UNROLLED POLICY ITERATION FOR TINY RECURSIVE MODELS

Bahram Behzadian* Meta
Brett Daley* Meta
Gopeshh Subbaraj Mila – Quebec AI Institute
Houssam Nassif Meta

ABSTRACT

We study recursive self-improvement via internal evaluators trained from verifier feedback, focusing on stability when recursive compute is scaled at test time. In plan-editing models such as Tiny Recursive Models (TRMs), an inner evaluator is unrolled for n recurrent steps to produce a value estimate; the unroll depth n is an architectural compute budget. We formalize plan editing as a Markov decision process and analyze approximate policy iteration with truncated internal evaluation. Under a contraction assumption on the inner recursion—requiring the latent update map to have Lipschitz constant $L_z < 1$—we decompose value error into an architectural Bellman-residual term plus a finite-unrolling bias that decays geometrically as L_z^n with depth. For conservative mixture updates that blend the current policy with a candidate at mixing weight α, we show that with statewise-centered advantage estimates, evaluation error enters the improvement bound scaled by α rather than at full strength, reducing sensitivity to imperfect evaluation. Experiments on Sudoku (4×4 and 9×9) trained solely from checker feedback validate feasibility and show that contraction strength and latent projection modulate stability under depth mismatch between training and evaluation. The analysis identifies the contraction modulus and unroll depth as practical stability controls for policy improvement under truncated internal computation.

1 INTRODUCTION

Recursive self-improvement requires that a model's internal evaluator remain stable as test-time compute is scaled. Tiny Recursive Models (TRMs) (Jolicoeur-Martineau, 2025) solve verifiable reasoning tasks by iteratively editing a candidate plan using a small recurrent evaluator, typically requiring ground-truth outputs with deep supervision. We investigate training TRM-style architectures *solely from checker feedback*—scalar signals from an automatic verifier indicating constraint satisfaction—without access to demonstrations. This setting arises whenever labeled solutions are expensive but verifiers are cheap (e.g., Sudoku constraints, unit tests, code compilation).

The stability problem. Training from checker feedback alone is challenging because the evaluator's value estimates are both *approximate* (limited network expressivity) and *computationally truncated* (finite unrolling of the inner recursion to depth n). Standard approximate policy iteration (API) and conservative policy iteration (CPI) bounds treat evaluation error as an opaque external quantity (Bertsekas & Tsitsiklis, 1996; Kakade & Langford, 2002). In TRMs, however, evaluation is performed by an internal dynamical system whose truncation depth n is a design choice—understanding how that truncation interacts with policy improvement requires making the dependency explicit.

Key idea. We formalize TRM plan editing as a discounted MDP over instance–plan pairs and analyze it through the lens of policy iteration with a compute-truncated value oracle. At each step, an inner evaluator runs a fixed number of recurrent updates to produce a latent representation, from which a value estimate and an edit policy are derived. We show that under a contraction assumption on the inner recursion, value error decomposes cleanly into an architectural residual plus a truncation bias that decays geometrically with depth, and that conservative mixture updates limit the impact of this error on policy improvement. Table 1 and Figure 1 summarize this mapping.

*Equal contribution. Correspondence to: buiksat@meta.com

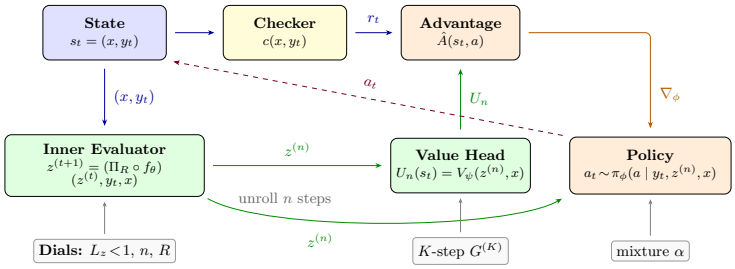

Figure 1: UPI–TRM architecture. Solid arrows: data flow; dashed arrow: edit action a_t updates plan $y_t \to y_{t+1}$. The inner evaluator unrolls n steps to produce latent $z^{(n)}$; the value head V_ψ yields $U_n(s)$. Checker reward r_t and advantages $\hat{A}$ drive policy-gradient updates; the new policy is a conservative mixture $\pi_{\text{new}}{=}(1{-}\alpha)\pi{+}\alpha\pi_{\text{cand}}$. **Legend:** *blue* boxes = TRM evaluator dials (L_z, n, R); *gray* boxes = standard algorithmic knobs (K, α).

Table 1: TRM edit loop as plan-space MDP. f_θ: latent update network; Π_R: projection onto radius-R ball; V_ψ: value head; $U_n(s)$: value after n unrolls; $L_z < 1$: contraction modulus; $s{=}(x, y)$: state; α: mixture weight. **Dials:** L_z, n. **Knobs:** K, α.

Inner evaluator (TRM)	
Latent unroll	$z^{(t+1)} = (\Pi_R \circ f_\theta)(z^{(t)}, y, x)$
Unrolled value	$U_n(s) = V_\psi(z^{(n)}(s), x)$
Bias	Dial $L_z < 1$; truncation error $\propto L_z^n$
Outer plan-space MDP	
State	$s = (x, y)$; instance x fixed per episode
Action	$a \in \mathcal{A}(y)$; edit $y' = \text{edit}(y, a; x)$
Reward	Checker + shaping; discount γ
Update	$\pi_{\text{new}} = (1 - \alpha)\pi + \alpha\pi_{\text{cand}}$

We call the TRM-specific evaluator parameters—contraction modulus and unroll depth—*dials*, and the standard API/CPI hyperparameters—bootstrap horizon and mixture weight—*knobs* (formalized in Section 2). Our bounds answer two questions from a recursive self-improvement perspective: (i) how do the evaluator dials govern internal evaluation accuracy? and (ii) how do conservative updates mitigate regressions when the evaluator is imperfect?

Contributions.

1. **Algorithm:** UPI–TRM, a policy-iteration template for TRMs trained from checker reward, combining K-step value regression, advantage-based improvement, and conservative mixture updates (Section 3).
2. **Value decomposition:** Under inner-loop contraction ($L_z < 1$), we decompose value error into an architectural Bellman residual plus a truncation bias that decays as L_z^n with unroll depth (Proposition 5.1).
3. **Conservative improvement:** For mixture updates with centered advantages, evaluation error enters the improvement bound scaled by the mixture weight α rather than at full strength (Theorem 5.7); approximate centering degrades gracefully via a centering defect term.

CPI (formalized in Section 2) is the natural template here because the TRM evaluator is inherently approximate; our contribution is making the *compute truncation* explicit, connecting TRM evaluator dials to CPI's safe-improvement thresholds.

2 Background and Setup

Approximate and conservative policy iteration. Approximate policy iteration (API) (Bertsekas & Tsitsiklis, 1996) alternates approximate evaluation and improvement; error propagation is governed by evaluation accuracy. Conservative policy iteration (CPI) (Kakade & Langford, 2002) restricts improvement to a mixture $\pi_{k+1} = (1{-}\alpha)\pi_k + \alpha\pi_{\text{cand}}$, yielding the bound $\eta(\pi_{k+1}) \geq L_{\pi_k}(\pi_{k+1}) -$

$\frac{2\gamma\varepsilon_{\mathrm{CPI}}}{(1-\gamma)^2}\alpha^2$ (where L_{π_k} is the local surrogate). This provides monotonic improvement that degrades gracefully with evaluation error, making CPI natural when the evaluator is approximate (as in TRMs with truncated unrolling).

Notation. $s = (x, y)$ denotes a state (instance–plan pair); $z \in \mathbb{R}^d$ is the latent; $z^{(n)}$ is the latent after n inner steps; z^* is the fixed point. L_z is the contraction modulus of the inner update; L_V is the Lipschitz constant of the value head. $\varepsilon^*_{\mathrm{res}}$, ε_A, $\varepsilon_{A,\mathrm{cand}}$, $\varepsilon_{\mathrm{cent}}$ denote error quantities defined in Section 5.

We formalize plan editing as an MDP and introduce the TRM evaluator structure.

Plan-space MDP. States $s = (x, y)$ pair an instance $x \in \mathcal{X}$ (e.g., a puzzle) with a candidate plan $y \in \mathcal{Y}$. Actions edit the plan: $y' = \mathrm{edit}(y, a; x)$. A deterministic checker provides a bounded score $c(x, y) \in [0, C_{\max}]$. We use potential-based shaping with potential $\Phi(s) = c(x, y)$, i.e., $r(s, a, s') = r_0(s, a, s') + \gamma\Phi(s') - \Phi(s)$, where r_0 is nonzero only on transitions into the absorbing state s_{abs}; we set $\Phi(s_{\mathrm{abs}}) = C_{\max}$ so $V^\pi(s_{\mathrm{abs}}) = -C_{\max}$ for all π (constant shift; advantages unchanged).

TRM evaluator. A Tiny Recursive Model maintains a latent $z \in \mathcal{Z} = \mathbb{R}^d$ updated via $z^{(t+1)} = f_\theta(z^{(t)}, y, x)$. After n inner steps (with optional projection Π_R), we define the unrolled value as $U_n(s) = V_\psi(z^{(n)}(s), x)$. The ideal (fixed-point) value is $U_*(s) = V_\psi(z^*(s), x)$ where z^* is the unique fixed point of the L_z-contractive map $z \mapsto (\Pi_R \circ f_\theta)(z, y, x)$.

Episodic-z vs. persistent-z. Our primary analysis uses *episodic-z*: the latent is reinitialized each edit step, giving Markov dynamics on (x, y). A persistent-z extension (carrying z across edits under a slow-drift assumption) is possible but not analyzed here.

Advantage-evaluation closure. Conservative improvement requires value accuracy on on-policy states and one-step successors under candidate actions. We write $\|\cdot\|^{\mathrm{adv}}_{\infty,\pi,\pi_{\mathrm{cand}}}$ for sup-norms over the closure $\mathcal{R}^{\mathrm{adv}}_{\pi,\pi_{\mathrm{cand}}} \supseteq \mathcal{R}_\pi$ that additionally contains these successors and is closed under K-step π-rollouts. Under this closure, $\mathcal{T}^\pi_K$ is a γ^K-contraction.

Key assumptions (used in bounds).

Assumption 2.1 (Bounded rewards)**.** Shaped rewards satisfy $|r(s, a, s')| \le R_{\max}$ for all transitions.

Assumption 2.2 (Bounded value head)**.** There exists $V_{\max} < \infty$ such that $|V_\psi(z, x)| \le V_{\max}$ for all z in the forward-invariant region $\mathcal{Z}_{\mathrm{inv}}$.

We assume the value head is L_V-Lipschitz in z:

$$\left|V_\psi(z, x) - V_\psi(z', x)\right| \le L_V \, \|z - z'\| . \tag{1}$$

Proposition 2.3 (Fixed point and finite-unrolling bias)**.** *Under Assumption 4.1, the projected recursion has a unique fixed point z^*, with $\|z^{(n)} - z^*\| \le \frac{L_z^n}{1-L_z} C_z$ where C_z is as defined in Assumption 4.1.*

3 Algorithm Sketch

We alternate three phases: (i) K-step bootstrapped value regression for U_n, (ii) a policy improvement step producing a candidate π_{cand}, and (iii) a conservative mixture update $\pi_{\mathrm{new}} = (1-\alpha)\pi + \alpha\pi_{\mathrm{cand}}$ (with optional distillation).

Phase 1: Value regression. We form K-step rollouts and define a target with target network $V_{\bar\psi}$:

$$G^{(K)}(s_0) = \sum_{k=0}^{K-1} \gamma^k r_k + \gamma^K \bar V(s_K). \tag{2}$$

where $\bar V(s) := -C_{\max}$ if $s = s_{\mathrm{abs}}$, and otherwise $\bar V(s) = V_{\bar\psi}(z^{(n)}(s), x)$. We minimize the bootstrapped value loss $\mathcal{L}_{\mathrm{val}}(\psi, \theta) = \mathbb{E}_{s_0\sim\mu}\big[(V_\psi(z^{(n)}(s_0), x_0) - \mathrm{sg}(G^{(K)}(s_0)))^2\big]$.

Phase 2: Advantage estimation. We define $\widehat{Q}_V(s, a) = \mathbb{E}_{s'}[r(s, a, s') + \gamma V(s')]$ and the centered advantage $\widehat{A}_V(s, a) = \widehat{Q}_V(s, a) - B_V(s)$, where $B_V(s) = \mathbb{E}_{b\sim\pi}[\widehat{Q}_V(s, b)]$ yields exact statewise centering (Theorem 5.7).

Phase 3: Conservative update. Form the candidate π_{cand} via policy gradient on $\widehat{A}$, then mix: $\pi_{\text{new}} = (1-\alpha)\pi + \alpha\pi_{\text{cand}}$. Distill back to a single network: $\pi_\phi \leftarrow \arg\min_\phi \mathrm{KL}(\pi_{\text{new}} \| \pi_\phi)$.

Remark 3.1 (Theory–implementation gap). Theorem 5.7 applies to the explicit mixture $\pi_{\text{new}}{=}(1{-}\alpha)\pi{+}\alpha\pi_{\text{cand}}$, not to the distilled policy π_ϕ. Distillation introduces approximation error not covered by our bounds. In experiments $\mathrm{KL}(\pi_{\text{new}} \| \pi_\phi)$ stays below 10^{-3}; by Pinsker's inequality, per-state TV distance is below 0.03. The bounds should be read as structural guidance for how (L_z, n, α) affect safe improvement, not as an end-to-end certificate for the implemented algorithm.

Dials and knobs. Our bounds expose two TRM evaluator dials—L_z (contraction modulus) and n (unroll depth)—controlling truncation bias, and two standard algorithmic knobs—K (bootstrap horizon) and α (mixture weight)—controlling residual amplification and conservativeness.

The theory in Sections 4–5.2 analyzes a single evaluation→improvement step of this template under idealized assumptions.

4 Stability via Contraction of the Inner Evaluator

The Bellman operator $\mathcal{T}^\pi$ and its K-step version $\mathcal{T}_K^\pi$ are standard (Sutton & Barto, 2018; Puterman, 1994):

$$(\mathcal{T}_K^\pi V)(s) \;:=\; \mathbb{E}_\pi\Big[\textstyle\sum_{t=0}^{K-1} \gamma^t r_t + \gamma^K V(s_K) \;\Big|\; s_0 = s\Big]. \tag{3}$$

The K-step operator is a γ^K-contraction with unique fixed point V^π. We train the unrolled value U_n by K-step bootstrapped regression, minimizing $\mathcal{L}_{\text{val}}(\psi, \theta) = \mathbb{E}_{s_0 \sim \mu}[(V_\psi(z^{(n)}(s_0), x_0) - \mathrm{sg}(G^{(K)}(s_0)))^2]$ with $G^{(K)}$ as defined in (2). We do not derive a general $L^2(\mu) \to \|\cdot\|_\infty$ conversion; our bounds should be read as conditional on controlling the sup-norm residuals over the relevant closure set.

We distinguish *Bellman-operator contraction* ($\mathcal{T}_K^\pi$ is a γ^K-contraction—a standard MDP fact) from *inner evaluator contraction* (the latent update $z \mapsto \tilde{f}_\theta(z, y, x)$ is an L_z-contraction with $L_z < 1$—an architectural constraint). Contractive $z \to z$ layers do *not* imply TD convergence; they control truncation bias within each forward pass.

Assumption 4.1 (Forward-invariant contraction)**.** There exists a closed bounded set $\mathcal{Z}_{\text{inv}} \subseteq \mathcal{Z}$ such that the projected update map $\tilde{f}_\theta(z, y, x) := (\Pi_R \circ f_\theta)(z, y, x)$ maps $\mathcal{Z}_{\text{inv}}$ to itself and is an L_z-contraction in z with $L_z \in (0, 1)$. Assume the initializer satisfies $z_{\text{init}}(x, y) \in \mathcal{Z}_{\text{inv}}$ for all (x, y). Let $z^*(x, y)$ denote the unique fixed point and define $C_z := \sup_{(x,y)} \|\tilde{f}_\theta(z_{\text{init}}(x, y), y, x) - z_{\text{init}}(x, y)\|$, and assume $C_z < \infty$.

We enforce Assumption 4.1 via operator-norm clamping on $z \to z$ layers plus per-layer scaling; projection Π_R ensures forward-invariance. Value-head spectral normalization is disabled in stability-dial experiments to avoid confounds.

Scope of contraction analysis. The contraction regime analyzed here is a *sufficient condition* for the stability and value-decomposition guarantees in Proposition 5.1—it is not a claim that TRM requires contraction to function. The original TRM work (Jolicoeur-Martineau, 2025) emphasizes that fixed-point convergence is not required for the model to reason effectively. Our analysis targets the contractive subset of parameter space where rigorous bounds hold, and treats contraction as a tunable stability dial that practitioners can enforce (or relax) based on their depth-mismatch tolerance.

Remark 4.2 (Generality beyond TRMs). The results apply to *any* architecture with (i) a recurrent inner loop whose Lipschitz constant can be bounded below 1, and (ii) a value head on the truncated latent—e.g., iterative refinement models, scratchpad reasoners, or Deep Equilibrium Models (Bai et al., 2019). TRMs are a concrete instantiation.

Trade-off in choosing L_z. Smaller L_z yields faster bias decay and greater depth-mismatch stability, but restricts evaluator expressivity (latent dynamics cannot locally expand). Larger L_z preserves expressivity but requires deeper unrolling. Our experiments illustrate: contraction ($L_z{=}0.9$) trades a modest success drop (90.7% vs. 93.3%) for $4.1\times$ lower Δ_V and $33\times$ lower Δ_π at $8\times$ depth mismatch. Practitioners should treat L_z and n as coupled dials.

4.1 SCOPE OF GUARANTEES

Our results are single-step API/CPI error-propagation bounds under idealized contraction/Lipschitz assumptions; we do not prove end-to-end SGD/TD convergence. The bounds provide structural guidance on how (L_z, n, K, α) affect truncation error and conservative improvement.

5 ERROR BOUNDS AND CONSERVATIVE IMPROVEMENT

We state results for episodic-z under Assumption 4.1 and bounded/Lipschitz value heads.

5.1 VALUE ERROR BOUNDS (EPISODIC-z)

We first define the architectural residual that will appear in our bounds:

$$\varepsilon^*_{\text{res}} \triangleq \|U_* - \mathcal{T}^\pi_K U_*\|^{\text{adv}}_{\infty,\pi,\pi_{\text{cand}}}. \tag{4}$$

This measures how well the ideal fixed-point value U_* satisfies the Bellman equation. While $\varepsilon^*_{\text{res}}$ depends on the ideal fixed-point U_*, it can be upper-bounded using the measurable residual of U_n plus the truncation gap; see Lemma 5.4.

Proposition 5.1 (Value-error decomposition)**.** *Let $\gamma \in (0,1)$ and $K \geq 1$.*

1. *For any bounded $V : \mathcal{S}_{\text{plan}} \cup \{s_{\text{abs}}\} \to \mathbb{R}$, by the γ^K-contraction of $\mathcal{T}^\pi_K$ on $\mathcal{R}^{\text{adv}}_{\pi,\pi_{\text{cand}}}$ (which holds under the K-step π-closure condition (iii) in the definition of $\mathcal{R}^{\text{adv}}_{\pi,\pi_{\text{cand}}}$; Setup):*

$$\|V - V^\pi\|^{\text{adv}}_{\infty,\pi,\pi_{\text{cand}}} \leq \frac{1}{1-\gamma^K} \|V - \mathcal{T}^\pi_K V\|^{\text{adv}}_{\infty,\pi,\pi_{\text{cand}}}. \tag{5}$$

2. *Under Assumption 4.1, Assumption 2.2, and Lipschitz value head* (1)*, for the unrolled value $U_n(s) = V_\psi(z^{(n)}(s), x)$ with initialization $z^{(0)} = z_{\text{init}}(x, y)$ (where U_* is bounded):*

$$\|U_n - V^\pi\|^{\text{adv}}_{\infty,\pi,\pi_{\text{cand}}} \leq \underbrace{\frac{\varepsilon^*_{\text{res}}}{1-\gamma^K}}_{\textit{arch.}} + \underbrace{L_V \frac{L_z^n}{1-L_z} C_z}_{\textit{unroll.}}. \tag{6}$$

Remark 5.2 (Episodic-z fixed point). In episodic-z, z is reinitialized each edit step; $U_*(s) = V_\psi(\lim_{n\to\infty}(\Pi_R \circ f_\theta)^n(z_{\text{init}}, y, x), x)$ is an *analysis-only reference*—never realized in practice. The residual $\varepsilon^*_{\text{res}}$ is likewise a structural diagnostic, not an optimization target; the decomposition (6) separates architectural expressivity from truncation effects.

Proof sketch: (i) Standard Bellman contraction. (ii) Triangle inequality: $\|U_n - V^\pi\| \leq \|U_* - V^\pi\| + \|U_n - U_*\|$; apply (i) and Lipschitz bound with Prop. 2.3.

Remark 5.3 (Practical L_V control). The Lipschitz bound L_V is an idealized assumption. In our experiments, we disable value-head spectral normalization to avoid training instability, so L_V may be large in practice. **Implication:** When L_V is uncontrolled, the bound (6) provides only qualitative structural guidance (showing *which* quantities matter), not quantitative guarantees. Practitioners should monitor empirical value instability Δ_V as a proxy.

Value-to-advantage error. The one-step centered advantage estimator $\widehat{A}_V(s,a) = \widehat{Q}_V(s,a) - \mathbb{E}_{b\sim\pi}[\widehat{Q}_V(s,b)]$ satisfies $|\widehat{A}_V(s,a) - A^\pi(s,a)| \leq 2\gamma \|V - V^\pi\|^{\text{adv}}_{\infty,\pi,\pi_{\text{cand}}}$, where the factor 2γ arises because $\widehat{Q}_V$ and the baseline each contribute a $\gamma\|V - V^\pi\|$ term through the successor value. Combining with (6) yields the full advantage-error bound.

This decomposition is central: the first term $\varepsilon^*_{\text{res}}$ reflects how well the *ideal* fixed-point value U_* can approximate V^π (a fixed-point Bellman residual at the learned evaluator parameters, reflecting representation + optimization), while the second term is finite-unrolling bias that decays geometrically with n.

Lemma 5.4 (Residual transfer)**.** *For any bounded U_n, U_*:*

$$\begin{aligned}\varepsilon^*_{\text{res}} \leq\ & \|U_n - \mathcal{T}^\pi_K U_n\|^{\text{adv}}_{\infty,\pi,\pi_{\text{cand}}} \\ & + (1+\gamma^K) \|U_n - U_*\|^{\text{adv}}_{\infty,\pi,\pi_{\text{cand}}}.\end{aligned} \tag{7}$$

Proof sketch (Lemma 5.4): Write $U_* - \mathcal{T}_K^\pi U_* = (U_* - U_n) + (U_n - \mathcal{T}_K^\pi U_n) + \mathcal{T}_K^\pi(U_n - U_*)$; take $\|\cdot\|_{\infty,\pi,\pi_{\text{cand}}}^{\text{adv}}$; apply the triangle inequality and γ^K-contraction of $\mathcal{T}_K^\pi$ to obtain $(1+\gamma^K)\|U_n - U_*\|_{\infty,\pi,\pi_{\text{cand}}}^{\text{adv}}$.

Corollary 5.5 (Operational value-error bound)**.** *Under the conditions of Proposition 5.1, combining Lemma 5.4 with the truncation bound yields*

$$\|U_n - V^\pi\|_{\infty,\pi,\pi_{\text{cand}}}^{\text{adv}} \le \frac{1}{1-\gamma^K}\,\|U_n - \mathcal{T}_K^\pi U_n\|_{\infty,\pi,\pi_{\text{cand}}}^{\text{adv}} + \Big(\frac{1+\gamma^K}{1-\gamma^K} + 1\Big)\,L_V\,\frac{L_z^n}{1-L_z}\,C_z.$$

From value error to advantage error. The same value-error bound implies an advantage-error bound (up to a constant factor depending on the estimator).

This makes explicit how (L_z, n, K) and architectural factors shape advantage error. The architectural residual $\varepsilon_{\text{res}}^*$ depends on the expressivity of (f_θ, V_ψ), while the unrolling term is controlled by the dials (L_z, n).

5.2 Conservative policy improvement

For conservative mixture updates $\pi_{\text{new}} = (1-\alpha)\pi + \alpha\pi_{\text{cand}}$, the standard CPI bound (Kakade & Langford, 2002) guarantees monotonic improvement up to an $O(\alpha^2)$ penalty. The $O(\alpha)$ scaling for mixture policies is standard in CPI (Kakade & Langford, 2002); our contribution is combining it with the value decomposition (Proposition 5.1) to make explicit how TRM evaluator dials (L_z, n) affect the safe-improvement threshold. With centered advantage estimators (which hold exactly for discrete action spaces with exact baseline computation), evaluation error contributes as $O(\alpha\,\varepsilon_{A,\text{cand}})$ rather than $O(\varepsilon_A)$, and the value decomposition shows how (L_z, n) and (K, α) together shape safe improvement.

Assumption 5.6 (Uniform advantage error on states visited by π)**.** For some $\varepsilon_A \ge 0$:

$$|\widehat{A}(s,a) - A^\pi(s,a)| \le \varepsilon_A \tag{8}$$

for all (s,a) with $s \in \mathcal{R}_\pi$ and $a \in \text{supp}(\pi_{\text{cand}}(\cdot|s)) \cup \text{supp}(\pi(\cdot|s))$.

The uniform advantage error ε_A is a structural placeholder subsuming architectural residual, finite-sample error, and coverage factors. In practice, we monitor empirical TD errors; the guarantees provide structural guidance.

Warning: Converting $L^2(\mu)$ training loss to sup-norm bounds over $\mathcal{R}_\pi$ requires concentrability coefficients $C_\mu := \sup_s d_\pi(s)/\mu(s)$ that can be exponentially large in state dimension; our bounds should be interpreted as structural guidance rather than quantitative guarantees in high-dimensional settings.

Theorem 5.7 (CPI with centered evaluation error)**.** *Let $\pi_{\text{new}} = (1-\alpha)\pi + \alpha\pi_{\text{cand}}$ with $\alpha \in [0,1]$ and suppose Assumption 5.6 holds. Assume the advantage estimator is* statewise-centered under π*:*

$$\mathbb{E}_{a\sim\pi(\cdot|s)}[\widehat{A}(s,a)] = 0 \qquad \textit{for all } s \in \mathcal{R}_\pi. \tag{9}$$

We define the candidate-policy bias in advantage estimation

$$\varepsilon_{A,\text{cand}} \triangleq \sup_{s\in\mathcal{R}_\pi} \Big|\mathbb{E}_{a\sim\pi_{\text{cand}}(\cdot|s)}\Big[\widehat{A}(s,a) - A^\pi(s,a)\Big]\Big|. \tag{10}$$

Then:

$$\eta(\pi_{\text{new}}) \ge \widehat{L}_\pi(\pi_{\text{new}}) - \frac{\alpha}{1-\gamma}\,\varepsilon_{A,\text{cand}} - \frac{2\varepsilon_{\text{CPI}}\,\gamma}{(1-\gamma)^2}\,\alpha^2, \tag{11}$$

where $\widehat{L}_\pi(\pi') = \eta(\pi) + \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\pi, a\sim\pi'(\cdot|s)}[\widehat{A}(s,a)]$ and $\varepsilon_{\text{CPI}} := \max_s \big|\mathbb{E}_{a\sim\pi_{\text{cand}}(\cdot|s)}[A^\pi(s,a)]\big|$.

Implemented regime. The exact centering condition (9) does *not* hold in our experiments: we use GAE-style baselines, which provide only approximate centering. Consequently, the $O(\alpha\,\varepsilon_{A,\text{cand}})$ scaling represents an idealized limit; in the presence of the centering defect $\varepsilon_{\text{cent}}$ inherent to GAE, the bound degrades to $O(\alpha\,\varepsilon_{A,\text{cand}} + \varepsilon_{\text{cent}})$. The applicable deterministic bound is the $\varepsilon_{\text{cent}} > 0$ variant stated in Remark 5.8. In our experiments, $\varepsilon_{\text{cent}}$ is not directly measured; it remains an unmeasured but relevant quantity that affects the gap between the idealized bound and actual performance.

Remark 5.8 (Centering in practice). The $O(\alpha\,\varepsilon_{A,\mathrm{cand}})$ bound assumes exact statewise centering $\mathbb{E}_{a\sim\pi}[\widehat{A}(s,a)] = 0$ for all s. Using GAE or batch-level centering introduces a *centering defect* $\varepsilon_{\mathrm{cent}} := \sup_s |\mathbb{E}_{a\sim\pi}[\widehat{A}(s,a)]|$, which degrades the guarantee to $O(\alpha\,\varepsilon_{A,\mathrm{cand}} + \varepsilon_{\mathrm{cent}})$. Our experiments use approximate centering, so the tighter bound serves as structural guidance rather than a strict guarantee.

The backbone is CPI (Kakade & Langford, 2002); our observation is that for mixture updates with centered estimators, evaluation error contributes $O(\alpha\varepsilon_{A,\mathrm{cand}})$ rather than $O(\varepsilon_A)$. Centering holds exactly when $\widehat{A}(s,a) = \widehat{Q}(s,a) - \mathbb{E}_{b\sim\pi}[\widehat{Q}(s,b)]$ with exact baseline computation (feasible for small discrete action spaces).

The role of α. The evaluation-error penalty scales as $\alpha\varepsilon_{A,\mathrm{cand}}/(1-\gamma)$, where $\varepsilon_{A,\mathrm{cand}}$ is the candidate-policy expected advantage estimation bias from (10). This $O(\alpha)$ reduction occurs because the estimator is statewise-centered under the current policy: $\mathbb{E}_{a\sim\pi(\cdot|s)}[\widehat{A}(s,a)] = 0$, so for the mixture $\pi_{\mathrm{new}} = (1-\alpha)\pi + \alpha\pi_{\mathrm{cand}}$, only the $\alpha\pi_{\mathrm{cand}}$ component contributes nonzero expected estimated advantage. A sufficient condition is exact baseline computation (e.g., summation over a discrete action space).

Approximate centering (centering defect). When the baseline is approximated (e.g., via GAE), exact centering may fail. Define the *centering defect* $\varepsilon_{\mathrm{cent}} := \sup_{s\in\mathcal{R}_\pi} |\mathbb{E}_{a\sim\pi(\cdot|s)}[\widehat{A}(s,a)]|$. The bound generalizes to:

$$\eta(\pi_{\mathrm{new}}) \geq \widehat{L}_\pi(\pi_{\mathrm{new}}) - \frac{(1-\alpha)\varepsilon_{\mathrm{cent}} + \alpha\varepsilon_{A,\mathrm{cand}}}{1-\gamma} - \frac{2\varepsilon_{\mathrm{CPI}}\gamma\,\alpha^2}{(1-\gamma)^2}.$$

Theorem 5.7 is the special case $\varepsilon_{\mathrm{cent}} = 0$. This shows that centered advantage estimation is important but not critical: the bound degrades gracefully with the centering defect.

Proof sketch: (a) Start from CPI: $\eta(\pi_{\mathrm{new}}) \geq L_\pi(\pi_{\mathrm{new}}) - \frac{2\varepsilon_{\mathrm{CPI}}\gamma}{(1-\gamma)^2}\alpha^2$. (b) Write $L_\pi(\pi_{\mathrm{new}}) = \widehat{L}_\pi(\pi_{\mathrm{new}}) - \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\pi}\mathbb{E}_{a\sim\pi_{\mathrm{new}}}[\widehat{A}(s,a) - A^\pi(s,a)]$. (c) Expand $\pi_{\mathrm{new}} = (1-\alpha)\pi + \alpha\pi_{\mathrm{cand}}$. (d) Centering gives $\mathbb{E}_{a\sim\pi}[\widehat{A}(s,a)] = 0$ and $\mathbb{E}_{a\sim\pi}[A^\pi(s,a)] = 0$, so the $(1-\alpha)$ term vanishes. (e) The remaining term is bounded by $\alpha\varepsilon_{A,\mathrm{cand}}/(1-\gamma)$.

Summary: TRM evaluator dials and standard knobs. The value decomposition (Proposition 5.1) and Theorem 5.7 together show how the TRM-specific evaluator dials (L_z, n) and standard algorithmic knobs (K, α) shape safe improvement. Smaller L_z or larger n reduce unrolling bias (L_z^n); larger K tightens the residual coefficient; smaller α yields safer updates.

Persistent-z. The original TRM carries the latent across edits; bounds extend under a slow-drift assumption, with $L_z^n C_z/(1-L_z)$ replaced by a drift term $C_{\mathrm{drift}}(n)$ (see Section F).

6 Experiments

We show that contraction and projection modulate stability under depth mismatch on 4×4 and 9×9 Sudoku (setup details in Section B; 3 seeds throughout).

Stability under depth mismatch (Figure 2). We evaluate checkpoints trained at depth n_{train}=2 at mismatched depths $n_{\mathrm{eval}} \in \{4, 8, 16\}$. At 8× mismatch, contraction reduces Δ_V (0.156 → 0.038) and Δ_π (0.0063 → 0.0002). Both conditions use projection R=10 (which alone provides 6–10× Δ_V improvement); contraction adds marginal stability. With projection off, contraction still reduces drift on initial states.

6.1 Feasibility Results

On easy 4×4 (1–4 empties, 5k steps, 3 seeds), UPI–TRM achieves $93.3 \pm 2.3\%$ success (no contraction) and $90.7 \pm 5.0\%$ (contraction), vs. 52% random. On harder 4×4 (6–8 empties, 20k steps, 3 seeds), UPI–TRM reaches 48–57% while tuned PPO/A2C/DQN (11 configs) achieve **0%**. The modest contraction drop (90.7 vs. 93.3%) reflects the stability/expressivity trade-off.

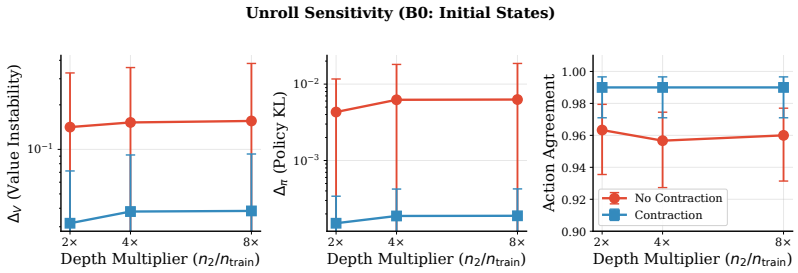

Figure 2: Unroll sensitivity (n_{train}=2, $n_{\text{eval}} \in \{4, 8, 16\}$). Contraction ($L_z$=0.9) vs. no contraction, R=10. Contraction reduces drift under depth mismatch.

Table 2: 9×9 Sudoku (50k steps, 3 seeds, mean ± std). All methods use the same TRM backbone and constraint-aware action masking. Score: mean cells correct out of 81; initial puzzle score ≈26.8. UPI–TRM is the only method that solves any puzzles.

Algorithm	Success	Score	Improvement
UPI–TRM	**4.0 ± 4.0%**	**53.3 ± 1.1**	**+26.5**
A2C	0.0 ± 0.0%	31.9 ± 0.4	+5.1
DQN	0.0 ± 0.0%	29.5 ± 0.3	+2.7
PPO	0.0 ± 0.0%	28.2 ± 1.5	+2.2

Why do baselines fail on 9×9? Standard RL fills a few obvious cells (+2 to +5) but cannot propagate constraints across the 81-step horizon via a single-pass value estimate. UPI–TRM's n-step latent unroll performs iterative constraint propagation *within each evaluation*; under the contraction condition ($L_z < 1$), the latent iterates converge to a fixed point at rate L_z^n (Proposition 2.3), bounding the truncation bias from finite unrolling. See Section D for a detailed mechanistic analysis.

6.2 Discussion and Limitations

Two of the three qualitative predictions of the theory are empirically supported (contraction → less drift, contraction modulus as stability dial); the third (α-related) is qualitatively consistent with observed training stability but has not been isolated in a controlled ablation, and several quantities in the bounds (L_V, $\varepsilon_{\text{cent}}$, $\varepsilon^*_{\text{res}}$) remain unmeasured. A detailed mapping is in Section C. **Limitations:** 3 seeds (descriptive); theory–implementation gaps remain (distillation, centering defect, uncontrolled L_V; see Section A); experiments restricted to Sudoku (Section E discusses broader task scope); projection dominates contraction when both active.

Related work. TRMs/HRMs (Jolicoeur-Martineau, 2025; Wang et al., 2025a) use iterative refinement; we add RL with checker feedback. Our plan-space MDP relates to learning-to-search (Daumé III et al., 2009; Chang et al., 2015); contraction analysis connects to DEQs (Bai et al., 2019); we build on CPI (Kakade & Langford, 2002) and RLVR (Wang et al., 2025b; DeepSeek-AI, 2025).

References

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, volume 32, pp. 690–701, 2019.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996. ISBN 1886529108.

Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daumé III, and John Langford. Learning to search better than your teacher. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2058–2066. PMLR, 2015.

Hal Daumé III, John Langford, and Daniel Marcu. Search-based structured prediction. *Machine Learning*, 75(3):297–325, 2009.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025.

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

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning*, pp. 267–274, 2002.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994. ISBN 0471619779.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.

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

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example, 2025b.

A Theory–Implementation Gap Details

Three quantities in our bounds are not directly measured in current experiments.

(a) Distillation gap. Theorem 5.7 applies to the exact mixture $\pi_{\text{new}}{=}(1{-}\alpha)\pi{+}\alpha\pi_{\text{cand}}$; our implementation distills into a single network π_ϕ. We monitor $\mathrm{KL}(\pi_{\text{new}}\|\pi_\phi)$ on replay states during training and observe values below 10^{-3} throughout, suggesting the distillation gap is small relative to the improvement signal. By Pinsker's inequality, this implies per-state total-variation distance below 0.03. *Caveat:* this measurement uses replay-buffer states, not a held-out set. Because our replay buffer is refreshed each iteration with on-policy rollouts and covers all reachable states in the small 4×4 domain, replay-state KL is a reasonable proxy for held-out KL; confirming this on a separate held-out partition is planned. A formal distillation-gap bound (e.g., via $\eta(\pi_{\text{new}}) - \eta(\pi_\phi) \le \frac{2\gamma}{(1-\gamma)^2}\sup_s \|\pi_{\text{new}}(\cdot|s) - \pi_\phi(\cdot|s)\|_{\text{TV}}$) remains future work.

(b) Centering defect. Our experiments use GAE-style baselines, providing approximate rather than exact statewise centering. The centering defect is defined as $\varepsilon_{\text{cent}} := \sup_s |\sum_a \pi(a|s)\,\widehat{A}(s,a)|$; for discrete action spaces this is directly computable by enumerating actions at each state in a held-out batch. We expect $\varepsilon_{\text{cent}}$ to be small in our setting for three reasons: the action space is discrete and moderate-sized (97 actions for 4×4, 729 for 9×9), the baseline $B_V(s) = \mathbb{E}_{b\sim\pi}[\widehat{Q}_V(s,b)]$ is computed by full summation over valid actions (not sampled), and the policy is near-converged at evaluation time. We leave empirical measurement of $\varepsilon_{\text{cent}}$ to future work. With nonzero $\varepsilon_{\text{cent}}$, the bound degrades to $O(\alpha\varepsilon_{A,\text{cand}} + \varepsilon_{\text{cent}})$ rather than $O(\alpha\varepsilon_{A,\text{cand}})$; the factors above suggest this degradation is modest, but quantifying it is a concrete open item.

(c) Value-head Lipschitz constant L_V. We disable value-head spectral normalization to avoid training instability, so L_V in Eq. (6) is uncontrolled during training. The value-head is a 2-layer MLP without explicit norm constraints; we do not measure its spectral norm post-hoc in the current experiments. This means the bound in Eq. (6) provides qualitative structural guidance (identifying which quantities matter) rather than a quantitative guarantee. Measuring and reporting L_V across training checkpoints is a concrete diagnostic we leave to future work.

B Experiment Setup Details

4×4 Sudoku: horizon $T{=}16$, $\gamma{=}0.99$, 97 actions with state-dependent masking. Contraction ($L_z{=}0.9$) enforced via operator-norm clamping on $z \to z$ layers; projection Π_R onto radius-R ball. Value-head spectral normalization disabled in all stability experiments. Episodic-z mode (latent

reinitialized each edit step); shaped reward as in Section 2. All networks are 2-layer MLPs (d=128); π_ϕ uses masked softmax. Adam optimizer, LR 3×10^{-4}; PPO clip ϵ=0.2, entropy coefficient 0.01. Step budgets refer to environment steps (each step = one edit action).

9×9 Sudoku: 81 cells, 729 actions (81 positions × 9 digits), constraint-aware masking reduces to ∼250 valid actions per state. Horizon T=81, γ=0.99, 50k training steps, 3 seeds. Score: filled cells $-2.0\times$ violations $-5.0\times$ zero-candidate cells; initial puzzle score ≈26.8, maximum 81.

C THEORY–PRACTICE BRIDGE

Our theoretical bounds make qualitative predictions that map to experimental observations:

1. *Stronger contraction → less drift under depth mismatch* (Proposition 5.1, L_z^n term): empirically supported—contraction reduces Δ_V by $4.1\times$ and Δ_π by $33\times$ at $8\times$ depth mismatch (Figure 2).
2. *Contraction modulus as stability dial*: supported when projection is disabled—L_z^{pre} correlates monotonically with depth-mismatch stability (Pearson $\rho = -0.87$ on initial states).
3. *Conservative α limits sensitivity to evaluation error* (Theorem 5.7): qualitatively consistent with stable training, though α has not been isolated in a controlled ablation.

Full quantitative validation of Eq. (6) requires measuring L_V, $\varepsilon^*_{\mathrm{res}}$, and $\varepsilon_{\mathrm{cent}}$; the centering defect is expected to be small (discrete actions, exact baseline summation; see Section A).

D 9×9 SUDOKU: MECHANISTIC ANALYSIS

Standard RL methods (PPO, A2C, DQN) learn to fill a few obvious cells (score improvements of $+2$ to $+5$) but cannot propagate constraint information across the 81-step edit horizon needed to complete the puzzle—they rely on a single forward pass for value estimation, which is insufficient for the combinatorial credit assignment required by 9×9 Sudoku. UPI–TRM's recursive evaluator provides a structural advantage: the n-step latent unroll performs iterative internal computation *within each evaluation*, enabling multi-step constraint propagation that a single forward pass cannot achieve. Under the contraction condition ($L_z < 1$), the latent iterates converge to a fixed point at rate L_z^n (Proposition 2.3), bounding the truncation bias introduced by finite unrolling; the Bellman-residual term in Proposition 5.1 captures how well the converged architecture approximates long-horizon value. Note that the current experiments use *episodic-z* (the latent is reinitialized at each edit step); a persistent-z extension that carries the latent across steps is an untested direction sketched in Section 2. Flat baselines lack this iterative internal computation altogether.

E BROADER TASK SCOPE

Our experiments are restricted to Sudoku, a controlled constraint-satisfaction domain. The theoretical results apply to any architecture meeting the contraction assumptions (Remark 4.2), but empirical generality remains unvalidated. Testing generality requires domains with qualitatively different structure:

- *Maze navigation:* sparse terminal reward, longer horizons ($T \gg 16$)—stresses the K-step bootstrap and discount sensitivity. The contraction dial would need to balance expressivity against stability over many more edit steps.
- *ARC-AGI subsets:* variable grid sizes, abstract pattern rules—challenges action-space scaling and out-of-distribution generalization of the learned evaluator. The inner recursion must generalize across structurally diverse inputs.
- *Code repair with unit tests:* large discrete action space, binary pass/fail reward—probes whether checker feedback scales beyond constraint satisfaction to domains with sparser and more delayed reward signals.

A meaningful positive result would show that the contraction/unroll-depth dials transfer: stronger contraction reduces depth-mismatch drift in the new domain. A meaningful negative result would identify regimes where contraction is too restrictive to permit learning, bounding the applicability of our analysis and motivating relaxations (e.g., local contraction or adaptive L_z schedules).

F Deferred Technical Results

Persistent-z (sketch). The original TRM carries the latent across edits. We extend bounds under a slow-drift assumption on the moving fixed point $z^*(x, y_t)$: value/advantage error has the same form as the episodic-z case, but with $L_z^n C_z/(1-L_z)$ replaced by a drift term $C_{\mathrm{drift}}(n)$ capturing tracking error of the moving fixed point. Persistent-z variants reuse computation but incur an additional "value-of-memory" / tracking penalty when history matters.