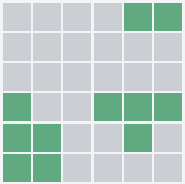
Adaptive Supervision Helps Tiny Recursive Models Learn Better

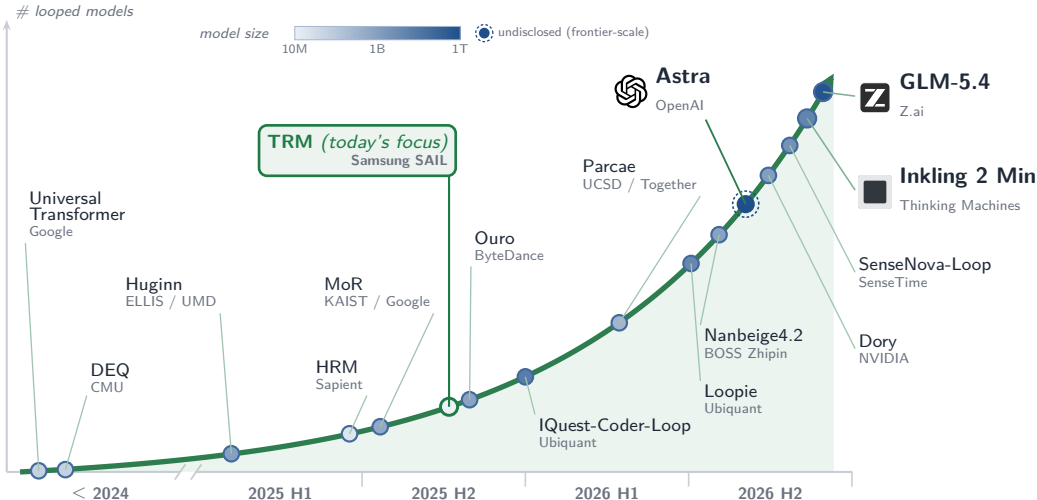
AS-TRM: supervise what the current reasoning state can support

Zhe Wang[1], Siqi Liu[2], Linyi Li[1]

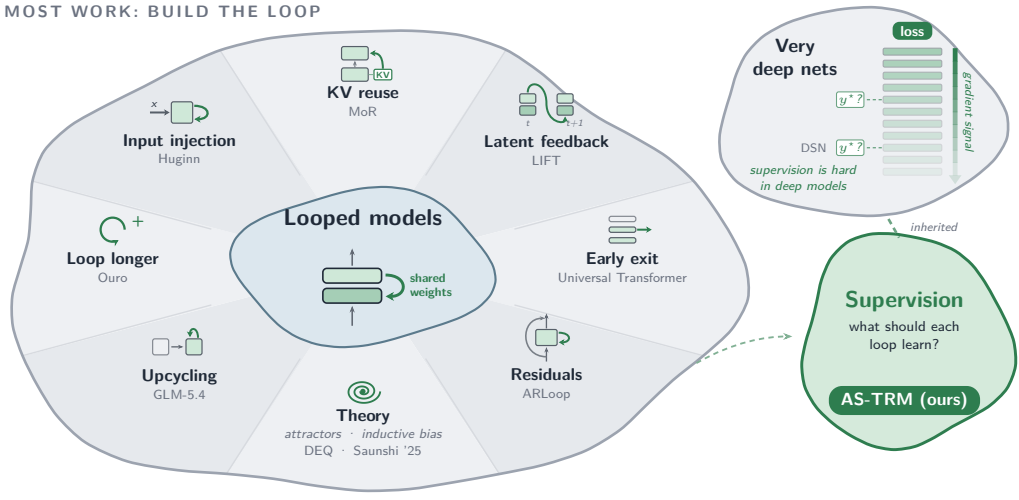
[1]Simon Fraser University [2]RBC Borealis



Looped models: from research to production

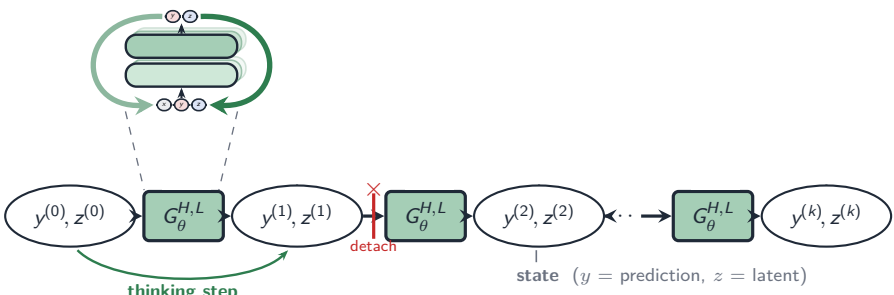


The research landscape of looped models



Others **build** the loop. We **supervise** it.

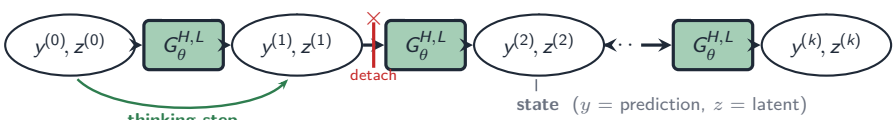
Background: tiny recursive models (TRM)



Background: tiny recursive models (TRM)

Gradient: detach

Between thinking steps, the state is detached (stop-gradient) — each loss only updates the step that produced it.



Why not unroll? Full backprop through the whole trajectory underperforms the detached baseline.

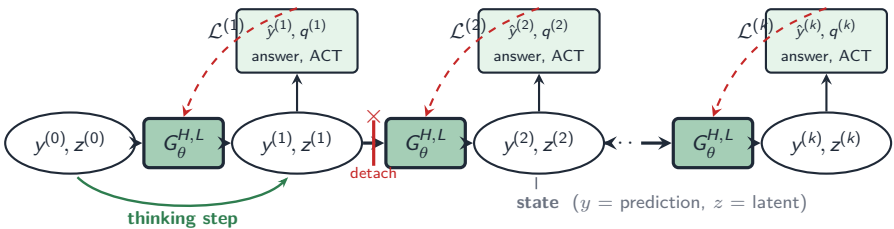
Background: tiny recursive models (TRM)

Gradient: detach

Between thinking steps, the state is detached (stop-gradient) — each loss only updates the step that produced it.

Supervision: full answer

Every step is supervised against the complete final answer $y^\star$ — the same dense target, every time.



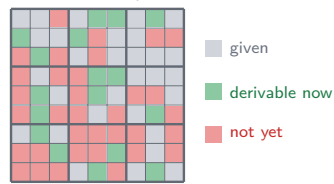
Why not unroll? Full backprop through the whole trajectory underperforms the detached baseline.

The problem: a state-supervision mismatch

Detach makes credit **step-local** — yet every step is asked for the **full** answer.

$$\ell^{(k)} = \underbrace{\sum_{u \in \text{derivable}} \ell_u^{(k)}}_{\text{state can support}} + \lambda \underbrace{\sum_{u \in \text{not yet}} \ell_u^{(k)}}_{\text{needs more steps}}$$

Sudoku at step k



AS-TRM: supervise the **focus set** $\mathrm{F} = \text{trusted} \cup \text{derivable}$;
$\lambda = 0$ removes the red terms, $0 < \lambda < 1$ down-weights them.

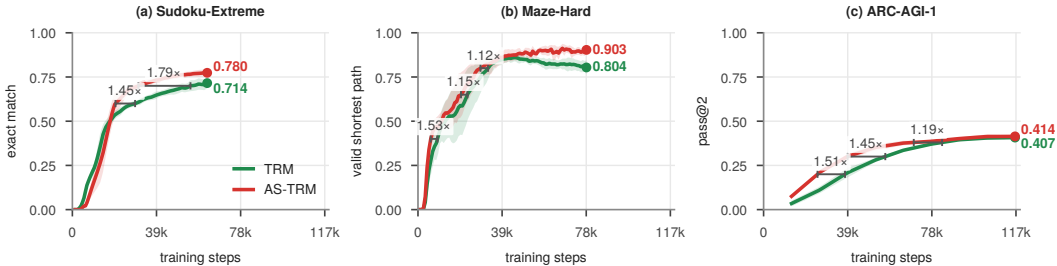
Guiding principle: place supervision near the model's current reasoning frontier, extend it to a bounded next frontier, and defer targets that need more computation.

Main results: higher accuracy on all three tasks

Task (H, L)	HRM	TRM	TRM+SPL	DIS	**AS-TRM**	Δ vs. 2nd best
Sudoku-Extreme (1,2)	46.4	59.9	59.5	38.0	**63.6**	**+3.7**
Sudoku-Extreme (2,4)	55.0	69.7	65.4	64.8	**77.6**	**+7.9**
Sudoku-Extreme (3,6)	61.6	71.4	69.5	69.2	**78.0**	**+6.6**
Maze-Hard (1,2)	81.0	86.8	86.9	72.0	**93.2**	**+6.3**
ARC-AGI-1 (3,4)	40.3	40.7	38.1	41.3	**41.4**	**+0.1**

Other curriculum-style (SPL) or diffusion-style (DIS) baselines don't consistently beat plain dense supervision (TRM) — AS-TRM does, especially on Sudoku-Extreme, and uniformly across every model size.

Faster training

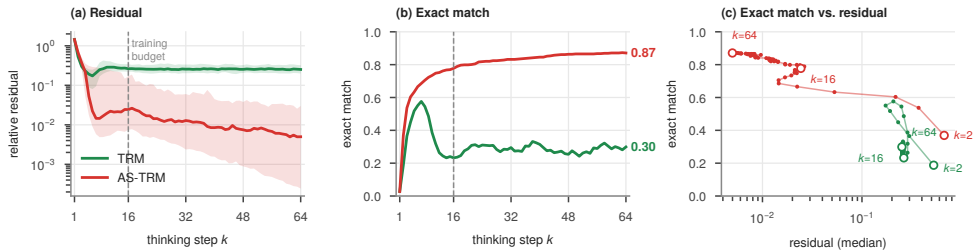


1.79×
Sudoku-Extreme (3,6), later in training

1.53×
Maze-Hard, early in training

1.51×
ARC-AGI-1, early in training

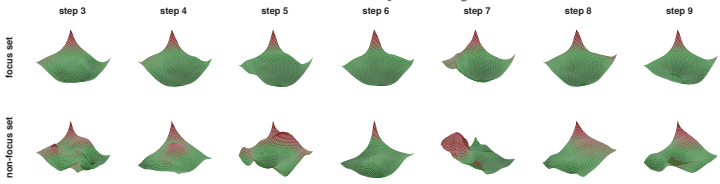
AS-TRM scales better



AS-TRM scales better beyond the training step budget: its residual keeps decreasing toward a smaller fixed point as thinking steps increase, while TRM's plateaus and its accuracy degrades.

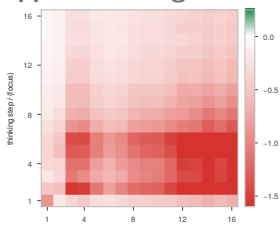
AS-TRM smooths the training

Focus loss landscape stays stable



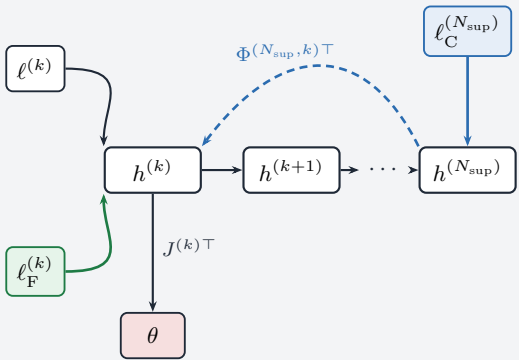
Sudoku-Extreme puzzle, thinking steps 3–9

Non-focus gradients oppose focus gradients



The non-focus set has larger curvature flips in the loss landscape — AS-TRM removes it.

Focus-set gradient is more aligned with the unrolled terminal gradient



$$J^{(k)} = \frac{\partial h^{(k)}}{\partial \theta}$$

$$\Phi^{(N_{\mathrm{sup}},k)} = \frac{\partial h^{(N_{\mathrm{sup}})}}{\partial h^{(k)}}$$

Jacobian of the final state w.r.t. step k

C: cells correct at the last step

$$g_{\mathrm{term}}^{(k)} = J^{(k)\top}\, \Phi^{(N_{\mathrm{sup}},k)\top}\, \nabla \ell_{\mathrm{C}}^{(N_{\mathrm{sup}})}$$

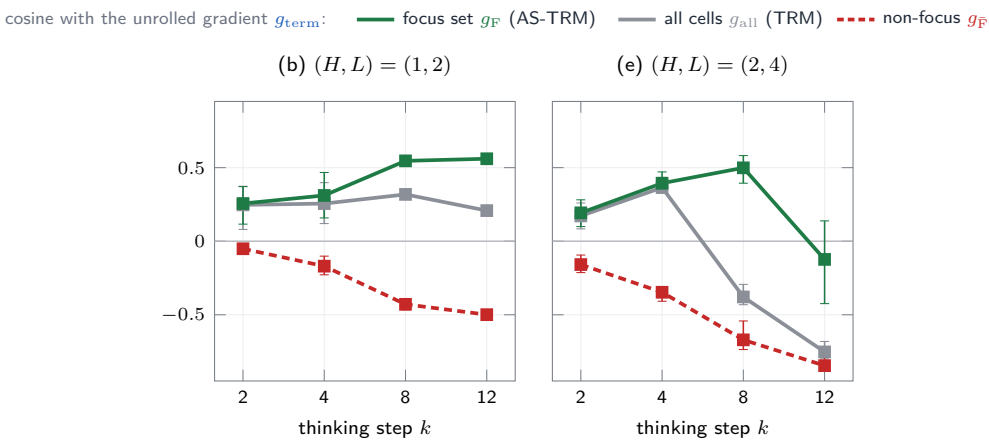
$$g_{\mathrm{all}}^{(k)} = J^{(k)\top}\, \nabla \ell^{(k)}$$

$$g_{\mathrm{F}}^{(k)} = J^{(k)\top}\, \nabla \ell_{\mathrm{F}}^{(k)}$$

$$g_{\bar{\mathrm{F}}}^{(k)} = g_{\mathrm{all}}^{(k)} - g_{\mathrm{F}}^{(k)}$$

Verify the hypothesis

Hypothesis (i) $\cos(g_{\mathrm{F}}, g_{\mathrm{term}}) > \cos(g_{\mathrm{all}}, g_{\mathrm{term}})$ (ii) $\cos(g_{\bar{\mathrm{F}}}, g_{\mathrm{term}}) < 0$



Conclusion

1. The mismatch

We identified a mismatch between state and supervision in TRM: step-local gradients paired with dense, full-target supervision force early steps to guess prematurely.

2. AS-TRM

We propose **AS-TRM** to address this. It works by adaptively supervising a focus set determined by the model's current state, expanding it with reasoning depth and training progress.

3. Why it works

We show gains in accuracy and speed, and our analysis suggests it works because the focus-set gradient aligns better with the unrolled terminal gradient.

Backup

focus-set construction · solver-free selector · future work

How the focus set and loss are defined

Focus set

recomputed at every thinking step k

$$\mathrm{F}^{(k)} = \mathrm{T}^{(k)} \cup \mathrm{R}^{(k)}$$

T: trusted positions; R: derivable frontier

Adaptive loss

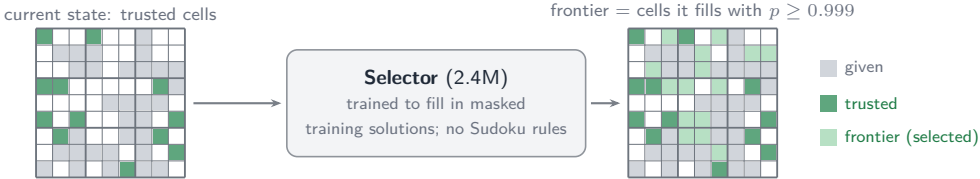
$$\mathcal{L}^{(k)} = \frac{\sum_{u \in \mathrm{F}} \ell_u + \lambda \sum_{u \notin \mathrm{F}} \ell_u}{|\mathrm{F}| + \lambda\,|\bar{\mathrm{F}}|}$$

$\lambda{=}0$ defers all non-focus supervision; $\lambda{=}1$ recovers dense TRM. Halting loss unchanged.

Task-specific construction

Task	**Trusted set** T	**Frontier** R	λ
Sudoku-Extreme	Correct, high-confidence cells	Cells derivable by constraint propagation	0
Maze-Hard	High-confidence shortest-path cells predicted as path	Shortest-path cells within $0.2d^\star$ layers of the trusted chains from both ends	0
ARC-AGI-1	Puzzles predicted entirely correctly	Unsolved puzzles with the lowest loss	0.11

Solver-free focus: a learned selector



the selector only decides where *the loss is computed — never* what *the target is*

	Sudoku (2,4)	Sudoku (3,6)	Maze (1,2)
TRM	69.7	71.4	86.8
AS-TRM, task rules	77.6	78.0	93.2
AS-TRM, learned selector	**74.5**	**79.1**	**91.3**

Using the model's own loss or confidence to pick cells does **not** help (51.5 / 59.3 / 65.7 on Sudoku (2,4)).

ARC-AGI-1 is already solver-free: its frontier is the unsolved puzzles with the lowest loss.

Future directions

Beyond puzzles

The same principle may apply across LRRMs more broadly, and to looped transformers and recursive language models — anywhere intermediate predictions are supervised through a limited gradient horizon.

The key challenge

It requires a reliable signal of *partial progress*. Task structure gives us this for Sudoku (constraint propagation) and Maze (shortest-path frontier), but many domains lack a readily verifiable frontier.

Extending to the language domain

The same idea could extend to LM-scale looped models like Huginn, Ouro, and (reportedly) OpenAI's Astra: adaptively choosing which tokens or positions to supervise at each recurrent step, instead of all of them.