	68.2 ±1.6	25.2 ±0.2	60.3 ±1.2	64.9 ±0.7
TAGS (Ours)	**49.9** ±0.3	**78.0** ±1.1	**81.2** ±1.7	**89.6** ±2.3	**72.5** ±1.4	25.0 ±1.5	**64.0** ±0.6	**67.1** ±0.9

gradient propagation to a fixed temporal window; and **fixed-point training**. The main structural comparisons use a truncation window of eight steps. Additional maze comparisons include adaptive computation, deep supervision, and alternative truncation and suppression rules (Table 4). For mathematical reasoning, we also report the base Huginn model without fine-tuning. We report accuracy (%) on ID and OOD splits, with **OOD Avg.** denoting the unweighted mean across a task family's OOD columns. Main trained-model results are means and standard deviations over three seeds; Table 4 reports the separate ablation point estimates.

Implementation details and compute. All experiments are run on NVIDIA A40 GPUs. For structural reasoning tasks, we train 2-layer recurrent transformers from scratch with hidden size 512, 8 attention heads, RoPE-2D positional embeddings, and batch size 64. Models are trained for 5,000 epochs on mazes and 50,000 epochs on Sudoku and graph shortest path, with evaluation every 500 epochs. For symbolic mathematical reasoning, we LoRA-finetune the pretrained Huginn model on the same GPU type, using rank 8, scaling factor 8, batch size 1, and gradient accumulation over 4 steps. Additional optimizer settings, TAGS hyperparameters, LoRA target modules, random seeds, and runtime details are provided in the Appendix.

TAGS improves structural OOD accuracy while maintaining comparable ID performance. At the reported evaluation depths, TAGS achieves higher average OOD accuracy than BPTT, truncated BPTT, and fixed-point training on maze navigation, Sudoku, and graph shortest path (Tables 1 and 2). On mazes, its advantage over truncated BPTT is concentrated in grid-size extrapolation; truncated BPTT has higher mean accuracy on the two path-difficulty splits. The symbolic results also show an average improvement of 1.9 percentage points on top of BPTT (Table 3).

Scaling with additional recurrence. Tables 1–3 compare methods at fixed extrapolation depths. We separately sweep T_{test} for TAGS, BPTT, truncated BPTT, and fixed-point training while keeping each trained checkpoint fixed (Figure 4). BPTT often saturates or degrades beyond the training depth, while TAGS consistently improves OOD accuracy as recurrence increases.

Gradient diagnostics support the geometric motivation. We examine the gradient quantities discussed in Section 3. Gradient alignment between training-depth and evaluation-depth rollouts decreases as the depth mismatch grows (Figure 2b), indicating that backward signals depend on rollout depth. The tangent-aligned gradient ratio

$$R_t = \frac{\|\lambda_t^{\parallel}\|}{\|\lambda_t^{\perp}\| + \epsilon}$$

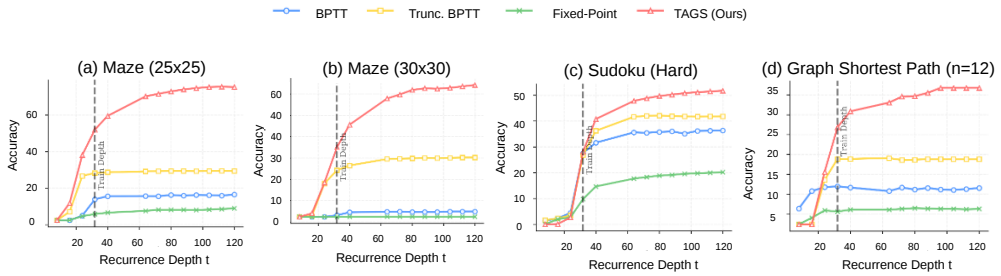


Figure 4: **Scaling with test-time recurrence depth.** Structural models are trained at $T_{\text{train}} = 32$ and evaluated at increasing recurrence depths with frozen parameters. Dashed vertical lines indicate the training depth.

Table 4: **Maze-navigation ablations.** Reported accuracy (%) for alternative training and suppression rules (top) and gating signals within the same tangent-suppression family (bottom). These are point estimates from the ablation runs, separate from the three-seed summaries in the main results tables. In the lower block, gate-specific scales are selected using the same in-distribution validation budget. Bold and underlined entries indicate the best and second-best values in each column.

Method	In Distribution	Near OOD		Far OOD		OOD Avg.
		Path Difficulty (50–80)	Grid Size (25×25)	Path Difficulty (80–100)	Grid Size (30×30)	
Training and suppression variants						
Trunc. BPTT ($n = 8$)	**100.00**	**99.51**	28.02	**94.82**	29.49	62.96
Trunc. BPTT ($n = 16$)	**100.00**	88.57	26.75	55.17	18.45	47.24
Trunc. BPTT (random n)	97.75	61.32	33.10	16.11	12.98	30.88
Adaptive computation head	99.20	96.50	39.00	78.50	24.50	59.63
Deep supervision	**100.00**	87.50	21.48	60.25	6.34	43.89
Random-direction suppression	55.85	3.61	2.63	2.34	2.34	2.73
Hard projection	**100.00**	92.67	**70.21**	48.63	**60.74**	<u>68.06</u>
Gate-signal comparison: same tangent-suppression family						
Step-size gate, $z_t = s_t$	99.70	93.26	42.67	73.73	21.48	57.78
Downstream-gradient-norm gate	<u>99.90</u>	63.86	<u>65.10</u>	8.69	49.60	46.81
TAGS: curvature gate, $z_t = \kappa_t$	**100.00**	<u>99.31</u>	**70.21**	<u>90.33</u>	<u>57.91</u>	**79.44**

measures the relative magnitude of tangent-aligned and transverse state gradients (Figure 2c). Eq. 3 connects the tangent-aligned BPTT state gradient to the first-order loss change under an additional recurrence. Accumulated curvature also correlates with tangent-aligned gradient magnitude (Figure 2d). This association motivates curvature as a candidate gating signal; it does not establish that curvature causes harmful gradient amplification.

Ablations support tangent suppression with curvature-based gating. Table 4 compares TAGS with hard tangent projection, random-direction suppression, fixed or randomly sampled truncation depths, adaptive computation, and deep supervision on maze navigation. TAGS achieves the highest OOD average among these alternatives. The advantage is not uniform across splits: hard projection performs better on far-OOD grid size, and truncated BPTT performs better on path difficulty. Random-direction suppression also substantially reduces ID accuracy, so its failure alone does not isolate an effect on depth generalization.

To compare gating signals, we retain the same tangent direction and gate family $\gamma_t(z_t) = \gamma_{\max} z_t/(z_t + \beta_z)$, using step size, downstream state-gradient norm, or curvature as z_t. All three variants share the architecture, optimizer, training budget, and maximum suppression strength; each signal-specific scale is selected using the same ID-only validation budget. Curvature gating achieves an OOD average of 79.44%, compared with 57.78% for step-size gating and 46.81% for gradient-norm gating. This comparison supports curvature among the tested signals in the maze-navigation setting.

6 Conclusion

In this work, we studied test-time compute generalization in recurrent transformers through trajectory geometry, linking tangent-aligned state gradients to first-order depth sensitivity and characterizing how curvature enters its variation across horizons. Motivated by this analysis, we introduced Trajectory-Aligned Gradient Suppression (TAGS), a training-time intervention based on a curvature-weighted signal-filtering objective. TAGS attenuates the tangent-aligned component of each local backward proposal while preserving its transverse component and leaving inference unchanged. Across maze navigation, Sudoku, graph shortest path, and symbolic reasoning, TAGS achieves higher average OOD accuracy than BPTT, truncated BPTT, and fixed-point baselines.

REFERENCES

Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, Jiyoun Ha, Tal Schuster, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Aaron Courville, and Se-Young Yun. Mixture-of-recursions: Learning dynamic recursive depths for adaptive token-level computation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=QuqsEIVWIG.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Andrea Banino, Jan Balaguer, and Charles Blundell. Pondernet: Learning to ponder. In *8th ICML Workshop on Automated Machine Learning (AutoML)*, 2021. URL https://openreview.net/forum?id=1EuxRTe0WN.

Arpit Bansal, Avi Schwarzschild, Eitan Borgnia, Zeyad Emam, Furong Huang, Micah Goldblum, and Tom Goldstein. End-to-end algorithm synthesis with recurrent networks: Extrapolation without overthinking. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=PPjSKy40XUB.

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.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3563–3578, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.212. URL https://aclanthology.org/2024.findings-acl.212/.

Manfredo P. Do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976. ISBN 978-0132125895.

Jonas Geiping, Sean Michael McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=S3GhJooWIC.

Angeliki Giannou, Shashank Rajput, Jy-Yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11398–11442. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/giannou23a.html.

Alex Graves. Adaptive computation time for recurrent neural networks, 2017. URL https://arxiv.org/abs/1603.08983.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi

Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Zhiyuan He. Exploring depth generalization in large language models for solving recursive logic tasks, 2025. URL https://arxiv.org/abs/2512.02677.

Yeskendir Koishekenov, Aldo Lipani, and Nicola Cancedda. Encode, think, decode: Scaling test-time reasoning with recursive latent thoughts, 2025. URL https://arxiv.org/abs/2510.07358.

Samuele Marro, Davide Evangelista, X. Angelo Huang, Emanuele La Malfa, Michele Lombardi, and Michael J. Wooldridge. Language models are implicitly continuous. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SMK0f8JoKF.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang,

Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=Tsp2PL7-GQ.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

P.J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990. doi: 10.1109/5.58337.

Ronald J. Williams and David Zipser. *Gradient-based learning algorithms for recurrent networks and their computational complexity*, pp. 433–486. L. Erlbaum Associates Inc., USA, 1995. ISBN 0805812598.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=5Xc1ecxO1h.

Xiaojing Zhang, Haifeng Wu, Gang He, Jiyang Shen, Bochen Lyu, and Zhanxing Zhu. Modr: Mixture-of-depth-recurrent transformers for test-time reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=9Pba4rcQbE.

A Additional Details for the Geometric Analysis

This appendix describes the diagnostics in Figure 2 and proves the results in Section 3. We distinguish three statements: the first-order connection between the training-horizon BPTT state gradient and the loss under one additional recurrence (Eq. 3); a smoothness-controlled bound on tangent-aligned gradient propagation (Theorem 1); and a horizon-transfer decomposition that separates the change in depth sensitivity between neighboring horizons into speed-change, turning, and loss-nonlinearity contributions (Proposition 1). The last result characterizes how depth sensitivity varies across horizons; it does not establish that curvature causes harmful gradient amplification, that tangent suppression reduces this variation, or that curvature uniquely determines a suppression rule. A supplementary current-frame mixing bound is given in Appendix A.5.

A.1 Diagnostic Experimental Setup

The diagnostics in Figure 2 use recurrent models trained on maze navigation. The training distribution consists of 20×20 mazes with shortest-path difficulty in $[20, 50]$, where difficulty is the length of the ground-truth shortest path from start to goal. The training recurrence depth is $T_{\text{train}} = 32$. Parameters are frozen during evaluation and diagnostic gradient computation.

Depth extrapolation. Figure 2(a) evaluates an OOD split of 25×25 mazes with the same path-difficulty range $[20, 50]$. This comparison changes grid size while holding the stated path-difficulty range fixed. We sweep T_{test} and report accuracy, distinguishing improvements from saturation or degradation beyond the training depth.

Gradient consistency across recurrence depths. For Figure 2(b), let $g_T^{\text{BPTT}} = \nabla_\theta \mathcal{L}^{(T)}$ denote the ordinary parameter gradient at a fixed checkpoint. Gradients at different depths use the same parameters and matching examples from the held-out ID maze set. We report the stabilized cosine statistic

$$\text{Align}(T', T) = \frac{\langle g_{T'}^{\text{BPTT}}, g_T^{\text{BPTT}} \rangle}{\|g_{T'}^{\text{BPTT}}\| \, \|g_T^{\text{BPTT}}\| + \epsilon_{\text{align}}}, \qquad T = T_{\text{train}}, \tag{11}$$

where $\epsilon_{\text{align}} > 0$ stabilizes the denominator. A decrease in alignment indicates that the parameter-gradient direction depends on rollout depth. It does not identify which state-gradient component accounts for that difference. If a diagnostic instead uses a TAGS backward signal, the resulting parameter vector is a pseudogradient and must be distinguished from g_T^{BPTT}.

Tangent-aligned state gradients. For nonzero displacement $v_t = h_{t+1} - h_t$, write $s_t = \|v_t\|$, $\tau_t = v_t / s_t$, and $Q_t = I - \tau_t \tau_t^\top$. For a BPTT state gradient $\lambda_t = \lambda_t^{(T)}$, the exact orthogonal decomposition is

$$a_t = \langle \lambda_t, \tau_t \rangle, \qquad \lambda_t^{\parallel} = a_t \tau_t, \qquad \lambda_t^{\perp} = Q_t \lambda_t. \tag{12}$$

The ideal ratio discussed in the main text is $R_t = |a_t| / (\|\lambda_t^{\perp}\| + \epsilon_R)$, where $\epsilon_R > 0$ stabilizes the ratio. It measures relative component magnitudes, not the difference between gradients at two horizons.

The numerical diagnostic described here uses the stabilized direction $\bar{\tau}_t = v_t / (s_t + \epsilon) = \rho_t \tau_t$, with $\rho_t = s_t / (s_t + \epsilon)$. Its computed component and residual are

$$\hat{p}_t = \langle \lambda_t, \bar{\tau}_t \rangle \bar{\tau}_t = \rho_t^2 \lambda_t^{\parallel}, \qquad \hat{q}_t = \lambda_t - \hat{p}_t = \lambda_t^{\perp} + (1 - \rho_t^2) \lambda_t^{\parallel}. \tag{13}$$

Thus $\hat{q}_t$ is generally not orthogonal to the tangent. The stabilized proxy for Figure 2(c) is

$$R_t^{\epsilon} = \frac{\|\hat{p}_t\|}{\|\hat{q}_t\| + \epsilon_R} = \frac{\rho_t^2 |a_t|}{\sqrt{\|\lambda_t^{\perp}\|^2 + (1 - \rho_t^2)^2 a_t^2} + \epsilon_R}. \tag{14}$$

This proxy approaches the unit-tangent ratio as $\rho_t \to 1$, but the two need not agree near stationary states. The reported diagnostic is computed on the held-out ID set. Comparisons that use modified backward signals obey the same algebra with that signal in place of λ_t, but are not measurements of the true BPTT gradient.

Curvature–gradient association. Figure 2(d) compares accumulated numerical curvature with a stabilized tangent-alignment magnitude. The numerical quantities are

$$\hat{\kappa}_t = \frac{\|\bar{\tau}_{t+1} - \bar{\tau}_t\|}{s_t + \epsilon}, \qquad \hat{A}_t = |\langle \lambda_t, \bar{\tau}_t \rangle| = \rho_t |a_t|, \qquad 0 \le t \le T-2. \tag{15}$$

In particular, $\hat{A}_t$ differs from $\|\hat{p}_t\| = \rho_t^2 |a_t|$. The reported Pearson correlation is $r = 0.86$ on the held-out ID maze diagnostic. This is an association between the reported numerical statistics; it does not establish causation, identify harmful gradient components, or prove that the selected gate is optimal. Accumulated curvature and accumulated turning are also different quantities: summing $\hat{\kappa}_j$ differs from summing $(s_j + \epsilon)\hat{\kappa}_j = \|\bar{\tau}_{j+1} - \bar{\tau}_j\|$.

A.2 Proof of the Training-Depth-Sensitivity Expansion

Fix the parameters, context, initial state, and target, and write $F(h) = f_\theta(h, c)$ and $\psi(h) = \ell(q_\theta(h), y)$. The argument assumes the same deterministic recurrent map and readout at every depth. Define the remaining-computation loss $C_r(h) = \psi(F^r(h))$, with F^0 the identity. For $0 \le t < T$, shared recurrence gives the exact identities

$$C_{T-t}(h_t) = \mathcal{L}^{(T)}, \qquad C_{T-t}(h_t + v_t) = C_{T-t}(h_{t+1}) = \mathcal{L}^{(T+1)}. \tag{16}$$

Both inputs undergo the same $T - t$ remaining applications of F. The second input is already one recurrence farther along the trajectory, so it reaches h_{T+1}. This identity defines a theoretical comparison of losses; it does not require an extra rollout during TAGS training.

Suppose C_{T-t} is twice continuously differentiable on a neighborhood of $[h_t, h_{t+1}]$, with Hessian norm bounded by $H_{t,T}$ on that segment. Taylor's theorem with integral remainder yields

$$C_{T-t}(h_t + v_t) - C_{T-t}(h_t) = \langle \nabla C_{T-t}(h_t), v_t \rangle + R_{t,T}, \tag{17}$$

$$R_{t,T} = \int_0^1 (1-a)\, v_t^\top D^2 C_{T-t}(h_t + a v_t) v_t \, da, \qquad |R_{t,T}| \le \frac{H_{t,T}}{2} s_t^2. \tag{18}$$

By the chain rule, $\nabla C_{T-t}(h_t) = \lambda_t^{(T)}$, the true BPTT state gradient of the training-horizon terminal loss. Substitution of $v_t = s_t \tau_t$ proves Eq. 3:

$$\mathcal{L}^{(T+1)} - \mathcal{L}^{(T)} = s_t \langle \lambda_t^{(T)}, \tau_t \rangle + R_{t,T}. \tag{19}$$

The exact loss difference is independent of t, whereas its first-order estimate and remainder depend on t. The constant $H_{t,T}$ concerns the composed remaining-computation loss, depends on the horizon, and is distinct from the recurrent-map smoothness constant M. The first-order term can have either sign. It controls the sign of the exact loss difference only when its magnitude exceeds the remainder bound; the expansion alone does not establish that suppressing this component improves the task loss. At $v_t = 0$, the vector Taylor identity still holds, but the unit tangent is undefined.

Loss along a continued rollout. A separate expansion describes extension from depth T to $T' > T$. Let $\bar{\lambda}_j = \nabla \psi(h_j)$ be the readout-loss gradient at depth j. If $\|D^2 \psi(h)\| \le \bar{H}_j$ on $[h_j, h_{j+1}]$, Taylor expansion of each readout-loss increment and telescoping give

$$\mathcal{L}^{(T')} - \mathcal{L}^{(T)} = \sum_{j=T}^{T'-1} \langle \bar{\lambda}_j, v_j \rangle + \bar{R}_{T,T'}, \qquad |\bar{R}_{T,T'}| \le \frac{1}{2} \sum_{j=T}^{T'-1} \bar{H}_j s_j^2. \tag{20}$$

For positive displacements, each inner product equals $s_j \langle \bar{\lambda}_j, \tau_j \rangle$. This result concerns readout-loss gradients at the extended states, which generally differ from the BPTT gradients of the original training-horizon loss. It does not turn the one-loop sensitivity statement into a guarantee for arbitrary additional depth.

A.3 Tangent-Aligned State-Gradient Propagation

Section 3 summarizes how tangent-aligned credit is carried backward through BPTT. We state the result and its assumptions here.

Assumption 1 (Bounded Jacobian) *The recurrent state Jacobian satisfies* $\|J_t\| \le L$ *for* $0 \le t \le T-1$*, where* $L > 0$*.*

Assumption 2 (Local smoothness and nonzero motion) *The recurrent map* $f_\theta(\cdot, c)$ *is twice continuously differentiable on a neighborhood of each segment* $[h_t, h_{t+1}]$ *for* $0 \le t \le T-2$*, with* $\|D_h^2 f_\theta(h, c)\| \le M$ *throughout these segments. The second-derivative norm is the bilinear operator norm. The step sizes satisfy* $s_t > 0$ *for* $0 \le t \le T-1$*.*

The theorem requires neither a prescribed lower bound $s_t \ge \delta > 0$ nor a bound on adjacent speed ratios. Exactly stationary steps have undefined unit tangents and are excluded; arbitrarily small positive displacements remain covered.

Theorem 1 (An upper bound on tangent-aligned state-gradient propagation) *Under Assumptions 1 and 2, for* $0 \le t \le T-1$*,*

$$\|\lambda_{t,\parallel}^{(T)}\| \le L^{T-1-t}\|\lambda_{T-1,\parallel}^{(T)}\| + \frac{M}{2}\sum_{j=t}^{T-2} L^{j-t} s_j \|\lambda_{j+1,\perp}^{(T)}\|. \tag{21}$$

Here $\lambda_{j+1,\perp}^{(T)}$ *is transverse to* τ_{j+1}*, and the sum is empty when* $t = T-1$*.*

The bound separates two sources of tangent-aligned credit: credit inherited from the end of the rollout, and credit injected from downstream transverse components at a rate bounded by the smoothness constant M. Curvature does not appear in this bound, and the counterexample below shows that curvature alone cannot replace this smoothness control under the stated assumptions. Proposition 1 describes where curvature enters instead.

All vector norms are Euclidean; matrix norms are induced operator norms, and $\|D^2F\|$ is the bilinear operator norm. Write $\lambda_t = \lambda_t^{(T)}$, $P_t = \tau_t \tau_t^\top$, and $Q_t = I - P_t$. These projectors use unit tangents.

Step 1: Transport the trajectory displacement. For $0 \le t \le T-2$, Taylor expansion of F along $[h_t, h_{t+1}]$ gives

$$v_{t+1} = J_t v_t + R_t, \qquad R_t = \int_0^1 (1-a) D^2 F(h_t + a v_t)[v_t, v_t]\, da, \qquad \|R_t\| \le \frac{M}{2} s_t^2. \tag{22}$$

Here $F(h_t + v_t) - F(h_t) = h_{t+2} - h_{t+1} = v_{t+1}$. Since $Q_{t+1} v_{t+1} = 0$, division by $s_t > 0$ and projection yield

$$e_t := Q_{t+1} J_t \tau_t = -\frac{Q_{t+1} R_t}{s_t}, \qquad \|e_t\| \le \frac{M}{2} s_t. \tag{23}$$

Step 2: Bound one backward step. Decompose the transported tangent as

$$J_t \tau_t = b_t \tau_{t+1} + e_t, \qquad b_t = \langle \tau_{t+1}, J_t \tau_t \rangle, \qquad |b_t| \le \|J_t\| \le L. \tag{24}$$

Using BPTT and the orthogonality $e_t \perp \tau_{t+1}$, we obtain

$$\langle \lambda_t, \tau_t \rangle = \langle \lambda_{t+1}, J_t \tau_t \rangle \tag{25}$$

$$= b_t \langle \lambda_{t+1}, \tau_{t+1} \rangle + \langle \lambda_{t+1,\perp}, e_t \rangle. \tag{26}$$

The downstream parallel component drops out of the second term exactly. Therefore,

$$\|\lambda_{t,\parallel}\| \le L\|\lambda_{t+1,\parallel}\| + \frac{M}{2} s_t \|\lambda_{t+1,\perp}\|, \qquad 0 \le t \le T-2. \tag{27}$$

Step 3: Unroll to the final available tangent. Iterating Eq. 27 for $T-1-t$ steps gives Eq. 21. At $t = T-1$, the sum is empty and the statement is equality. At $t = T-2$, it is the one-step bound. The anchor uses λ_{T-1} and τ_{T-1}; it is not a projection of λ_T onto an unavailable τ_T.

No prescribed lower bound on positive speeds or bound on adjacent speed ratios is needed. The division by s_t in Eq. 23 cancels one power of s_t in the Taylor remainder. The theorem remains valid for arbitrarily small positive displacements, although exact stationarity makes the unit tangent undefined. The recursion analyzed here is for a terminal-loss objective; intermediate supervision or learned halting can introduce additional terms and is not covered by this statement without modification.

Why the smoothness term cannot be discarded. Consider $F(x, y) = (x + 1, y + x(x - 1))$ and $h_0 = (0, 0)$. Then $h_1 = (1, 0)$, $h_2 = (2, 0)$, $s_0 = s_1 = 1$, and $\tau_0 = \tau_1 = (1, 0)$, so $\kappa_0 = 0$. Nevertheless, $J_0\tau_0 = (1, -1)$ and $Q_1 J_0\tau_0 = (0, -1)$. Thus the transverse transport residual is nonzero despite zero discrete curvature. The recurrent Hessian has norm $M = 2$, and Eq. 23 is exact in this example. A curvature-only bound of the form $\|e_t\| \le C_\kappa \kappa_t s_t$ cannot hold under the stated assumptions.

A.4 Proof of Proposition 1

Fix $0 \le t \le T - 2$ with $s_t, s_{t+1} > 0$, and write $C = C_{T-t}$, $G = \nabla C(h_t) = \lambda_t^{(T)}$, $u = v_t$, and $w = v_t + v_{t+1} = h_{t+2} - h_t$. Because the recurrent operator is shared, $C(h_{t+i}) = \mathcal{L}^{(T+i)}$ for $i = 0, 1, 2$: starting from h_{t+i}, the remaining $T - t$ applications of F reach h_{T+i}. Hence

$$D_{T+1} - D_T = C(h_t + w) - 2C(h_t + u) + C(h_t). \tag{28}$$

As with Eq. 16, this comparison is theoretical and does not require the states h_{T+1} or h_{T+2} during training.

By Taylor's theorem with integral remainder, for $x \in \{u, w\}$,

$$C(h_t + x) = C(h_t) + \langle G, x\rangle + r(x), \qquad |r(x)| \le \frac{H}{2}\|x\|^2, \tag{29}$$

using $\|D^2 C\| \le H$ on the segments $[h_t, h_{t+1}]$ and $[h_t, h_{t+2}]$. Substituting,

$$D_{T+1} - D_T = \langle G, w - 2u\rangle + r(w) - 2r(u), \qquad w - 2u = v_{t+1} - v_t. \tag{30}$$

The displacement change splits exactly as

$$v_{t+1} - v_t = s_{t+1}\tau_{t+1} - s_t\tau_t = (s_{t+1} - s_t)\tau_{t+1} + s_t(\tau_{t+1} - \tau_t), \tag{31}$$

which gives Eq. 5 with $R_t' = r(w) - 2r(u)$. By definition of the discrete curvature, $\|\tau_{t+1} - \tau_t\| = \kappa_t s_t$, so the Cauchy–Schwarz inequality gives

$$\big|s_t\langle G, \tau_{t+1} - \tau_t\rangle\big| \le s_t\|\tau_{t+1} - \tau_t\|\,\|G\| = \kappa_t s_t^2\|G\|. \tag{32}$$

Finally, $\|w\| \le s_t + s_{t+1}$ and $\|u\| = s_t$ give $|R_t'| \le \frac{H}{2}(s_t + s_{t+1})^2 + H s_t^2$, which is the stated bound.

Leading-order structure of the turning term. Since τ_t and τ_{t+1} are unit vectors, $\langle \tau_t, \tau_{t+1} - \tau_t\rangle = -\frac{1}{2}\|\tau_{t+1} - \tau_t\|^2 = -\frac{1}{2}\kappa_t^2 s_t^2$. Decomposing $G = \langle G, \tau_t\rangle\tau_t + G^\perp$ relative to τ_t therefore gives

$$s_t\langle G, \tau_{t+1} - \tau_t\rangle = s_t\langle G^\perp, \tau_{t+1} - \tau_t\rangle - \tfrac{1}{2}\kappa_t^2 s_t^3\langle G, \tau_t\rangle, \tag{33}$$

so for small turning, the turning term is first order in the transverse component $G^\perp$ and only second order in the tangent change through the tangent-aligned coefficient $a = \langle G, \tau_t\rangle$.

Consequence for tangent suppression. Substituting a tangent-attenuated gradient $G_\gamma = G - \gamma a\tau_t$ with $0 \le \gamma \le 1$ into the turning term gives

$$s_t\langle G_\gamma, \tau_{t+1} - \tau_t\rangle = s_t\langle G, \tau_{t+1} - \tau_t\rangle + \tfrac{\gamma}{2}\kappa_t^2 s_t^3\, a. \tag{34}$$

The transverse contribution is unchanged, and attenuation can remove a cancellation between the two terms, so the magnitude of the turning term need not decrease. Proposition 1 is therefore a decomposition of how depth sensitivity varies across horizons, not a derivation of tangent suppression.

Speed change versus turning. Hold G fixed and consider the linearized increments $\hat{d}_t = s_t\langle G, \tau_t\rangle$ and $\hat{d}_{t+1} = s_{t+1}\langle G, \tau_{t+1}\rangle$. If $\tau_{t+1} = \tau_t$, then $\hat{d}_{t+1} = (s_{t+1}/s_t)\hat{d}_t$, so a speed change alone rescales the linearized increment by a positive factor and preserves its sign. Turning changes which components of G contribute and can reverse the sign. This statement concerns the linearized coefficients: the exact increments can still change sign through loss nonlinearity, even along a straight, constant-speed trajectory. The directional change is $\omega_t = \|\tau_{t+1} - \tau_t\| = \kappa_t s_t$; gating on the displacement-normalized quantity κ_t rather than ω_t is an additional modeling choice.

Scope of the remainder. The remainder R'_t captures nonlinearity of the loss along the path. It is not determined by the cached trajectory geometry, although given two additional forward recurrences it can be evaluated directly as

$$R'_t = \mathcal{L}^{(T+2)} - 2\mathcal{L}^{(T+1)} + \mathcal{L}^{(T)} - \langle G, v_{t+1} - v_t \rangle. \tag{35}$$

Its bound is $O(s_t^2)$ when adjacent step sizes are comparable and H is controlled. The turning term is also $O(s_t^2)$ under bounded curvature, but $O(s_t)$ when the turning angle stays fixed as the step size decreases; the stated assumptions do not select either regime. We therefore do not assume that R'_t is negligible, and the proposition does not establish that suppressing tangent-aligned credit improves extrapolation.

A.5 Supplementary Bound: Mixing in the Current Tangent Frame

Theorem 1 decomposes the incoming gradient relative to the next tangent τ_{t+1}. The following supplementary bound instead measures mixing relative to the current tangent τ_t. It is looser than Theorem 1 and is not used in the main argument; we include it because curvature appears in it through the choice of frame.

Lemma 1 (Curvature-dependent mixing in the current tangent frame) *Under Assumptions 1 and 2, define $Q_t = I - \tau_t \tau_t^\top$ and $m_t = Q_t J_t \tau_t$. For $0 \le t \le T-2$ and any incoming state-gradient vector w,*

$$|\langle \tau_t, J_t^\top Q_t w \rangle| = |\langle m_t, w \rangle| \le \left(L\kappa_t + \frac{M}{2} \right) s_t \|w\|. \tag{36}$$

Since $Q_t^\top = Q_t$, for any incoming vector w,

$$\langle \tau_t, J_t^\top Q_t w \rangle = \langle J_t \tau_t, Q_t w \rangle = \langle m_t, w \rangle. \tag{37}$$

The left side is the tangent-aligned output produced by an incoming signal projected transverse to the current tangent. Applying Eq. 24,

$$m_t = b_t Q_t \tau_{t+1} + Q_t e_t, \qquad |b_t| \le L, \qquad \|e_t\| \le \frac{M}{2} s_t. \tag{38}$$

Because $Q_t \tau_t = 0$ and $\|Q_t\| \le 1$,

$$\|Q_t \tau_{t+1}\| = \|Q_t(\tau_{t+1} - \tau_t)\| \le \|\tau_{t+1} - \tau_t\| = \kappa_t s_t. \tag{39}$$

Consequently,

$$|\langle m_t, w \rangle| \le \|m_t\| \, \|w\| \le \left(L\kappa_t + \frac{M}{2} \right) s_t \|w\|. \tag{40}$$

The coefficient $b_t = \tau_{t+1}^\top J_t \tau_t$ is an orthogonal-projection coefficient, not the adjacent speed ratio. Its bound follows directly from the Jacobian bound. This result neither orders the realized mixing across different trajectories nor establishes its effect on task loss.

B Derivation and Scope of TAGS

B.1 Curvature Weighting and the Local Objective

Fix a nondegenerate interior step $0 \le t \le T-2$. Define the local proposal $u_t = J_t^\top \tilde{\lambda}_{t+1}$ and unit-tangent projector $P_t = \tau_t \tau_t^\top$. Downstream filtering generally makes u_t different from $\lambda_t^{(T)}$. For any candidate signal z, the scalar $\langle z, \tau_t \rangle$ is its tangent-aligned coefficient. For $z = \lambda_t^{(T)}$, the scalar $s_t \langle z, \tau_t \rangle$ is the true first-order coefficient in Eq. 3; using the modified proposal generally gives a surrogate.

To motivate the linear curvature weight, let r denote arc length along a smooth curve, $\kappa_{\mathrm{cont}}(r) = \|d\tau/dr\|$, and Θ its accumulated turning. For a coefficient $a(r)$, the chosen turning-weighted penalty satisfies

$$\int a(r)^2 \, d\Theta = \int \kappa_{\mathrm{cont}}(r) a(r)^2 \, dr. \tag{41}$$

The choice to penalize per unit turning motivates a linear curvature weight; it is not uniquely imposed by Theorem 1 or Proposition 1. In the discrete setting, $\omega_t = \|\tau_{t+1} - \tau_t\| = \kappa_t s_t$ measures chordal tangent variation. If ϑ_t is the angle between the unit tangents, then $\omega_t = 2\sin(\vartheta_t/2)$, which approximates angular turning for small ϑ_t and is not an exact finite-angle identity with ϑ_t.

The local fidelity and turning penalty can be written as

$$\frac{\alpha s_t}{2}\|z - u_t\|^2 + \frac{\omega_t}{2}\langle z, \tau_t\rangle^2. \tag{42}$$

Dividing by $s_t > 0$ gives the objective in Eq. 9,

$$\mathcal{E}_t(z) = \frac{\alpha}{2}\|z - u_t\|^2 + \frac{\kappa_t}{2}\langle z, \tau_t\rangle^2, \qquad \alpha > 0. \tag{43}$$

Thus α has the same units as κ_t. The second term is a penalty on squared tangent-aligned magnitude. For a true BPTT signal, it is proportional to squared first-order depth sensitivity divided by s_t^2. Choosing this penalty does not establish that the tangent-aligned signal is harmful.

Closed-form minimizer. The Hessian is $\alpha I + \kappa_t P_t \succeq \alpha I$, so the objective is strongly convex. Its unique minimizer satisfies

$$(\alpha I + \kappa_t P_t) z_t^\star = \alpha u_t. \tag{44}$$

Multiplying by $Q_t = I - P_t$ and by P_t, respectively, gives

$$Q_t z_t^\star = Q_t u_t, \qquad P_t z_t^\star = \frac{\alpha}{\alpha + \kappa_t} P_t u_t. \tag{45}$$

Hence

$$z_t^\star = u_t - \frac{\kappa_t}{\alpha + \kappa_t} P_t u_t. \tag{46}$$

The optimization is over all vectors z; tangent-only correction and transverse preservation are consequences of the chosen objective rather than additional constraints on the solution.

Under-relaxation and suppression strength. Let $\Gamma = \gamma_{\max} \in [0, 1]$. The relaxed signal is

$$\tilde{\lambda}_t = (1 - \Gamma)u_t + \Gamma z_t^\star = (I - \gamma_t P_t)u_t, \qquad \gamma_t = \Gamma\frac{\kappa_t}{\alpha + \kappa_t}. \tag{47}$$

This recovers Eq. 6. The relaxed signal equals the minimizer when $\Gamma = 1$ or when the proposal is already optimal; otherwise it takes a partial step toward the minimizer. In particular, Γ is a relaxation factor and upper bound on suppression, not an additional constraint whose minimizer automatically gives the product above.

Writing $H_t = \alpha I + \kappa_t P_t$, the quadratic expansion at $z_t^\star$ is

$$\mathcal{E}_t(z) - \mathcal{E}_t(z_t^\star) = \frac{1}{2}(z - z_t^\star)^\top H_t (z - z_t^\star). \tag{48}$$

Since $\tilde{\lambda}_t - z_t^\star = (1 - \Gamma)(u_t - z_t^\star)$, we obtain

$$\mathcal{E}_t(u_t) - \mathcal{E}_t(\tilde{\lambda}_t) = \Gamma(2 - \Gamma)\big[\mathcal{E}_t(u_t) - \mathcal{E}_t(z_t^\star)\big], \tag{49}$$

proving Eq. 10. If the available objective reduction is positive, $\Gamma(2 - \Gamma)$ is the achieved fraction; otherwise both reductions vanish.

Local properties. For $0 \le \gamma_t \le 1$, orthogonality of P_t and Q_t implies

$$Q_t\tilde{\lambda}_t = Q_t u_t, \qquad P_t\tilde{\lambda}_t = (1 - \gamma_t)P_t u_t, \tag{50}$$

$$\|\tilde{\lambda}_t\|^2 = \|u_t\|^2 - \gamma_t(2 - \gamma_t)\|P_t u_t\|^2, \qquad \|\tilde{\lambda}_t - u_t\| = \gamma_t\|P_t u_t\|. \tag{51}$$

These are local statements relative to the modified incoming proposal u_t. They do not assert preservation of the true BPTT transverse component across the complete backward recursion or descent in the task loss. For the unit-tangent rule, $\Gamma = 0$ recovers BPTT, $\kappa_t = 0$ leaves the proposal unchanged, and $\Gamma = 1$ approaches hard projection only as $\kappa_t/\alpha \to \infty$.

B.2 Numerical Geometry, Boundaries, and Backward Ordering

The numerical implementation in Algorithm 1 caches $\bar{\tau}_t = v_t/(s_t+\epsilon)$ and $\hat{\kappa}_t = \|\bar{\tau}_{t+1} - \bar{\tau}_t\|/(s_t+\epsilon)$ and holds this geometry fixed during the modified backward pass. The structural training configurations use $\epsilon = 10^{-6}$. For an active gate, $\hat{\gamma}_t = \Gamma\hat{\kappa}_t/(\alpha+\hat{\kappa}_t)$. If $s_t > 0$, then $\bar{\tau}_t = \rho_t\tau_t$ and the implemented update is exactly

$$\tilde{\lambda}_t = u_t - \hat{\gamma}_t\bar{\tau}_t\bar{\tau}_t^\top u_t = u_t - \gamma_t^{\text{eff}} P_t u_t, \qquad \gamma_t^{\text{eff}} = \hat{\gamma}_t\rho_t^2. \tag{52}$$

It therefore preserves the true transverse component of u_t, with extra damping of the tangent correction. Equivalently, it is a $\Gamma\rho_t^2$-relaxed step toward the minimizer of the unit-tangent objective with weight $\hat{\kappa}_t$. The achieved objective-reduction factor for that objective is $\Gamma\rho_t^2(2-\Gamma\rho_t^2)$, rather than $\Gamma(2-\Gamma)$.

Numerical curvature is not identical to the unit-tangent curvature. For $s_t, s_{t+1} > 0$, direct expansion gives

$$\hat{\kappa}_t^2 = \frac{(\rho_{t+1}-\rho_t)^2 + \rho_t\rho_{t+1}\kappa_t^2 s_t^2}{(s_t+\epsilon)^2}. \tag{53}$$

Thus speed changes can produce nonzero numerical curvature even when the unit tangents agree. The ideal and stabilized rules approach one another when the displacement scales are large compared with ϵ; the zero-curvature statement for the ideal rule must not be transferred unchanged to the stabilized implementation.

At $s_t = 0$, the numerical direction and resulting correction are zero. The near-stationary safeguard additionally sets the gate to zero when $s_t < \delta_{\text{num}}$. This numerical cutoff is distinct from the theoretical nonzero-motion assumption, which imposes no prescribed positive lower bound on s_t. The training safeguard does not by itself specify which near-stationary samples are omitted or retained in a diagnostic plot.

With only $h_0, \ldots, h_T$ available, tangents are available for $0 \le t \le T-1$ and curvatures only for $0 \le t \le T-2$. We do not define a terminal curvature. Initialize $\tilde{\lambda}_T = \lambda_T^{(T)}$ and leave the final state VJP unfiltered, $\tilde{\lambda}_{T-1} = J_{T-1}^\top\tilde{\lambda}_T$. Earlier steps use Eq. 52. No additional forward state is required. At each step, both $u_t = J_t^\top\tilde{\lambda}_{t+1}$ and the parameter contribution $B_t^\top\tilde{\lambda}_{t+1}$ use the same incoming signal; only the state-VJP output is filtered. Filtering the incoming signal before both VJPs would generally give a different method. All parameter updates occur during training; inference uses frozen parameters and the usual forward recurrence.

B.3 Deviation from BPTT

We compare TAGS and BPTT on the same forward trajectory and terminal loss. Define the actual removed vector and its norm by

$$d_t = u_t - \tilde{\lambda}_t, \qquad \delta_t = \|d_t\|, \qquad d_{T-1} = 0. \tag{54}$$

For ideal unit tangents, $d_t = \gamma_t P_t u_t$ and $\delta_t = \gamma_t|\langle u_t, \tau_t\rangle|$. For the stabilized implementation, $d_t = \hat{\gamma}_t\bar{\tau}_t\bar{\tau}_t^\top u_t$, so $\delta_t = \gamma_t^{\text{eff}}|\langle u_t, \tau_t\rangle|$ when $s_t > 0$. These expressions use the actual local proposal, not the true BPTT gradient. Skipped gates have $d_t = 0$.

Proposition 2 (Deviation of the recurrent parameter pseudogradient from BPTT) *Assume $\|J_t\| \le L$ and $\|B_t\| \le \bar{B}$. For fixed initial state and context, let $g_T^{\text{BPTT,rec}} = \sum_{t=0}^{T-1} B_t^\top\lambda_{t+1}^{(T)}$ and $g_T^{\text{TAGS,rec}} = \sum_{t=0}^{T-1} B_t^\top\tilde{\lambda}_{t+1}$ denote the recurrent parameter contributions. Under the boundary convention above,*

$$\|g_T^{\text{BPTT,rec}} - g_T^{\text{TAGS,rec}}\| \le \bar{B}\sum_{j=1}^{T-2}\left(\sum_{k=0}^{j-1} L^k\right)\delta_j. \tag{55}$$

An index range with no elements contributes zero.

Write $\Delta_t = \lambda_t^{(T)} - \tilde{\lambda}_t$. The two backward recursions give

$$\Delta_t = J_t^\top\Delta_{t+1} + d_t, \qquad \Delta_T = 0, \qquad \Delta_{T-1} = 0. \tag{56}$$

For $j \geq i$, define $\Phi_{j,i} = J_{j-1} \cdots J_i$, with $\Phi_{i,i} = I$. Exact unrolling yields

$$\Delta_t = \sum_{j=t}^{T-2} \Phi_{j,t}^\top d_j, \qquad \|\Delta_t\| \leq \sum_{j=t}^{T-2} L^{j-t} \delta_j. \tag{57}$$

The parameter contribution at step t uses Δ_{t+1}, not Δ_t. Therefore,

$$g_T^{\mathrm{BPTT,rec}} - g_T^{\mathrm{TAGS,rec}} = \sum_{t=0}^{T-1} B_t^\top \Delta_{t+1} \tag{58}$$

$$= \sum_{t=0}^{T-3} \sum_{j=t+1}^{T-2} B_t^\top \Phi_{j,t+1}^\top d_j. \tag{59}$$

Taking norms and exchanging the finite sums gives

$$\|g_T^{\mathrm{BPTT,rec}} - g_T^{\mathrm{TAGS,rec}}\| \leq \bar{B} \sum_{t=0}^{T-3} \sum_{j=t+1}^{T-2} L^{j-t-1} \delta_j = \bar{B} \sum_{j=1}^{T-2} \left(\sum_{k=0}^{j-1} L^k \right) \delta_j, \tag{60}$$

as claimed.

For another exact form, define the forward parameter sensitivity $S_0 = 0$ and $S_{j+1} = J_j S_j + B_j$, with the initial state and context held fixed. Then Eq. 59 equals $\sum_{j=1}^{T-2} S_j^\top d_j$. For $T = 1$ or $T = 2$, the recurrent parameter deviation is zero under the stated boundary convention. Suppression at $t = 0$ changes the signal delivered to an upstream trainable initial-state module, but does not enter the recurrent parameter sum when h_0 is fixed.

Other trainable modules. A direct readout-parameter contribution is unchanged for the same forward trajectory and terminal cotangent. If $h_0 = h_0(\theta)$, its additional gradient deviation is $(D_\theta h_0)^\top \Delta_0$. If $c = c(\theta)$ enters the recurrence and $C_t = D_c f_\theta(h_t, c)$, the contribution through this context has deviation $(D_\theta c)^\top \sum_t C_t^\top \Delta_{t+1}$. These terms can be bounded using Eq. 57 and the corresponding Jacobian norms. Proposition 2 bounds the explicitly stated recurrent contribution and does not silently omit such paths from a claimed full-model bound.

B.4 Scope of the Local Guarantees

The deviation bound controls the difference from BPTT at one fixed horizon using the actual suppressed signals. It does not establish descent in the task loss, convergence of training, or consistency between pseudogradients at different horizons. The local objectives have different incoming proposals at different steps and do not establish that the complete TAGS parameter vector is the gradient of a single global scalar objective. Likewise, the state-gradient norm contraction proved above does not imply parameter-gradient descent or a guarantee for Adam updates. We make no general cross-horizon consistency claim based only on a contractive state Jacobian and bounded curvature.

State-level suppression need not preserve parameter descent. A scalar shared-parameter example makes this distinction explicit. Let $F_\theta(h) = 1 - h + \theta(1 + h/2)$, $h_0 = 0$, $T = 3$, and $\mathcal{L}^{(3)} = h_3^2/2$. At $\theta = 0$, the trajectory is $(0, 1, 0, 1)$, the tangents are $(1, -1, 1)$, and $\kappa_0 = \kappa_1 = 2$. The state Jacobians are $J_0 = J_1 = J_2 = -1$, and the direct parameter Jacobians are $(B_0, B_1, B_2) = (1, 3/2, 1)$. Choose the ideal rule with $\Gamma = 1$ and $\alpha = 2/3$, giving $\gamma_0 = \gamma_1 = 3/4$ while the final state VJP remains ungated. BPTT gives $\lambda_3 = 1$, $\lambda_2 = -1$, and $\lambda_1 = 1$, whereas TAGS gives $\tilde{\lambda}_3 = 1$, $\tilde{\lambda}_2 = -1$, and $\tilde{\lambda}_1 = 1/4$. Hence

$$g_T^{\mathrm{BPTT}} = 1 - \frac{3}{2} + 1 = \frac{1}{2}, \qquad g_T^{\mathrm{TAGS}} = \frac{1}{4} - \frac{3}{2} + 1 = -\frac{1}{4}. \tag{61}$$

The update $\theta \leftarrow \theta - \eta g_T^{\mathrm{TAGS}} = \eta/4$ increases the task loss by $\eta/8 + \mathcal{O}(\eta^2)$ for small positive η. Every local filter is positive and nonexpansive, yet the shared-parameter gradient reverses sign. The method's empirical benefit therefore requires experimental evidence beyond the local optimization and norm identities.

C LIMITATIONS

Our analysis assumes bounded state Jacobians and locally smooth recurrent dynamics on trajectory segments with nonzero displacement. Exactly stationary states have undefined unit tangents, although the numerical TAGS correction is zero at zero displacement. Theorem 1 bounds tangent-aligned gradient propagation through local smoothness, and Proposition 1 decomposes how depth sensitivity changes across neighboring horizons, with curvature entering through the turning term alongside step size, gradient magnitude, and speed change. Neither result establishes that curvature causes harmful gradient amplification, that tangent suppression reduces this variation, or that curvature uniquely determines the suppression signal; speed-change and unnormalized-turning (ω_t) gates are not evaluated. Curvature weighting is an inductive bias supported by the maze gating ablation; the local objective and transverse-preservation identities do not guarantee task-loss descent or improvement at arbitrary extrapolation depths. TAGS retains full-depth backpropagation and adds geometry computations, so its compute and memory costs remain those of full BPTT with additional overhead. Our experiments focus on tasks with explicit recurrent depth, and the symbolic results show smaller, mixed differences across datasets. Broader evaluation in large-scale language-model training and open-ended generation remains future work.

D COMPUTATIONAL ANALYSIS

TAGS modifies the training-time backward pass and leaves the forward recurrent computation unchanged. Given the cached states $h_0, \ldots, h_T$, write $v_t = h_{t+1} - h_t$ and $s_t = \|v_t\|$. The numerical implementation computes the stabilized trajectory direction

$$\bar{\tau}_t = \frac{v_t}{s_t + \epsilon}, \qquad 0 \le t \le T-1,$$

followed by numerical curvature and the suppression gate

$$\hat{\kappa}_t = \frac{\|\bar{\tau}_{t+1} - \bar{\tau}_t\|}{s_t + \epsilon}, \qquad \hat{\gamma}_t = \gamma_{\max} \frac{\hat{\kappa}_t}{\hat{\kappa}_t + \alpha}, \qquad 0 \le t \le T-2.$$

For an active gate, the state-gradient update is

$$u_t = J_t^\top \tilde{\lambda}_{t+1}, \qquad \tilde{\lambda}_t = u_t - \hat{\gamma}_t \langle u_t, \bar{\tau}_t \rangle \bar{\tau}_t.$$

Here u_t is the local proposal incorporating downstream filtering, rather than the true BPTT state gradient. The stabilized direction need not have unit length, so the removed vector is not an orthogonal projection onto a unit tangent. Geometry is computed once in a pre-backward sweep and held fixed during the modified backward pass. Curvature is unavailable at $t = T-1$, and the final state VJP is left ungated, as specified in Section 4. Each added geometry or filtering operation is linear in the number of state entries and requires no additional Jacobian-vector products. Parameter VJPs use the same incoming signals as their corresponding state VJPs; filtering is applied only to the state-VJP outputs.

Asymptotic complexity. Let d be the per-token hidden width, n the number of tokens, and $D = nd$ the number of scalar entries in a complete recurrent state, with batch size fixed. Let C_f and C_b denote the costs of one forward and backward recurrent step, respectively. Full BPTT over T recurrent steps has cost $T(C_f + C_b)$. TAGS adds trajectory geometry and state-gradient filtering, giving

$$T(C_f + C_b) + O(TD) = T(C_f + C_b) + O(Tnd).$$

For standard dense transformer blocks with a fixed MLP expansion ratio, the projection and MLP operations alone cost $O(nd^2)$ per step, in addition to attention operations. The $O(nd)$ geometry term therefore does not change the asymptotic order of full BPTT. This comparison concerns operation counts; the measured wall-clock overhead also depends on kernels, memory traffic, and the implementation.

Memory. Without activation checkpointing, TAGS uses the forward states already available during full BPTT. Caching stabilized directions for $0 \le t \le T-1$ and numerical curvatures and gates for $0 \le t \le T-2$ adds $O(TD)$ storage. The precise overhead depends on whether geometry is

Table 5: Training-time overhead of TAGS relative to full BPTT. Values are reported mean per-iteration timings on a single NVIDIA A40 GPU; relative overhead is $100(t_{\mathrm{TAGS}}/t_{\mathrm{BPTT}} - 1)$.

Setting	BPTT time / iter.	TAGS time / iter.	Relative overhead
Maze, $T = 32$, $d = 512$, batch 64	1.52	1.58	3.95%
Huginn, $T = 16$, batch 1	23.24	24.66	6.11%

computed per state or per token and which quantities are cached. TAGS requires neither explicit Jacobian matrices nor auxiliary networks; the cached geometry is detached from the computation graph. With activation checkpointing, geometry can instead be recomputed from rematerialized states, subject to the boundary requirements below. The asymptotic storage bound does not establish a particular percentage increase in peak GPU memory.

Comparison with baselines. When truncated BPTT propagates gradients only through the final $w < T$ recurrent steps, its approximate recurrent compute cost is

$$TC_f + wC_b.$$

This reduces backward compute and activation storage by omitting earlier steps from the differentiable suffix. It is specific to the final-window schedule, rather than to every possible implementation of truncated BPTT. TAGS retains full-depth backward propagation, with the additional $O(TD)$ geometry term, while selectively modifying tangent-aligned state-gradient components.

Wall-clock measurements. We benchmark per-iteration wall-clock time on the maze setting with $T = 32$, per-token width $d = 512$, and batch size 64, and on Huginn fine-tuning with $T = 16$ and batch size 1, using a single NVIDIA A40 GPU.

The measured overhead is approximately 4% on mazes and 6.1% on Huginn. This overhead is incurred only during training. At inference, the trained parameters are frozen and TAGS adds no geometry computation or gradient filtering to the usual forward recurrence. This comparison is at a fixed inference depth; increasing recurrence depth still increases the model's forward computation.

Compatibility with gradient checkpointing. We use activation checkpointing in the Huginn fine-tuning experiments. TAGS can preserve the same backward rule under checkpointing by caching or rematerializing the geometry needed at each gated step. Computing $\hat{\kappa}_t$ requires h_t, h_{t+1}, h_{t+2}, so a checkpoint segment boundary may require a state from the neighboring segment. Equivalent recomputation requires matching forward states, including consistent stochastic operations where present, and treating geometry as fixed during backward propagation. Subject to these conditions, checkpointing changes storage and recomputation costs rather than the specified TAGS pseudogradient, apart from numerical effects.

E Hyperparameters

Tables 6–9 report model and optimization settings. Main trained-model results are summarized over three seeds, while the maze ablation table reports separate point estimates. Structural models use $T_{\mathrm{train}} = 32$ and $T_{\mathrm{test}} = 64$ for the main comparisons; Huginn uses $T_{\mathrm{train}} = 16$ and $T_{\mathrm{test}} = 32$.

Hyperparameter and checkpoint selection. In the maze gating comparison, the tangent-suppression family, architecture, optimizer, training budget, and maximum suppression strength are shared. Signal-specific scales are selected with the same ID-only validation budget, as reported in the main text. OOD results are not used to choose those scales.

E.1 Dataset Splits and OOD Construction

Tables 10 and 11 summarize the reported datasets and evaluation groups. Structural splits vary task parameters while retaining the task format: maze splits vary shortest-path length or grid size, Sudoku compares the designated easy and hard puzzle sets, and graph shortest-path splits increase the number

Table 6: Shared model and optimization hyperparameters for structural reasoning experiments.

Hyperparameter	Value
Backbone	Recurrent transformer
Training recurrence depth	32
Main evaluation recurrence depth	64
Backpropagation steps for TAGS	32
Transformer layers	2
Hidden size per token	512
Attention heads	8
MLP expansion factor	4
Puzzle embedding dimension	512
Puzzle embedding length	16
Positional encoding	2D RoPE
Forward dtype	bfloat16
Optimizer learning rate	1×10^{-4}
Puzzle embedding learning rate	1×10^{-4}
Weight decay	1.0
Puzzle embedding weight decay	1.0
EMA	Enabled

Table 7: TAGS hyperparameters for structural reasoning experiments. All recurrent steps participate in backpropagation; curvature-based filtering is applied only where the required geometry is available, as in Algorithm 1.

Task	Dataset	Epochs	Batch size	$\gamma_{\max}$	ϵ	α
Maze navigation	`maze_20x20`	5,000	64	1.0	10^{-6}	0.01
Sudoku	`sudoku-id-easy`	50,000	64	0.2	10^{-6}	0.03
Graph shortest path	`graph-shortest-path-n10-d2-8`	50,000	64	1.0	10^{-6}	0.10

of nodes. These evaluations test performance under distribution shifts at the reported recurrence depths. Separate depth sweeps with each checkpoint held fixed assess the benefit of additional recurrence.

For mathematical reasoning, we follow the experimental setup of MoDR (Zhang et al., 2026). The main results group GSM8K and MAWPS under ID, and the remaining four benchmarks under OOD. These labels specify the reported aggregation groups; their interpretation as held-out distribution shifts depends on the actual training subset and evaluation exclusions. AQuA additionally differs in its multiple-choice answer format. For each task family, the reported OOD average is the unweighted mean across the designated evaluation columns; this aggregation does not itself establish that those columns contain unseen examples or unseen problem templates.

Table 8: Gradient-schedule hyperparameters for the main structural reasoning comparisons. The fixed-point baseline uses a one-step backward approximation.

Method	Forward recurrent steps	Backward steps
Full BPTT	32	32
Truncated BPTT	32	8
Fixed-point training	32	1
TAGS	32	32

Table 9: Shared fine-tuning hyperparameters for mathematical reasoning experiments.

Hyperparameter	Value
Base model	`huginn-0125`
Maximum training samples (cap)	10,000
Training recurrence depth	16
Main evaluation recurrence depth	32
Label type	Solution
Epochs	1
Per-device batch size	1
Gradient accumulation steps	4
Maximum sequence length	512
Learning rate	4×10^{-5}
LR scheduler	Cosine
Warmup ratio	0.1
Weight decay	1×10^{-4}
Max gradient norm	0.2
PEFT method	LoRA
LoRA rank	8
LoRA alpha	8
Save steps	500
Save total limit	1
Gradient checkpointing	Enabled

Table 10: **Structural reasoning dataset splits.** Reported training and evaluation counts. The principal shift axes are maze path length or grid size, Sudoku puzzle difficulty, and graph size.

Task	Split	Role	Split definition	Change relative to ID	# Samples
Maze navigation	ID	In-distribution evaluation	20×20 mazes; shortest-path length 20–50	Same grid size and path-length range as training	1,000
	Near OOD – Path difficulty	Difficulty extrapolation	20×20 mazes; shortest-path length 50–80	Longer paths with unchanged grid size	1,000
	Near OOD – Grid size	Size extrapolation	25×25 mazes; path-length range matched to ID	Larger grid with comparable path-length range	1,000
	Far OOD – Path difficulty	Difficulty extrapolation	20×20 mazes; shortest-path length 80–100	Longer paths with unchanged grid size	1,000
	Far OOD – Grid size	Size extrapolation	30×30 mazes; path-length range matched to ID	Larger grid-size shift	1,000
Sudoku	ID	In-distribution evaluation	Easy Sudoku puzzles	Same designated difficulty regime as training	1,000
	OOD – Hard	Difficulty extrapolation	Designated hard Sudoku puzzles with fewer givens	Reduced clue density	1,000
Graph shortest path	ID	In-distribution evaluation	Graphs with $n = 10$ nodes	Same graph size as training	1,000
	Near OOD	Size extrapolation	Graphs with $n = 12$ nodes	Larger graph size than training	1,000
	Far OOD	Size extrapolation	Graphs with $n = 14$ nodes	Larger graph-size shift than near OOD	1,000

Table 11: **Symbolic mathematical reasoning dataset splits.** For mathematical reasoning, we LoRA-finetune a pretrained Huginn model using the in-distribution training mixture and evaluate transfer to both in-distribution and out-of-distribution benchmarks. OOD benchmarks differ either in reasoning template, problem format, or answer format.

Dataset	Split	Role	Dataset / format	OOD change relative to ID	# Samples
GSM8K	ID	In-distribution evaluation	Same dataset family and answer format as fine-tuning	Same reasoning style and output format as training	1319
MAWPS	ID	In-distribution evaluation	Arithmetic word problems with numerical answers	Same broad arithmetic word-problem format as training	238
SingleEq	OOD	Reasoning / template shift	Single-equation algebraic word problems	Shift in problem template and symbolic structure	508
MultiArith	OOD	Reasoning / template shift	Multi-step arithmetic word problems	Shift toward different multi-step arithmetic templates	600
AddSub	OOD	Reasoning / template shift	Addition/subtraction word problems	Shift in arithmetic template distribution	395
AQuA	OOD	Output-format shift	Multiple-choice quantitative reasoning	Shift from free-form numerical answers to multiple-choice output format	254