TROPICAL REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning for large language models typically maximizes expected return, adding up the probabilities of all successful trajectories. However, the classical sum formulation can only report *how often the model policy succeeds, not which solution actually worked*, and because probabilities sum to one, reinforcing one solution can make the model forget another that was never shown to be wrong. This makes expected return a poor fit for compositional reasoning, where a solution must be assembled from reasoning steps that the model produces in separate, often failed, attempts but rarely produces together. To address this, we propose Tropical Reinforcement Learning, which rests on a simple change of algebra: instead of adding the probabilities of alternative solutions, we take their maximum, which yields the *tropical semiring*. The value of a state then becomes the log-probability of its most likely verified solution, together with an explicit path that can be replayed and reused. This enables true composition, since the best prefix and the best suffix meeting at a shared state can be joined even when they come from different rollouts. To put this into practice, we introduce TROPIC, a training algorithm for deterministic, resettable environments with verifiable outcomes. On four agentic tasks (Sokoban, Countdown, FrozenLake, WebShop), TROPIC outperforms the strongest on-policy baselines by up to 16 percentage points. Changing the algebra of reinforcement learning, not just its estimators, can thus substantially improve compositional reasoning in language models.

1 INTRODUCTION

The best solution a language model can produce is often one it has never produced. A model may know how to formalize a problem, derive an intermediate result, invoke the right theorem, and check an answer, and may have done each of these in earlier attempts, yet never have chained them in the order a new problem requires (Press et al., 2023; Lake & Baroni, 2018). Reinforcement learning with verifiable rewards (RLVR) is the standard post-training tool for closing this gap by sampling many attempts and reinforcing the ones a verifier accepts (Kimi Team et al., 2025; DeepSeek-AI et al., 2026; GLM-5-Team et al., 2026; Yuan et al., 2026; Wen et al., 2026). However, the classical expected-return formulation learns only from what it samples, as a trajectory is credited only after the policy has generated it end-to-end. Recent works therefore focus on trajectory design, improving which rollouts are sampled and how they are credited (Wang et al., 2026; Feng et al., 2025; Li et al., 2026). Yet these methods keep the expected-return objective, and with it the requirement that a solution be sampled in one piece. For composition, this requirement is the bottleneck: a solution that chains d fragments, each found in a separate attempt, is sampled in one piece with probability exponentially small in d, even though the model has already produced every part of it.

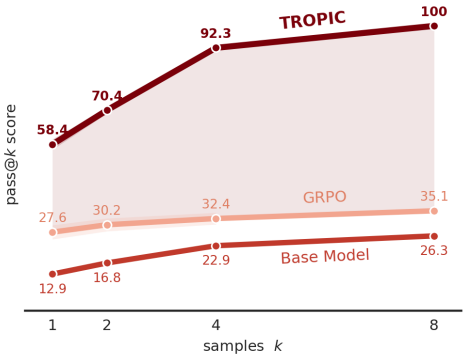

Figure 1: *Sokoban pass@k results*

Beyond failing to reward composition, expected return can also actively suppress the fragments it requires. Suppose the model produces outcomes A, B, their composition $A \circ B$, and a failure 0, with probabilities (p_A, p_B, p_{AB}, p_0). The first three receive reward one, so $J_{\mathrm{ER}} = p_A + p_B + p_{AB}$, but the composition has not yet been sampled. Moving from $(0.30, 0.30, 0, 0.40)$ to $(0.90, 0.05, 0, 0.05)$

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.

raises expected reward from 0.60 to 0.95. Yet the probability that a batch of twenty rollouts contains B at least once drops from $1 - 0.70^{20} \approx 0.999$ to $1 - 0.95^{20} \approx 0.642$. The objective has improved, but one of the two fragments that $A \circ B$ requires is now absent from about a third of all batches. In general, a high expected reward bounds the *total* probability of currently successful behavior, but places no lower bound on any *individual* fragment that a future composition requires. This is consistent with recent findings that loss of policy entropy and reasoning diversity limits what RL post-training can achieve (Cui et al., 2025; Jin et al., 2026).

Both problems stem from the same choice: expected return *adds up* the probabilities of alternative trajectories. A sum does not record which trajectory succeeded, so fragments from different rollouts cannot be combined, and, because probabilities sum to one, raising one term lowers another. Tropical RL changes this algebra. It still adds log-probabilities *along* each trajectory, but combines alternative trajectories by their *maximum*, which yields the max-plus, or *tropical*, semiring. The value of a state becomes the log-probability of its most likely verified path, and, because addition distributes over the maximum, it splits at any shared state into a best prefix and a best suffix, possibly from different rollouts. Compositions can thus be assembled without ever being sampled in one piece, and since the maximum ignores all but the best path, the value never loses a verified path once found. RL thus shifts from learning what the policy samples to learning what can be assembled from it.

We develop this shift from first principles to a practical algorithm. Our contributions are: (i) a DIAGNOSIS: we show that expected return, by adding probabilities over alternative trajectories, loses track of which solution succeeded and lets verified solutions be forgotten (Section 3); (ii) a FORCING RESULT: we prove that the maximum is the only aggregation whose value always names a verified path and never decreases as new paths are found (Theorem 2); and (iii) TROPIC: an algorithm of Tropical RL for agentic environments that joins fragments across rollouts and outperforms all baselines (Fig. 4, Fig. 1) on every task and model we test, by up to 16 points over the strongest one.

2 PRELIMINARIES

Notations. Let x denote a reasoning problem and let a state $s \in \mathcal{S}$ represent a computational interface, such as a token prefix. An action $a \in \mathcal{A}(s)$ is a token, reasoning step, tool call, or reusable computation that produces the next state $s' = T(s, a)$. Let $\bar{\pi}(a|s)$ denote a frozen language model used as a decoding or search prior. For a trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots, s_H)$, its probability under the frozen model is $P_{\bar{\pi}}(\tau \mid s_0) = \prod_{t=0}^{H-1} \bar{\pi}(a_t|s_t)$. Consequently, $\log P_{\bar{\pi}}(\tau \mid s_0) = \sum_{t=0}^{H-1} \log \bar{\pi}(a_t \mid s_t)$. Let $R_x(y) \in \{0, 1\}$ be a terminal verifier applied to a complete response y. We write $\mathcal{Y}_x^+(s) = \{\tau : \tau \text{ begins at } s \text{ and terminates at } y \text{ with } R_x(y) = 1\}$ for the set of verified successful continuations from s. A verified trajectory may receive a terminal quality score $R_x(\tau)$. $\mathcal{S}_\mathrm{T}$ is the set of terminal states, and $R_x(s)$ is the terminal score for task x.

Compositional reasoning and generalization. We call a solution *compositional* when it is a sequence of reusable computations, $\tau = f_1 \circ \cdots \circ f_d$, in which each fragment f_i, such as a reduction, a theorem application, or an arithmetic step, starts at the state produced by the previous one. *Compositional reasoning* is the ability to produce such a sequence for a given problem. *Compositional generalization* is the ability to solve problems whose solutions combine known fragments in a way that has not been observed before (Lake & Baroni, 2018; Keysers et al., 2020; Press et al., 2023). In this paper, "not observed" has a specific meaning: every fragment of the solution may already have appeared in some rollout, but never all of them together in a single one. The question we study is how a learner can turn fragments found in separate attempts into a verified solution that it was never able to sample.

Expected-Return RL. In a deterministic computation graph, the corresponding Bellman equations are $Q^{\bar{\pi}}(s, a) = r(s, a) + V^{\bar{\pi}}(T(s, a))$, $V^{\bar{\pi}}(s) = \sum_{a \in \mathcal{A}(s)} \bar{\pi}(a \mid s) Q^{\bar{\pi}}(s, a)$. With binary verification on a deterministic computation graph, the classical Bellman equation is $V^{\bar{\pi}}(s) = \sum_a \bar{\pi}(a \mid s)\, V^{\bar{\pi}}\big(T(s, a)\big)$ with $V^{\bar{\pi}}(y) = R_x(y)$ at terminals. It uses two operations: probabilities of successive steps are *multiplied*, and probabilities of alternative actions are *added*. Unrolling the recursion yields

$$V^{\bar{\pi}}(s) = \sum_{\tau \in \mathcal{Y}_x^+(s)} \prod_t \bar{\pi}(a_t \mid s_t) = \sum_{\tau \in \mathcal{Y}_x^+(s)} P_{\bar{\pi}}(\tau \mid s). \tag{1}$$

The classical value therefore has a direct operational meaning: *$V^{\bar{\pi}}(s)$ is the probability that one continuation sampled from $\bar{\pi}$, beginning at s, will eventually be accepted by the verifier.*

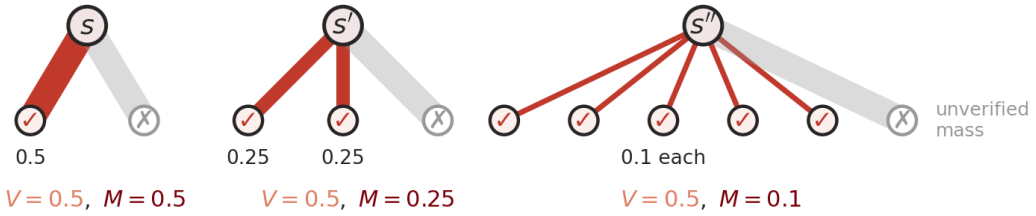

Figure 2: A difference between expected V and max M continuation.

3 The Algebra of Expected Return

Expected return is a path sum in the semiring $(\mathbb{R}_{\geq 0}, +, \times)$ (Eq. 1): steps along a path combine by $\otimes = \times$, and alternative paths combine by $\oplus = +$. The two operations play different roles. We keep the first and identify the second as the source of the problem.

Multiplying step probabilities is the autoregressive factorization of the model. In log space it becomes a sum of fragment scores: for an edge $f = (s, u_f, s')$ and a path τ,

$$w(f) = \sum_{j=1}^{|u_f|} \log \bar{\pi}(u_{f,j} \mid x, s, u_{f,<j}), \qquad W(\tau) = \sum_{f \in \tau} w(f) = \log P_{\bar{\pi}}(\tau \mid s). \tag{2}$$

A composed path is scored from its fragments without resampling, and because multiplication distributes over addition, the value admits dynamic programming.

Adding the probabilities of alternative continuations has three properties. Each is a strength when the goal is the success probability of a single sample, and a weakness when the goal is to retain and compose computations. *(i) Addition is not selective.* $a + b$ is the score of no single path, so the value reports *how much* of the actor succeeds, **not** *which* computation succeeds (Observation 1). *(ii) Addition acts on normalized weights.* Since $\sum_a \pi(a \mid s) = 1$, raising one alternative lowers the others (Proposition 1). *(iii) Addition is not idempotent.* $a + a = 2a$: a computation found twice counts twice, so what is sampled often is reinforced more (Appendix A.3).

The sum does not say which computation succeeds. Composition needs the most accessible verified continuation, the one to reuse:

$$M^{\bar{\pi}}(s) = \max_{\tau \in \mathcal{Y}_x^+(s)} P_{\bar{\pi}}(\tau \mid s). \tag{3}$$

Observation 1 (Non-sufficiency)**.** *No function g satisfies $M^{\bar{\pi}}(s) = g\big(V^{\bar{\pi}}(s)\big)$ for all states and continuation sets.*

Proof. Let $\mathcal{Y}_x^+(s) = \{\tau_1\}$ with $P_{\bar{\pi}}(\tau_1 \mid s) = 0.5$, and $\mathcal{Y}_x^+(s') = \{\tau_2, \tau_3\}$ with $P_{\bar{\pi}}(\tau_2 \mid s') = P_{\bar{\pi}}(\tau_3 \mid s') = 0.25$. Then $V^{\bar{\pi}}(s) = V^{\bar{\pi}}(s') = 0.5$, while $M^{\bar{\pi}}(s) = 0.5$ and $M^{\bar{\pi}}(s') = 0.25$. □

The gap can be arbitrarily large: n continuations of probability $1/n$ give $V^{\bar{\pi}} = 1$ but $M^{\bar{\pi}} = 1/n$. The converse also fails, since $\{0.5\}$ and $\{0.5, 0.25\}$ share $M^{\bar{\pi}} = 0.5$ but differ in $V^{\bar{\pi}}$. Once alternatives are added, the sum cannot tell which computations produced it (Fig. 2).

The sum can forget a verified solution. We call a value *policy-indexed* if it depends only on the current policy, $V_t(s) = \Phi(\pi_{\theta_t}, s)$. Every on-policy expected-return method is policy-indexed, with or without baselines, critics, or entropy bonuses.

Proposition 1 (Policy-indexed values forget)**.** *Let s have two actions a, b with independent softmax logits. The continuation after b always succeeds, $V^{\bar{\pi}}(T(s,b)) = 1$, and the continuation after a succeeds half the time, $V^{\bar{\pi}}(T(s,a)) = \frac{1}{2}$. Suppose a batch samples only a at s, some of these rollouts succeed, and the group baseline is below 1, so a receives estimated advantage $\widehat{A} > 0$. Then after one policy-gradient step, $\pi^+(b \mid s) < \pi(b \mid s)$ and $V_+^{\bar{\pi}}(s) < V^{\bar{\pi}}(s)$, even though no rollout observed b and the verifier's judgement of the continuation through b is unchanged.*

Proof. From $\partial \log \pi(a \mid s)/\partial z_j = \mathbf{1}\{j = a\} - \pi(j \mid s)$, the step gives $\Delta z_a = \eta \widehat{A}\,(1 - \pi(a \mid s)) > 0$ and $\Delta z_b = -\eta \widehat{A}\,\pi(b \mid s) < 0$, so $\pi^+(b \mid s) < \pi(b \mid s)$. Only the logits at s change, so $V^{\bar{\pi}}(s) = \frac{1}{2}\pi(a \mid s) + \pi(b \mid s) = \frac{1}{2} + \frac{1}{2}\pi(b \mid s)$ strictly decreases. □

A policy-indexed learner can restore $\pi(b \mid s)$ only by sampling b again. Normalization makes reinforcement zero-sum, so a verified capability can disappear without ever being refuted.

3.1 REPLACING THE AGGREGATION

The diagnosis points to a single change. We keep composition by adding log-scores along a path (Eq. 2), keep the model's log-probability as the score of each path, and replace only the operation that combines alternatives. Write $\oplus$ for that operation. The value of a state s is the aggregate of the scores of its verified continuations $V_{\oplus}(s) = \bigoplus_{\tau \in \mathcal{Y}_x^+(s)} W(\tau)$, where $\oplus$ is commutative and associative, so that the value does not depend on the order in which continuations were found. Expected return is one choice: with $\oplus = \mathrm{LSE}$, Eq. 1 gives $V_{\oplus}(s) = \log V^{\bar{\pi}}(s)$. For composition, we ask two things of the value.

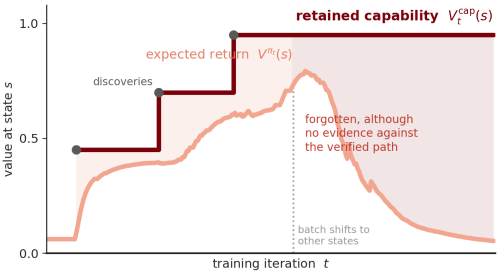

Figure 3: *Lemma 3 schematic illustration*

Certification. The value of s is always the score of one of its verified continuations: $a \oplus b \in \{a, b\}$. The value then names a concrete path that can be replayed, verified, and joined with others.

Retention. Finding a new verified continuation never lowers the value: $V_{\oplus}(s)$ cannot decrease when a path is added to $\mathcal{Y}_x^+(s)$.

Theorem 2 (Certification and retention force the maximum)**.** *If $\oplus$ satisfies* (Certification) *and* (Retention)*, then $a \oplus b = \max\{a, b\}$. With additive scores, the resulting structure is $(\mathbb{R} \cup \{-\infty\}, \max, +)$, the max–plus or tropical semiring.*

Proof. Adding a continuation of score b to one of score a gives $a \oplus b \geq a$ by (Retention); adding a to b gives $b \oplus a \geq b$, and $b \oplus a = a \oplus b$. Hence $a \oplus b \geq \max\{a, b\}$. By (Certification), $a \oplus b$ equals a or b, so it equals $\max\{a, b\}$. □

Each requirement rules out a familiar choice. Addition and LSE satisfy (Retention) but fail (Certification): their value exceeds every term and is the score of no path. The minimum satisfies (Certification) but fails (Retention). Every policy-indexed value fails (Retention) over the course of training, by Proposition 1: its value can fall although no verified continuation was removed.

By Theorem 2, the value of a state is the score of its best verified continuation,

$$V_{\mathrm{trop}}^{\bar{\pi}}(s) = \max_{\tau \in \mathcal{Y}_x^+(s)} W(\tau) = \log M^{\bar{\pi}}(s), \tag{4}$$

which is exactly the quantity that Observation 1 shows expected return cannot supply. It also has a direct meaning: it is the distance from the actor to its nearest verified computation, $V_{\mathrm{trop}}^{\bar{\pi}}(s) = -\min_{\tau \in \mathcal{Y}_x^+(s)} D_{\mathrm{KL}}\big(\delta_\tau \,\|\, P_{\bar{\pi}}(\cdot \mid s)\big)$, where δ_τ executes τ deterministically (Lemma 5, Appendix A.2). Section 4 builds reinforcement learning on this value.

The maximum reverses the three failures. It is *selective*: the value is the score of one path, recoverable by backpointers (Observation 1). It is *idempotent*: $\max\{a, a\} = a$, so rediscovering a computation changes nothing (Appendix A.3). And it is *monotone under set growth*, which the normalized sum cannot be (Proposition 1). Let $\mathcal{G}_t = (\mathcal{S}_t, \mathcal{E}_t)$ be the graph of transitions discovered by iteration t, $\mathcal{C}_t^+(s)$ the verified continuations constructible from it, and

$$V_{\mathrm{cap}}[t](s) = \max_{\tau \in \mathcal{C}_t^+(s)} R_x(\tau), \qquad \max \emptyset = -\infty. \tag{5}$$

Lemma 3 (Retained capability is monotone)**.** *If the verifier is fixed and the memory retains its discoveries, $\mathcal{E}_t \subseteq \mathcal{E}_{t+1}$, then $V_{\mathrm{cap}}[t+1](s) \geq V_{\mathrm{cap}}[t](s)$ for every state s and iteration t.*

Proof. $\mathcal{E}_t \subseteq \mathcal{E}_{t+1}$ implies $\mathcal{C}_t^+(s) \subseteq \mathcal{C}_{t+1}^+(s)$, and the maximum is monotone under inclusion. □

Under the maximum, discovery enlarges a set, and a maximum over a growing set cannot fall. Distributivity also survives the change, $c + \max\{a, b\} = \max\{c + a,\ c + b\}$, so a best prefix and a best suffix can be chosen independently at a shared state (Eq. 9), and dynamic programming over exponentially many compositions remains possible.

Table 1: Classical and tropical RL on the same verified paths. Proofs in Appendix A.

	Classical (sum)	Tropical (max)
The value measures	how much of the actor already succeeds	how far the actor is from its nearest verified solution (Lem. 5)
Advantage	relative to the average action, must be estimated	≤ 0, exactly 0 on the best path, computed by the DP
Performance difference	expectation over the new policy's trajectories	a sum over one path (Thm. 14)
Gradient weight on a step	visit prob. $\times$ action prob. $\times$ completion prob.	1 on every step of the best path (Cor. 3)
Rare path of depth d	escape takes $\sim \rho^{-d}$ steps	$O(\log \frac{1}{\rho})$ steps; MaxRL matches this on one path (Thm. 15)
Training on one path	can lower an unsampled solution (Prop. 1)	never lowers any state; shared parts gain in full (Thm. 4)
Rediscovery	counts again	changes nothing (Thm. 7)

4 Tropical Reinforcement Learning

Section 3.1 fixed the value: the score of the best verified continuation. This section turns it into reinforcement learning. We write its Bellman equations (Section 4.1), prove the property that separates it from classical RL, namely that improvement composes (Section 4.2), and build the algorithm TROPIC on it (Section 4.3).

4.1 Tropical Bellman equations

With $\lambda \geq 0$ weighing how much the model's likelihood counts, the tropical Bellman equations are

$$Q^{\bar{\pi}}_{\text{trop}}(s,a) = \lambda \log \bar{\pi}(a \mid s) + r(s,a) + V^{\bar{\pi}}_{\text{trop}}\big(T(s,a)\big), \qquad V^{\bar{\pi}}_{\text{trop}}(s) = \max_{a \in \mathcal{A}(s)} Q^{\bar{\pi}}_{\text{trop}}(s,a). \quad (6)$$

The classical recursion averages over actions; this one takes the best. In max–plus notation it is linear, $V^{\bar{\pi}}_{\text{trop}}(s) = \bigoplus_a \big(\lambda \log \bar{\pi}(a \mid s) \otimes r(s,a) \otimes V^{\bar{\pi}}_{\text{trop}}(T(s,a))\big)$, and unrolled it is the score of the best path,

$$V^{\bar{\pi}}_{\text{trop}}(s) = \max_{\tau} \Big\{ \lambda \log P_{\bar{\pi}}(\tau \mid s) + \sum_t r(s_t, a_t) + R_x(s_H) \Big\}. \quad (7)$$

With a binary verifier, internal rewards are zero, accepted terminals have value 0 and rejected ones $-\infty$, so failed paths drop out. At $\lambda = 1$,

$$V^{\bar{\pi}}_{\text{trop}}(s) = \max_{\tau \in \mathcal{Y}^+_x(s)} \log P_{\bar{\pi}}(\tau \mid s), \quad (8)$$

which is the tropical value of Eq. 4. Two special cases place the equation among known objects. At $\lambda = 0$ the model drops out, and Eq. 6 is classical Bellman optimality on a deterministic graph; its value is the capability of Lemma 3, which training can never lower. At $\lambda = 1$ it is Viterbi decoding under $\bar{\pi}$ restricted to verified paths, and it is the zero-temperature limit of the soft Bellman equation (Ziebart, 2010; Haarnoja et al., 2018). What is new is not the recursion but what it is used for: a value that composes, and a learning rule built on it.

Scores add along paths and addition distributes over max, so the best solution through a state s splits into a best prefix and a best suffix,

$$\max_{p \in \mathcal{P}_t(s),\, q \in \mathcal{C}^+_t(s)} W(p \circ q) = \max_{p \in \mathcal{P}_t(s)} W(p) + \max_{q \in \mathcal{C}^+_t(s)} W(q), \quad (9)$$

where $\mathcal{P}_t(s)$ are retained prefixes ending at s and $\mathcal{C}^+_t(s)$ retained verified suffixes starting there. The two halves may come from different rollouts. Eq. 9 requires that the score of a suffix depend only on the state where it starts, not on how that state was reached. Our approach therefore prompts the actor with the problem and the current environment state only, $\bar{\pi}(u \mid x, s)$, and includes the step count in s so that the graph is acyclic; any joined path is also replayed and rescored before it is used. A solution of d fragments, each produced with probability about ρ, takes about ρ^{-d} rollouts to sample in one piece, but is available to Eq. 9 as soon as each fragment has been found once. Table 1 lists how the change from sum to max alters every object built on the Bellman equation.

4.2 IMPROVEMENT COMPOSES

The central difference between the two algebras shows up when the actor is trained. Under expected return, reinforcing one solution takes probability from the others, and a verified solution can lose value without ever being refuted (Proposition 1). Under the maximum, the opposite holds.

Theorem 4 (Compositional policy improvement)**.** *Let $\mathcal{G}$ be the retained graph of a problem x, and for each state s let $\mathcal{C}(s)$ be the verified paths in $\mathcal{G}$ that start at s. Let the actor be a softmax policy with separate logits at each state, and let $V_\theta(s) = \max_{\tau \in \mathcal{C}(s)} \log P_\theta(\tau \mid s)$. Let $\tau^\star$ be the best verified path from the root, and take one gradient step on its log-likelihood, $\theta' = \theta + \eta \nabla_\theta \log P_\theta(\tau^\star \mid x)$, with any $\eta > 0$. Then:*

(i) $V_{\theta'}(s) \geq V_\theta(s)$ for every state s in $\mathcal{G}$;

(ii) if the best verified path from s passes through a state u of $\tau^\star$, then $V_{\theta'}(s) - V_\theta(s) \geq V_{\theta'}(u) - V_\theta(u) > 0$.

Best paths are taken with ties broken by a fixed rule. Proof in Appendix A.4.

The state s in (ii) need not lie on $\tau^\star$ and may never have been trained from. Its best solution can be a prefix from one rollout joined at u to a suffix trained as part of another, and it inherits the full gain made on that suffix. The reason is a property of the maximum: the step can lower the probability of actions that are not on the best path, but no best path uses them, so no value sees the loss. Under a sum every alternative counts, and the loss is always seen.

The theorem is exact for an actor with separate parameters per state. With shared parameters, as in a language model, a step on $\tau^\star$ also moves the actor at other states, and the guarantee holds up to this interference.

Tropical improves pass@k. Let $p(x) = \sum_{\tau \in \mathcal{Y}_x^+} P_\theta(\tau \mid x)$ be the pass@1 of problem x, so pass@k is $1 - (1 - p(x))^k$. The best verified path is one of the paths in this sum, so for every problem

$$\log p(x) \;\geq\; V^{\bar{\pi}}_{\mathrm{trop}}(s_0(x)), \qquad \text{pass@}k(x) \;\geq\; 1 - \left(1 - e^{V^{\bar{\pi}}_{\mathrm{trop}}(s_0(x))}\right)^k. \tag{10}$$

Raising the tropical value raises a lower bound on both quantities for every problem. The bound does not by itself force pass@1 up; what it explains is *which* problems improve. The expected-return gradient on problem x is $\nabla p(x) = p(x) \nabla \log p(x)$, so a problem solved once in a hundred attempts receives a hundredth of the push. The tropical gradient gives every problem with a verified path unit weight on that path (Corollary 3), however rare, and the archive supplies such paths, including composed ones, for problems the actor almost never solves alone. Pass@k is decided by exactly these problems: when $p(x)$ is small, pass@$k \approx k\, p(x)$. Expected return spends its gradient on problems that are already solved, which is why its pass@k curve flattens toward its pass@1 (Figure 1). Empirically, TROPIC also has the highest pass@1 on every task and model we test (Table 2).

Takeaway

Classical RL: improving one solution can erase another.
Tropical RL: improving one solution never lowers any other, and every solution that shares a piece of it gains at least as much as that piece.

4.3 TROPIC: TROPICAL POLICY OPTIMIZATION

Objective. TROPIC makes the best verified solution of each problem as likely as possible:

$$J_{\mathrm{trop}}(\theta) = \mathbb{E}_{x \sim \mu}\Big[\max_{\tau \in \mathcal{Y}_x^+(s_0(x))} \log P_\theta(\tau \mid x)\Big]. \tag{11}$$

Memory. For each problem, TROPIC keeps a graph whose edges are all valid steps $f = (s, u_f, s')$ seen so far. This includes steps from failed attempts, since a later attempt may succeed from where they ended. Complete paths that are replayed from the start and accepted by the verifier form the *archive*. At the start of each iteration the actor is frozen as $\bar{\pi}_t$, and every stored step is rescored under it, $w_t(f) = \log \bar{\pi}_t(u_f \mid x, s)$.

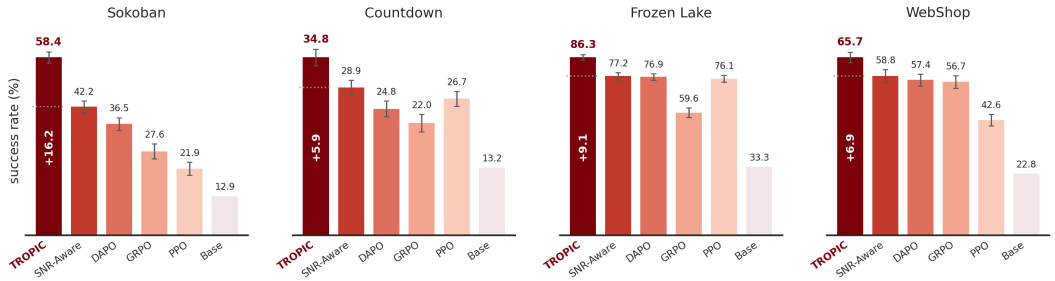

Figure 4: TROPIC performance against the baselines under the same compute budget.

Prefix and suffix values. For every state, TROPIC computes how easily the actor can *finish* from it (V_t) and how easily it can *reach* it (F_t):

$$V_t(s) = \max_{f=(s,u_f,s')} \{w_t(f) + V_t(s')\}, \qquad F_t(s') = \max_{f=(s,u_f,s')} \{F_t(s) + w_t(f)\}. \tag{12}$$

V_t is the tropical Bellman equation (Eq. 6) run backward from verified endings, and F_t is the same recursion run forward from the start. Their sum $F_t(s) + V_t(s)$ is the score of the best verified solution through s (Eq. 9). Both passes take time linear in the size of the graph.

Growing the memory. New paths enter the graph in two ways. *Sampling.* Each problem receives G groups of n attempts. The first group starts from the initial state, and the others start from *frontier* states, which the graph can reach but not yet finish from ($V_t(s) = -\infty$). To choose a start, TROPIC first picks the depth $r^\star$ from which the fewest attempts have started so far. At that depth it picks the state that is easy to reach and rarely tried:

$$s^\star = \arg \max_{s \text{ at depth } r^\star} \{F_t(s) - \kappa \log(1 + n_s)\}, \tag{13}$$

where n_s is the number of earlier attempts started from s. Each new attempt is prepended with the best known prefix to $s^\star$, then replayed and verified as a complete path. *Composition.* Wherever a state has both a known way in and a known way out, TROPIC joins the L best prefixes with the L best suffixes, then replays and verifies each joined path. Accepted joins enter the archive. This is how TROPIC finds solutions the actor never produced in one piece.

Training. For each problem, TROPIC selects up to K archived paths $\mathcal{B}_{x,t}$. The first is the best path $\tau_1 = \tau_x^\star$. The rest are the paths with the highest coverage score $c_t(\tau) = \sum_{f \in \tau} (1 + m_t(f))^{-1/2}$, where $m_t(f)$ is the number of earlier iterations in which step f was trained. These extra paths rehearse rarely trained steps so they are not forgotten (Proposition 1). The model then takes one gradient step on the per-token log-likelihood of the selected paths, averaged over the set $\mathcal{X}_t^+$ of problems that have at least one:

$$\mathcal{L}_t(\theta) = -\frac{1}{|\mathcal{X}_t^+|} \sum_{x \in \mathcal{X}_t^+} \frac{1}{|\mathcal{B}_{x,t}|} \sum_{\tau \in \mathcal{B}_{x,t}} \frac{1}{T_\tau} \log P_\theta(\tau \mid x), \tag{14}$$

where T_τ is the number of tokens in τ. It has no reward weights, no baseline, and no term that pushes any path down.

5 RELATED WORK

Recent work shows that RL post-training can teach language models to compose skills they already have (Yuan et al., 2026; Abdulsalam et al., 2026; Fu & Liu, 2026; Kong et al., 2026) and can improve reasoning beyond the base model (Wen et al., 2026), although its gains may stay within the base model's sampling support (Yue et al., 2025). Many methods improve RL training for LLMs: DAPO (Yu et al., 2025) and SNR-aware filtering (Wang et al., 2026) address training collapse, GiGPO (Feng et al., 2025) and VinePPO (Kazemnejad et al., 2024) refine step-level credit, process supervision rewards intermediate steps (Uesato et al., 2022; Lightman et al., 2024), and resets have been used for credit assignment (Samanta et al., 2026). All of these works optimize expected return, so a composition must be sampled in one piece before it is reinforced. They are complementary to ours, which changes how alternative trajectories are aggregated.

Other works move beyond expected return. Several optimize pass@k directly (Brown et al., 2024; Tang et al., 2025; Walder & Karkhanis, 2026), and MaxRL (Tajwar et al., 2026) maximizes the

Table 2: **Results (Qwen2.5-3B-Instruct & Gemma 4 E4B-it).** We report Success Rate (↑, mean ± standard deviation over three independent training seeds) and Generated Tokens per Episode (↓) on the held-out evaluation set, together with end-to-end Training Wall-Clock Time in hours (↓).

Task	Method	Qwen2.5-3B-Instruct			Gemma 4 E4B-it		
		Success Rate (↑)	Tokens (↓)	Time (h) (↓)	Success Rate (↑)	Tokens (↓)	Time (h) (↓)
Sokoban	Base Model	12.89	254	-	45.50	205	-
	PPO	21.90 ± 2.14	231	2.70	71.68 ± 1.71	209	8.85
	GRPO	27.60 ± 2.47	205	2.90	66.53 ± 2.32	254	8.20
	DAPO	36.55 ± 2.11	174	2.80	72.65 ± 1.35	219	8.40
	SNR-Aware	42.20 ± 1.96	201	2.10	76.55 ± 1.94	237	8.30
	TROPIC	**58.40 ± 1.83**	187	1.60	**83.10 ± 1.88**	224	8.10
Countdown	Base Model	13.23	150	-	42.18	204	-
	PPO	26.71 ± 1.46	164	2.50	66.53 ± 1.80	211	5.10
	GRPO	21.97 ± 1.72	152	2.00	67.65 ± 1.10	203	4.80
	DAPO	24.75 ± 1.51	158	2.10	63.10 ± 2.25	208	5.20
	SNR-Aware	28.93 ± 1.38	149	1.75	58.20 ± 1.95	201	4.50
	TROPIC	**34.79 ± 1.62**	147	1.20	**73.63 ± 1.48**	186	4.30
Frozen Lake	Base Model	33.34	192	-	71.30	218	-
	PPO	76.08 ± 1.65	110	2.55	84.40 ± 1.33	231	3.70
	GRPO	59.57 ± 2.28	101	2.20	83.90 ± 1.85	290	3.50
	DAPO	76.88 ± 1.57	106	2.30	85.10 ± 2.12	281	3.90
	SNR-Aware	77.24 ± 1.74	103	1.80	79.50 ± 2.45	220	3.20
	TROPIC	**86.35 ± 1.29**	100	1.35	**92.30 ± 1.10**	227	3.30
WebShop	Base Model	22.75	686	-	55.27	521	-
	PPO	42.57 ± 2.11	413	9.00	68.42 ± 1.94	496	18.20
	GRPO	56.68 ± 2.34	375	7.90	73.16 ± 1.81	472	16.60
	DAPO	57.39 ± 2.19	390	8.00	74.58 ± 1.69	481	17.10
	SNR-Aware	58.79 ± 2.51	382	5.90	76.84 ± 1.73	458	14.80
	TROPIC	**65.72 ± 1.96**	366	4.80	**81.40 ± 1.55**	439	14.50

logarithm of expected return, giving more weight to rarely solved problems. Expected return sums the probabilities of verified paths, MaxRL takes the logarithm of this sum, and Tropical RL takes its maximum, the zero-temperature limit in which alternatives are selected rather than summed.

Closer to our method, tree search retains and extends promising partial trajectories (Yao et al., 2023b; Zhang et al., 2024), T-STAR (Li et al., 2026) grafts corrected steps onto shared trajectory structure, and Liu et al. (2026) preserve correct prefixes when penalizing errors. These methods reuse trajectories that were sampled; TROPIC also composes verified solutions that were never sampled in one piece. We compare against T-STAR in Section 6.1. Finally, max-plus methods have been used to compute or approximate value functions of known deterministic control problems (Akian et al., 2008; McEneaney, 2006; Berthier & Bach, 2020; Gonçalves, 2021; Bach, 2019). We instead use the max-plus semiring to define the learning objective of a language-model policy.

6 Experiments and Results

Environments. We evaluate TROPIC on four interactive tasks from RAGEN-2 (Wang et al., 2026): (i) *Sokoban* (Weber et al., 2018), a grid-based planning task in which the agent pushes boxes onto targets while avoiding irreversible dead ends so that solutions depend on reusing good partial prefixes, (ii) *Countdown* (Katz et al., 2026), an exact-target arithmetic task with many valid solutions in which intermediate results from different attempts can be recombined, (iii) *FrozenLake* (Brockman et al., 2016), a grid world with terminal holes and sparse rewards in which safe partial paths must be retained and extended, and (iv) *WebShop* (Yao et al., 2023a), a long-horizon web-shopping task in which search, navigation, and attribute selection must be composed under delayed feedback. All environments are deterministic and resettable. Further details are provided in App. B.

Training and Baselines. For each task, we train Qwen-2.5-3B-Instruct (Qwen et al., 2025) and Gemma 4 E4B-it (Gemma Team, 2026) using PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), RAGEN's SNR-Aware method (Wang et al., 2026), and TROPIC. All methods are trained under a matched experimental setup, using the same GPU, task-relevant observations, rollout budget, environment workers, and the same observation representation. We discuss the differences in restarts, replay, and stopping in App. C. We evaluate all methods across three independent training seeds on a predefined held-out split of unseen instances. We report the task success rate, end-to-end training time, and the average number of generated tokens per episode.

Results. Table 2 and Fig. 4 show that TROPIC achieves the highest success rate on every task and for both models, while generating a comparable or smaller number of tokens and training faster in all but one setting. The largest gains appear on Sokoban, where irreversible dead ends make

complete solutions rare. TROPIC improves over the strongest baseline (SNR-Aware) by 16.2 points with Qwen and by 6.6 points with Gemma, and trains in 1.6 instead of 2.1 hours with Qwen. On Countdown, where exact verification leaves most attempts without signal, TROPIC gains 5.9 points over SNR-Aware with Qwen and 6.0 points over GRPO with Gemma. On FrozenLake, where a single misstep ends an episode, retaining safe partial paths yields 9.1 points over SNR-Aware with Qwen and 7.2 points over DAPO with Gemma. On WebShop, where late mistakes hide correct search and selection steps, TROPIC improves over SNR-Aware by 6.9 points with Qwen and 4.6 points with Gemma, while training fastest with both models. Among the non-tropical baselines, no method is consistently strongest. SNR-Aware is the best baseline on every task with Qwen but the weakest on Countdown and FrozenLake with Gemma, and the ordering of PPO, GRPO, and DAPO changes from task to task, suggesting that refinements within the expected-return objective do not transfer reliably across tasks and models. TROPIC, in contrast, is best in every setting and for both model families.

6.1 ABLATION RESULTS

To isolate the contribution of each core component in TROPIC, we evaluate three ablations: (i) *No Composition* retains persistent memory and frontier acquisition but disables cross-rollout graph recombination, restricting updates strictly to directly sampled paths, (ii) *Root-Only* preserves compositional reuse but forces all rollouts to initialize at s_0, removing frontier continuation, and (iii) *Successful-Only* maintains frontier exploration but restricts recombination to fragments originating from successful trajectories, ignoring useful transitions in failed rollouts. Additionally, we track *First-Solve*, the proportion of solved instances whose first verified solution was synthesized via cross-rollout composition before any complete successful trajectory was observed. By definition, *First-Solve* is zero for *No Composition* (which lacks recombination) and *Successful-Only* (which requires a pre-existing successful trajectory for fragment reuse).

Table 3: **Ablations and Cross-Comparison (Sokoban, Qwen2.5-3B-Instruct).** Success rate, tokens per episode, training time, and *First-Solve*.

Method	Success Rate (↑)	Tokens (↓)	Time (h) (↓)	First-Solve (↑)
Base Model	12.89	254	–	–
MaxRL	51.30±2.31	202	2.53	–
GiGPO	47.60±2.18	199	3.11	–
GiGPO + T-STAR	53.10±2.18	194	3.80	–
TROPIC (No Comp.)	40.80±2.26	205	1.45	0.0
TROPIC (Root-Only)	51.80±2.12	196	1.92	24.7±3.2
TROPIC (Succ.-Only)	53.30±1.97	192	1.51	0.0
Full TROPIC	**58.40±1.83**	187	1.58	**31.6±3.5**

Component Ablations. Table 3 reports ablations for Qwen on Sokoban. The results show that composition is the main source of improvement. Disabling it reduces success from 58.4% to 40.8%, the largest drop and roughly the level of the strongest baseline, SNR-Aware (42.2%), even though the archive and frontier restarts are kept. Removing frontier restarts (*Root-Only*) reduces success to 51.8% and *First-Solve* from 31.6% to 24.7%, since every attempt must regenerate known prefixes before exploring new suffixes. Restricting composition to successful trajectories reduces success to 53.3%, showing that fragments from failed rollouts remain useful building blocks.

Comparison with Alternative Methods. We further ask whether stronger success-oriented objectives or cross-trajectory credit assignment can match TROPIC. To this end, we compare against MaxRL (Tajwar et al., 2026), which replaces expected reward with a maximum-likelihood objective, GiGPO (Feng et al., 2025), which combines trajectory- and step-level advantages, and T-STAR (Li et al., 2026), which grafts corrected steps onto shared trajectory structure, instantiated on top of GiGPO. They reach 51.3%, 47.6%, and 53.1%, respectively, while TROPIC reaches 58.4%, 5.3 points above the strongest of them, with the fewest tokens and the shortest training time.

7 CONCLUSION AND DISCUSSION

RL has moved from tabular dynamic programming to deep function approximation and now to post-training large language models. Each wave changed how values are represented and estimated, but all kept the same objective. This objective measures how often a sampled attempt succeeds. Compositional reasoning instead asks which verified solution can be assembled from what the model has already produced. We showed that answering this question requires only changing the viewpoint of the algebra and herewith a single operation. Alternative trajectories can be combined by a maximum instead of a sum. The resulting tropical value names a concrete verified path, never loses a path once found, and splits at any shared state into a best prefix and a best suffix, so solutions can be assembled from fragments that were never sampled together. Our concrete implementation TROPIC builds on these properties and outperforms the strongest on-policy baselines on every task and model we test. We believe this change matters most for self-improving models. In particular, such a model must keep what it has verified and build new solutions from old ones. Transferring this growing set fully into the model's weights is the natural next step. Tropical RL asks what can be built from what the model has already found, and we believe this question will define the next wave.

REPRODUCIBILITY STATEMENT

In our paper we specifies the environments, and the exact definitions of the metrics and training procedure, C reports the compute used. All assumptions behind the theoretical claims are stated formally in A and each proposition from the main section is proved there. Source code is provided as anonymous supplementary material.

AI USE STATEMENT

In this work, we used generative AI tools to aid or polish writing, assist with code writing and with parts of the theoretical analysis. We have not used generative AI tools for retrieval and discovery (e.g., finding related work). All research ideas, including the proposed methods, as well as the core analysis, are original work of the authors. We have reviewed all AI-assisted work.

Text and code. We used OpenAI ChatGPT, Anthropic Claude, Codex, and Claude Code to edit author-written text and assist with code development.

Mathematical claims. We used Codex to transcribe images of our handwritten proofs into LaTeX. We used ChatGPT and Codex to assist with reviewing proofs throughout the paper. With Codex, we additionally used the Lean 4 skills package [1]. We subsequently reviewed and refined the analysis through several iterations with AI assistance.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Azwar Abdulsalam, Nishil Patel, and Andrew Saxe. RL Post-Training Builds Compositional Reasoning Strategies. arXiv preprint arXiv:2607.07646, 2026. URL https://arxiv.org/abs/2607.07646.

Marianne Akian, Stéphane Gaubert, and Asma Lakhoua. The Max-Plus Finite Element Method for Solving Deterministic Optimal Control Problems: Basic Properties and Convergence Analysis. *SIAM Journal on Control and Optimization*, 47(2):817–848, 2008. doi: 10.1137/060655286.

Francis Bach. Max-Plus Matching Pursuit for Deterministic Markov Decision Processes. *arXiv preprint arXiv:1906.08524*, 2019.

Eloise Berthier and Francis Bach. Max-Plus Linear Approximations for Deterministic Continuous-State Markov Decision Processes. *IEEE Control Systems Letters*, 4(3):767–772, 2020. doi: 10.1109/LCSYS.2020.2973199.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym, 2016. URL https://arxiv.org/abs/1606.01540.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large Language Monkeys: Scaling Inference Compute with Repeated Sampling. arXiv preprint arXiv:2407.21787, 2024. URL https://arxiv.org/abs/2407.21787.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The Entropy Mechanism of Reinforcement Learning for Reasoning Language Models, 2025. URL https://arxiv.org/abs/2505.22617.

DeepSeek-AI et al. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-Group Policy Optimization for LLM Agent Training, 2025. URL https://arxiv.org/abs/2505.10978.

Xiyan Fu and Wei Liu. Reinforcement Learning for Compositional Generalization with Outcome-Level Optimization. arXiv preprint arXiv:2605.04920, 2026. URL https://arxiv.org/abs/2605.04920.

[1] https://github.com/cameronfreer/lean4-skills

Gemma Team. Gemma 4 Technical Report, 2026. URL https://arxiv.org/abs/2607.02770.

GLM-5-Team et al. GLM-5: From Vibe Coding to Agentic Engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Vinicius Mariano Gonçalves. Max-Plus Approximation for Reinforcement Learning. *Automatica*, 129:109623, 2021. doi: 10.1016/j.automatica.2021.109623.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning (ICML)*, 2018.

Renren Jin, Pengzhi Gao, Yuqi Ren, Zhuowen Han, Tongxuan Zhang, Wuwei Huang, Wei Liu, Jian Luan, and Deyi Xiong. Revisiting Entropy in Reinforcement Learning for Large Reasoning Models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25300–25322, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1266. URL https://aclanthology.org/2026.findings-acl.1266/.

Michael Katz, Harsha Kokel, and Sarath Sreedharan. Seemingly Simple Planning Problems Are Computationally Challenging: The Countdown Game, 2026. URL https://arxiv.org/abs/2508.02900.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining Credit Assignment in RL Training of LLMs. *arXiv preprint arXiv:2410.01679*, 2024.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring Compositional Generalization: A Comprehensive Method on Realistic Data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SygcCnNKwr.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Weixin Xu, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, Zonghan Yang, and Zongyu Lin. Kimi k1.5: Scaling Reinforcement Learning with LLMs. arXiv preprint arXiv:2501.12599, 2025. URL https://arxiv.org/abs/2501.12599.

Lingjing Kong, Xin Liu, Guangyi Chen, Martin Q. Ma, Xiangchen Song, Yuekai Sun, Mikhail Yurochkin, Taylor W. Killian, Ruslan Salakhutdinov, Kun Zhang, Eric P. Xing, and Zhengzhong Liu. From Reasoning Traces to Reusable Modules: Understanding Compositional Generalization in Language Model Reasoning. arXiv preprint arXiv:2606.18089, 2026. URL https://arxiv.org/abs/2606.18089.

Brenden Lake and Marco Baroni. Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2873–2882. PMLR, 2018. URL https://proceedings.mlr.press/v80/lake18a.html.

Yu Li, Sizhe Tang, and Tian Lan. Reason in Chains, Learn in Trees: Self-Rectification and Grafting for Multi-Turn Agent Policy Optimization. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 4673–4686, 2026. URL https://arxiv.org/abs/2604.07165.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Haolin Liu, Dian Yu, Sidi Lu, Yujun Zhou, Rui Liu, Zhenwen Liang, Haitao Mi, Chen-Yu Wei, and Dong Yu. Save the Good Prefix: Precise Error Penalization via Process-Supervised RL to Enhance LLM Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 35450–35477, 2026.

William M. McEneaney. *Max-Plus Methods for Nonlinear Control and Estimation*. Systems and Control: Foundations and Applications. Birkhäuser, 2006.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and Narrowing the Compositionality Gap in Language Models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report, 2025. URL https://arxiv.org/abs/2412.15115.

Ankur Samanta, Akshayaa Magesh, Ayush Jain, Youliang Yu, Daniel Jiang, Kavosh Asadi, Kaveh Hassani, Paul Sajda, Jalaj Bhandari, and Yonathan Efroni. Credit Assignment with Resets in Language Model Reasoning. *arXiv preprint arXiv:2605.25507*, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation, 2018. URL https://arxiv.org/abs/1506.02438.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv preprint arXiv:2402.03300, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys)*, pp. 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Fahim Tajwar, Guanning Zeng, Yueer Zhou, Yuda Song, Daman Arora, Yiding Jiang, Jeff Schneider, Ruslan Salakhutdinov, Haiwen Feng, and Andrea Zanette. Maximum Likelihood Reinforcement Learning. *arXiv preprint arXiv:2602.02710*, 2026. URL https://arxiv.org/abs/2602.02710.

Yunhao Tang, Kunhao Zheng, Gabriel Synnaeve, and Rémi Munos. Optimizing Language Models for Inference Time Objectives Using Reinforcement Learning. *arXiv preprint arXiv:2503.19595*, 2025.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving Math Word Problems with Process- and Outcome-Based Feedback. arXiv preprint arXiv:2211.14275, 2022. URL https://arxiv.org/abs/2211.14275.

Christian Walder and Deep Tejas Karkhanis. Pass@k Policy Optimization: Solving Harder Reinforcement Learning Problems. *Advances in Neural Information Processing Systems*, 38:152416–152445, 2026.

Zenus Wang, Chi Gui, Xing Jin, Qineng Wang, Licheng Liu, Kangrui Wang, Shiqi Chen, Linjie Li, Zhengyuan Yang, Pingyue Zhang, Yiping Lu, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN-2: Reasoning Collapse in Agentic RL, 2026. URL https://arxiv.org/abs/2604.06268.

Théophane Weber, Sébastien Racanière, David P. Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adria Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, Razvan Pascanu, Peter Battaglia, Demis Hassabis, David Silver, and Daan Wierstra. Imagination-Augmented Agents for Deep Reinforcement Learning, 2018. URL https://arxiv.org/abs/1707.06203.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement Learning with Verifiable Rewards Implicitly Incentivizes Correct Reasoning in Base LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/517f9b9c227b9dd51dba4560f37165ed-Abstract-Conference.html.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents, 2023a. URL https://arxiv.org/abs/2207.01206.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. doi: 10.52202/075280-0517. URL https://papers.neurips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An Open-Source LLM Reinforcement Learning System at Scale, 2025. URL https://arxiv.org/abs/2503.14476.

Lifan Yuan, Weize Chen, Yuchen Zhang, Ganqu Cui, Hanbin Wang, Ziming You, Ning Ding, Zhiyuan Liu, Maosong Sun, and Hao Peng. From $f(x)$ and $g(x)$ to $f(g(x))$: LLMs Learn New Skills in RL by Composing Old Ones. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/eeae43a68515325cad64c0f54b2d0c70-Abstract-Conference.html.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1933. URL https://papers.nips.cc/paper_files/paper/2025/hash/537d5aa768c2d534016a4d06f87bc8fb-Abstract-Conference.html.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. ReST-MCTS*: LLM Self-Training via Process Reward Guided Tree Search. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2066. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/76ec4dc30e9faaf0e4b6093eaa377218-Abstract-Conference.html.

Brian D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Carnegie Mellon University, 2010.

A Structural Results and Proofs

This appendix collects the structural results behind Sections 3–4.3 and proves them in full. The main text established three facts: expected return is not a sufficient statistic for the best witness (observation 1), a policy-indexed value can forget a verified computation, and the requirements of retention force the max–plus semiring (Proposition **??**). Here we go further in four directions. We show that the three objectives in our related-work table are three *distances* from the actor to the set of certified computations, and that the tropical one is the only distance to a *computation* rather than to a distribution over computations (Section A.2); this is the result we regard as central, because it turns the choice of algebra into the choice of what the learner is being pulled toward. We show that the tropical learning signal is the unique one that depends only on which computations have been found and not on how often (Section A.3). We prove a policy improvement theorem in which training on a single certified path improves the tropical value at every state of the retained graph, with the improvement propagating undamped to interfaces that were never trained from, and we prove that this improvement step is not an expected-return improvement step and cannot be selected from expected-return information (Section A.4). Finally we prove the tropical analogue of the performance difference lemma, derive from it the tropical policy gradient and its lack of visitation weights, and give an exponential-versus-logarithmic separation between expected-return and tropical gradient ascent on a chain (Section A.5).

One idea organizes all of it. The tropical value is a functional of a *set* of verified computations, aggregated by a maximum; expected return is a functional of a *measure* over them, aggregated by a sum. A set grows under discovery and has a most accessible element; a measure is normalized and has only a total. Every theorem below is a consequence of one of three properties of the maximum that the sum lacks: it is idempotent, it is attained by one of its arguments, and it is insensitive to what happens to its non-maximal arguments.

A.1 Setting and conventions

Throughout, the environment is a finite acyclic computation graph. A nonterminal state s has a finite action set $\mathcal{A}(s)$ and a deterministic transition $T(s,a)$; terminal states $y \in \mathcal{S}_T$ carry a terminal score, which in the binary case is the boundary $R_x(y) \in \{0, -\infty\}$ of Section 4. A *path* from s is a sequence $\tau = \big((s_0,a_0),\dots,(s_{H-1},a_{H-1}),y\big)$ with $s_0 = s$, $s_{t+1} = T(s_t,a_t)$ and $y = s_H \in \mathcal{S}_T$. Because the graph is acyclic a path visits each state at most once; we write S_τ for its set of nonterminal states and $a_\tau(s)$ for the action it takes at $s \in S_\tau$. $\mathrm{Desc}(v)$ denotes the set of states reachable from v, including v, and $\mathrm{Anc}(v)$ the set of states from which v is reachable, excluding v; acyclicity means $\mathrm{Desc}(v) \cap \mathrm{Anc}(v) = \emptyset$.

An actor is a policy π with $\pi(a \mid s) > 0$ for all admissible a. It induces the path probability $P_\pi(\tau \mid s) = \prod_t \pi(a_t \mid s_t)$, a probability distribution over the set $\Omega(s)$ of complete paths from s. When the actor is *tabular softmax* we write $\pi_z(a \mid s) = \exp(z_{s,a}) / \sum_{a' \in \mathcal{A}(s)} \exp(z_{s,a'})$ with logits $z_{s,\cdot}$ that are independent parameters at each state. We use $\sigma(z) = 1/(1+e^{-z})$ for the two-action case.

A *certified set* $\mathcal{C}(s) \subseteq \Omega(s)$ is a set of paths from s that end at terminals with $R_x = 0$; it may be the full set $\mathcal{Y}_x^+(s)$ or the retained set $\mathcal{C}_t^+(s)$ of Section 4. The certified set inherits a graph: its states and edges are those visited by some certified path, and every retained-graph quantity below is computed on it. An action of the environment that is admissible at s but leads to no certified terminal is treated as an edge to a rejected terminal, with value $-\infty$; this lets the softmax range over all of $\mathcal{A}(s)$ while the dynamic program ranges over certified structure, and it is exactly how TROPIC's retained graph is used.

Path scores are additive with the accessibility weight $\lambda > 0$ and the internal reward r: $w(s,a) = \lambda \log \pi(a \mid s) + r(s,a)$ and $W_\pi(\tau) = \sum_t w(s_t,a_t) + g(y)$ with $g(y)$ the terminal score. In Sections A.2–A.4 we take $\lambda = 1$, $r \equiv 0$ and the binary boundary, so that $W_\pi(\tau) = \log P_\pi(\tau \mid s)$ for certified τ; Section A.5 keeps λ and r general. The tropical values $Q^{\bar\pi}_{\mathrm{trop}}, V^{\bar\pi}_{\mathrm{trop}}, A^{\bar\pi}_{\mathrm{trop}}$ are those of Equation equation 6; when the actor is clear from context we drop the superscript, and we write V', w' for quantities under an updated actor. The convention $\max \emptyset = -\infty$ is in force, and any sum containing $-\infty$ equals $-\infty$.

A.2 THREE PROJECTIONS ONTO THE CERTIFIED SET

The main text compares expected return, maximum likelihood and the tropical value by the operation each uses to aggregate verified paths: sum, log-sum-exp, max. That comparison is algebraic. There is an equivalent geometric comparison that is more revealing, because it says what each objective is pulling the actor toward. Fix a state s, the actor's path distribution $P = P_{\bar{\pi}}(\cdot \mid s)$ on $\Omega(s)$, and a certified set $\mathcal{C} = \mathcal{C}(s)$. The set of all distributions supported on $\mathcal{C}$ is a simplex $\Delta(\mathcal{C})$; its vertices are the point masses δ_τ, one per certified path, and each vertex is a *deterministic policy that executes one certified computation*. The three objectives are the actor's distance to this simplex measured three ways.

Lemma 5 (Variational identities)**.** *Let P be a probability distribution on a finite set Ω with $P(\omega) > 0$ for all ω, and let $\mathcal{C} \subseteq \Omega$ be nonempty. Write $q^\star = P(\cdot \mid \mathcal{C})$ for the conditional of P on $\mathcal{C}$ and $\tau^\star \in \arg\max_{\tau \in \mathcal{C}} P(\tau)$. Then*

(i) $\min_{q \in \Delta(\mathcal{C})} D_{\mathrm{KL}}(q \,\|\, P) = -\log P(\mathcal{C})$*, attained uniquely at $q = q^\star$;*

(ii) $\min_{\tau \in \mathcal{C}} D_{\mathrm{KL}}(\delta_\tau \,\|\, P) = -\max_{\tau \in \mathcal{C}} \log P(\tau)$*, attained exactly at the maximizers of P on $\mathcal{C}$;*

(iii) $\max_{\tau \in \mathcal{C}} \log P(\tau) = \log P(\mathcal{C}) + \log q^\star(\tau^\star)$.

Consequently, with $P = P_{\bar{\pi}}(\cdot \mid s)$ and $\mathcal{C} = \mathcal{C}(s)$,

$$V^{\bar{\pi}}(s) = P(\mathcal{C}), \qquad \log V^{\bar{\pi}}(s) = -\min_{q \in \Delta(\mathcal{C})} D_{\mathrm{KL}}(q \,\|\, P), \qquad V^{\bar{\pi}}_{\mathrm{trop}}(s) = -\min_{\tau \in \mathcal{C}} D_{\mathrm{KL}}(\delta_\tau \,\|\, P), \tag{15}$$

and the gap between the last two is $V^{\bar{\pi}}_{\mathrm{trop}}(s) - \log V^{\bar{\pi}}(s) = \log q^\star(\tau^\star \mid s) \le 0$, the log posterior share of the best witness among the actor's successes from s.

Proof. (i) For $q \in \Delta(\mathcal{C})$ every term of the divergence with $q(\tau) = 0$ vanishes, so $D_{\mathrm{KL}}(q\|P) = \sum_{\tau \in \mathcal{C}} q(\tau) \log \frac{q(\tau)}{P(\tau)}$. Since $P(\tau) = q^\star(\tau) P(\mathcal{C})$ for $\tau \in \mathcal{C}$,

$$D_{\mathrm{KL}}(q\|P) = \sum_{\tau \in \mathcal{C}} q(\tau) \log \frac{q(\tau)}{q^\star(\tau)} + \sum_{\tau \in \mathcal{C}} q(\tau) \log \frac{1}{P(\mathcal{C})} = D_{\mathrm{KL}}(q\|q^\star) - \log P(\mathcal{C}).$$

The first term is nonnegative and vanishes exactly when $q = q^\star$, which gives the minimum and its uniqueness. (ii) $D_{\mathrm{KL}}(\delta_\tau\|P) = 1 \cdot \log \frac{1}{P(\tau)} = -\log P(\tau)$, so minimizing over $\tau \in \mathcal{C}$ is maximizing $\log P(\tau)$. (iii) $q^\star(\tau^\star) = P(\tau^\star)/P(\mathcal{C})$; take logarithms. The identifications in equation 15 are the definitions of $V^{\bar{\pi}}$ and, by Equation equation 8, of $V^{\bar{\pi}}_{\mathrm{trop}}$ under the binary boundary with $\lambda = 1$; the gap is (iii). Since $q^\star$ is a probability, $\log q^\star(\tau^\star) \le 0$. □

The lemma says that expected return measures how much of the actor's mass already lies on certified computations; maximum likelihood measures how far the actor is from the *nearest distribution over* certified computations; the tropical value measures how far the actor is from the *nearest certified computation*. The gap term quantifies the difference between the last two: it is zero precisely when all of the actor's success mass at s is concentrated on one path, and it is the information Proposition **??** shows expected return cannot supply. The three are related by exponentiation and by adding the gap, so nothing is lost in the identities; what differs is the *minimizer*, which is the object learning moves toward. For the hull distance it is the posterior $q^\star$, a distribution; for the vertex distance it is a path.

We now show that if one asks the value to describe a path, and asks it never to decrease when a certified path is added, then the aggregation is forced to be the maximum. This is a second route to Proposition **??**, and the two hypotheses are exactly the two operational requirements of the system in Section 4.3: the value must refer to something that can be replayed, composed and verified, and the archive must never lose what it has certified.

Definition 1 (Aggregation; certification; retention)**.** *An* aggregation *is a commutative and associative binary operation $\oplus : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. For a finite nonempty multiset A of reals we write $\bigoplus A$ for the iterated operation, which is well defined by commutativity and associativity, and we set $\bigoplus \emptyset = -\infty$. Given path scores, the induced value is $V_\oplus(s) = \bigoplus\{W(\tau) : \tau \in \mathcal{C}(s)\}$. We consider three properties.*

(Cont) $\oplus$ *is continuous on* $\mathbb{R}^2$.

(Cert) $\oplus$ *is* selective*:* $a \oplus b \in \{a, b\}$ *for all* $a, b \in \mathbb{R}$*. By induction,* $\bigoplus A \in A$ *for every finite nonempty multiset: the value of a state is the score of one of its certified continuations.*

(Ret) *Adding a continuation never lowers the value:* $\bigoplus(A \uplus \{c\}) \geq \bigoplus A$ *for every finite multiset* A *and every* $c \in \mathbb{R}$*.*

Theorem 6 (Certification and retention force the tropical value)**.** *An aggregation satisfying* (Cont)*,* (Cert) *and* (Ret) *is the maximum:* $a \oplus b = \max\{a, b\}$ *for all* a, b*. Conversely the maximum satisfies all three. Consequently, with the additive scores of Equation equation 2, the induced value is the tropical value* $V_{\oplus} = V^{\bar{\pi}}_{\mathrm{trop}}$ *of Equation equation 6, and by Lemma 5 it is the negative KL distance from the actor to the nearest certified vertex.*

Proof. Let $U = \{(a, b) \in \mathbb{R}^2 : a > b\}$, an open half-plane and therefore a connected set. Define

$$F_1 = \{(a, b) \in U : \ a \oplus b = a\}, \qquad F_2 = \{(a, b) \in U : \ a \oplus b = b\}.$$

By (Cont) the maps $(a, b) \mapsto a \oplus b - a$ and $(a, b) \mapsto a \oplus b - b$ are continuous, so F_1 and F_2 are closed subsets of U. By (Cert), $F_1 \cup F_2 = U$. On U we have $a \neq b$, so $F_1 \cap F_2 = \emptyset$. A connected set cannot be written as the disjoint union of two nonempty closed subsets, hence $F_1 = U$ or $F_2 = U$.

Suppose $F_1 = U$. Then $a \oplus b = a = \max\{a, b\}$ whenever $a > b$. For $a < b$, commutativity gives $a \oplus b = b \oplus a = b = \max\{a, b\}$. On the diagonal (Cert) gives $a \oplus a = a$. So $\oplus = \max$.

Suppose instead $F_2 = U$. The same argument gives $\oplus = \min$. But then (Ret) fails: for $A = \{a\}$ and $c < a$, $\bigoplus(A \uplus \{c\}) = \min\{a, c\} = c < a = \bigoplus A$. Hence this case is excluded and $\oplus = \max$.

Conversely $\max$ is continuous, commutative, associative, selective, and $\max(A \uplus \{c\}) \geq \max A$. With additive scores, $V_{\max}(s) = \max_{\tau \in \mathcal{C}(s)} W(\tau)$, which is the path form of Equation equation 6 under the binary boundary, and the last claim is Lemma 5(ii). □

The proof is short, and it is worth being clear about where the content lies. (Cont) and commutativity are hygiene: values should vary continuously with accessibilities and should not depend on the order in which continuations are listed. The two substantive hypotheses each eliminate one candidate. Certification eliminates every soft aggregation, because a soft aggregate of two different scores lies strictly between or above them and is the score of nothing; retention eliminates the minimum, the only other selective candidate. What survives is the maximum, and with it the witness, idempotence, and distributivity. The next proposition records which familiar aggregations fail which hypothesis; the point is not that they are wrong, but that each is the right value for a different question.

Proposition 2 (Which aggregations fail which hypothesis)**.** *(i)* $\mathrm{LSE}(a, b) = \log(e^a + e^b)$*, the aggregation of maximum likelihood, satisfies* (Cont)*, commutativity, associativity and* (Ret)*, and fails* (Cert)*.*

(ii) Addition of probabilities $p \oplus q = p + q$*, the aggregation of expected return on the probability scale, satisfies* (Ret) *and fails* (Cert)*.*

(iii) The minimum satisfies (Cont)*,* (Cert)*, commutativity and associativity, and fails* (Ret)*.*

(iv) No policy-indexed value satisfies the dynamic form of (Ret)*: by Proposition 1, such a value can strictly decrease on a state all of whose certified continuations remain certified.*

Proof. (i) LSE is continuous, symmetric, and associative since $\mathrm{LSE}(\mathrm{LSE}(a, b), c) = \log(e^a + e^b + e^c)$. For $\bigoplus(A \uplus \{c\}) = \log\left(\sum_{a \in A} e^a + e^c\right) > \log \sum_{a \in A} e^a$, so (Ret) holds. For any a, b, $\mathrm{LSE}(a, b) > \max\{a, b\}$ because $e^a + e^b > e^{\max\{a,b\}}$; hence $\mathrm{LSE}(a, b) \notin \{a, b\}$ and (Cert) fails. (ii) $p + q \geq p$, so (Ret) holds; $p + q \notin \{p, q\}$ when $p, q > 0$, so (Cert) fails. (iii) $\min\{a, b\} \in \{a, b\}$, continuity and the algebraic identities are immediate; the failure of (Ret) was shown in the proof of Theorem 6. (iv) is Proposition 1. □

The consequences of the maximum that the rest of this appendix uses are collected next. The first is the distributivity that makes composition a local operation; the second is the fact that the gradient of a maximum is the gradient of its maximizer; the third is that the value can be certified by one verifier

call. Each is elementary; what matters is that each is available *because* of Theorem 6 and unavailable under any soft aggregation.

Corollary 1 (Consequences of the maximum)**.** *Let* $V(s) = \max_{\tau \in \mathcal{C}(s)} W(\tau)$ *with additive scores.*

(a) (Optimal substructure.) *If* $\tau = p \circ q$ *attains* $V(s)$ *and* q *starts at* v*, then* q *attains* $V(v)$*. Consequently, for the set* $\mathcal{P}(v)$ *of certified prefixes from* s *to* v *and the set* $\mathcal{C}(v)$ *of certified suffixes from* v*,* $\max_{p \in \mathcal{P}(v),\, q \in \mathcal{C}(v)} W(p \circ q) = \max_p W(p) + \max_q W(q)$*, which is Equation equation 9.*

(b) (Gradient of the maximum.) *Let the scores depend on a parameter* $\theta \in \mathbb{R}^n$ *through continuously differentiable functions* $\theta \mapsto W_\theta(\tau)$*, and suppose that at* θ_0 *the maximizer* $\tau^\star$ *over* $\mathcal{C}(s)$ *is unique. Then* $\theta \mapsto V_\theta(s)$ *is continuously differentiable on a neighbourhood of* θ_0 *and* $\nabla_\theta V_{\theta_0}(s) = \nabla_\theta W_{\theta_0}(\tau^\star)$*. For* $W_\theta(\tau) = \log P_\theta(\tau \mid s)$ *this is the score function of the witness,* $\sum_{(s_t, a_t) \in \tau^\star} \nabla_\theta \log \pi_\theta(a_t \mid s_t)$*.*

(c) (One-call certificate.) *If* $V(s) > -\infty$ *there is a path* $\tau^\star$ *with* $W(\tau^\star) = V(s)$*; replaying* $\tau^\star$ *and querying the verifier once confirms that the terminal is accepted, and the score of the replayed path equals the value. No finite number of verifier calls on individual paths determines* $\log V^{\bar{\pi}}(s)$ *without knowledge of the probabilities of all paths in* $\mathcal{C}(s)$*.*

Proof. (a) Suppose q' is a certified suffix from v with $W(q') > W(q)$. Then $p \circ q'$ is a certified path from s and, by additivity, $W(p \circ q') = W(p) + W(q') > W(p) + W(q) = W(\tau) = V(s)$, contradicting that τ attains the maximum. Hence $W(q) = \max_{q'} W(q') = V(v)$. For the second claim, additivity gives $W(p \circ q) = W(p) + W(q)$ for every compatible pair, and the maximum of a sum of independent terms is the sum of the maxima: $\max_{p,q}[W(p) + W(q)] \le \max_p W(p) + \max_q W(q)$ termwise, and the pair of maximizers attains the right-hand side.

(b) Write $f_\tau(\theta) = W_\theta(\tau)$ for $\tau \in \mathcal{C}(s)$, finitely many continuously differentiable functions. By uniqueness, $\varepsilon := f_{\tau^\star}(\theta_0) - \max_{\tau \neq \tau^\star} f_\tau(\theta_0) > 0$. By continuity of finitely many functions there is a neighbourhood N of θ_0 on which $|f_\tau(\theta) - f_\tau(\theta_0)| < \varepsilon/2$ for all τ; on N, $f_{\tau^\star}(\theta) > f_\tau(\theta)$ for every $\tau \neq \tau^\star$, hence $V_\theta(s) = f_{\tau^\star}(\theta)$ on N and the differentiability claim follows. For log-likelihood scores, $\nabla_\theta \log P_\theta(\tau^\star \mid s) = \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)$ by the autoregressive factorization equation 2.

(c) The maximum over a finite nonempty set is attained; take $\tau^\star$ a maximizer. Its terminal is accepted because $W(\tau^\star) > -\infty$ requires $R_x(y) = 0$. For the last claim, $\log V^{\bar{\pi}}(s) = \log \sum_{\tau \in \mathcal{C}(s)} P(\tau \mid s)$ depends on every summand, and two certified sets with the same verified paths queried and different unqueried probabilities give different values. □

A.3 Frequency-free learning signals

Proposition 1 located the cause of forgetting in normalization: probability granted to one computation is taken from another. A related cause of collapse is *frequency bias*: in on-policy learning the weight a computation receives in the update is its sampling probability, so computations that are found often are reinforced more, are then found more often, and so on. One would like a learning signal that depends on *which* computations have been discovered but not on *how many times*. This section shows that this requirement is exactly idempotence of the aggregation, hence, by Theorem 6, exactly the maximum.

Definition 2 (Frequency-free aggregation)**.** *Let* $\oplus$ *be an aggregation as in Definition 1. For a finite multiset* A *let* $\operatorname{supp}(A)$ *be the set of its distinct elements.* $\oplus$ *is* frequency-free *if* $\bigoplus A = \bigoplus \operatorname{supp}(A)$ *for every finite nonempty multiset* A*. It is* idempotent *if* $a \oplus a = a$ *for all* a*.*

Theorem 7 (Frequency-free if and only if idempotent)**.** *An aggregation is frequency-free if and only if it is idempotent. In particular the maximum is frequency-free, while* LSE *and addition are not.*

Proof. Suppose $\oplus$ is frequency-free. For the multiset $A = \{a, a\}$ we have $\operatorname{supp}(A) = \{a\}$, so $a \oplus a = \bigoplus A = \bigoplus \{a\} = a$: $\oplus$ is idempotent.

Suppose $\oplus$ is idempotent. For a finite nonempty multiset A let $n(A) = |A| - |\operatorname{supp}(A)| \ge 0$ count its repetitions. We show $\bigoplus A = \bigoplus \operatorname{supp}(A)$ by induction on $n(A)$. If $n(A) = 0$ then A is a set and there is nothing to prove. If $n(A) \ge 1$, some element a occurs at least twice, and we may write

$A = \{a, a\} \uplus A'$ for a multiset A'. If $A' = \emptyset$ then $\bigoplus A = a \oplus a = a = \bigoplus\{a\} = \bigoplus \mathrm{supp}(A)$. Otherwise, using commutativity and associativity to group the two copies,

$$\bigoplus A = (a \oplus a) \oplus \bigoplus A' = a \oplus \bigoplus A' = \bigoplus(\{a\} \uplus A').$$

The multiset $\{a\} \uplus A'$ has the same support as A and $n(\{a\} \uplus A') = n(A) - 1$, so by the induction hypothesis its aggregate is $\bigoplus \mathrm{supp}(A)$.

Finally $\max\{a, a\} = a$, whereas $\mathrm{LSE}(a, a) = a + \log 2 \neq a$ and $a + a = 2a \neq a$ for $a \neq 0$. □

The theorem is a statement about values. Its bearing on learning is through the gradient: if the value is a function of the support only, so is its gradient wherever it exists. The next proposition computes the gradients of the three objectives on a multiset of certified paths and makes the contrast concrete. It is the appendix form of Equation equation **??**.

Proposition 3 (The weights of the three objectives)**.** *Let A be a finite multiset of certified paths from $s_0(x)$ with scores $W_\theta(\tau) = \log P_\theta(\tau \mid x)$, continuously differentiable in θ. Define*

$$J^A_{\mathrm{ER}}(\theta) = \sum_{\tau \in A} e^{W_\theta(\tau)}, \qquad J^A_{\mathrm{ML}}(\theta) = \log \sum_{\tau \in A} e^{W_\theta(\tau)}, \qquad J^A_{\mathrm{trop}}(\theta) = \max_{\tau \in A} W_\theta(\tau),$$

where sums run over the multiset with multiplicity. Then each gradient has the form $\sum_{\tau \in A} \omega(\tau) \nabla_\theta \log P_\theta(\tau \mid x)$ with

$$\omega_{\mathrm{ER}}(\tau) = P_\theta(\tau \mid x), \qquad \omega_{\mathrm{ML}}(\tau) = \frac{P_\theta(\tau \mid x)}{\sum_{\tau' \in A} P_\theta(\tau' \mid x)}, \qquad \omega_{\mathrm{trop}}(\tau) = \mathbf{1}\{\tau = \tau^\star\},$$

the last wherever the maximizer $\tau^\star$ is unique. If a path $\tau_0 \in A$ is duplicated, ∇J^A_{ER} acquires an additional term $P_\theta(\tau_0 \mid x) \nabla \log P_\theta(\tau_0 \mid x)$, every weight of ∇J^A_{ML} changes, and $\nabla J^A_{\mathrm{trop}}$ is unchanged.

Proof. For the first, $\nabla e^{W_\theta(\tau)} = e^{W_\theta(\tau)} \nabla W_\theta(\tau) = P_\theta(\tau \mid x) \nabla \log P_\theta(\tau \mid x)$; sum over the multiset. For the second, $\nabla \log \sum_\tau e^{W_\theta(\tau)} = \big(\sum_\tau e^{W_\theta(\tau)} \nabla W_\theta(\tau)\big) / \sum_{\tau'} e^{W_\theta(\tau')}$, which is the stated posterior weighting. The third is Corollary 1(b). Under duplication of τ_0: the sum defining J^A_{ER} gains a copy of the term $e^{W_\theta(\tau_0)}$ and its gradient gains a copy of that term's gradient; the denominator of ω_{ML} increases by $P_\theta(\tau_0 \mid x)$ and the numerator of the weight on τ_0 doubles, so all weights change; the multiset maximum and its unique maximizer are unchanged, hence so is the gradient of J^A_{trop}. □

The proposition gives the three objectives a common form and isolates their difference in the weight placed on each path. The expected-return weight is the path's own probability, which is what makes on-policy learning frequency weighted by construction. The maximum-likelihood weight is a posterior, which removes the dependence on absolute probability but still counts each retained realization. The tropical weight is an indicator, and it is the only one of the three that Theorem 7 allows. In TROPIC this is why the acquisition process may be arbitrarily off-policy, why no importance weights appear, and why the coverage counter $m_t(f)$ in Section 4.3 counts rehearsals chosen by the algorithm rather than samples produced by the actor: the only frequency in the method is one the learner controls.

A.4 Compositional policy improvement

The classical policy improvement theorem says that acting greedily with respect to an exact Q^π produces a policy at least as good as π at every state. It requires $Q^\pi(s, a)$ for every action at every state visited, which for expected return means the total success mass of every subtree. This section proves a tropical analogue with a different shape. A single certified path is trained by maximum likelihood, exactly the $K = 1$ update of Equation equation 14, and the tropical value is shown to be non-decreasing at *every* state of the retained graph, including states that were never trained from and never visited by the trained path. Moreover the improvement is additive over shared fragments and reaches an interface undamped: a state whose best certified continuation merges into the trained path inherits the full gain accumulated on that path from the merge point on.

Three properties of the maximum carry the proof, and it is worth naming them before the formalities. *Substructure*: a best path that reaches a state on the trained path must continue along it, because suffixes of maximizers are maximizers. *Join*: the value dominates the score of any particular path, so lower-bounding the value only requires exhibiting one path whose score did not fall. *Blindness*: the update can lower the log-probabilities of non-greedy actions at trained states, but no best path uses any of them, so the value never sees the loss. The third is where the algebra bites; it is dominance (R2 of Section 3) in dynamic form, and it is false for any aggregation that counts non-maximal alternatives.

Throughout this section the actor is tabular softmax with $\lambda = 1$, the boundary is binary, internal rewards are zero, and every nonterminal state on the trained path has at least two admissible actions (a state with one action has nothing to learn). Values are computed on the certified graph as described in Section A.1.

Definition 3 (Greedy action and greedy path)**.** *Fix a total order on the actions at every state. Given logits z, hence $\pi = \pi_z$, scores $w(s,a) = \log \pi(a \mid s)$ and values $Q(s,a) = w(s,a) + V(T(s,a))$, $V(s) = \max_a Q(s,a)$, the* greedy action *at a nonterminal s with $V(s) > -\infty$ is the first maximizer $a^\star(s) = \min\{a \in \mathcal{A}(s) : Q(s,a) = V(s)\}$, and the* greedy path *$\tau^\star(s)$ is defined recursively by $\tau^\star(s) = (s, a^\star(s)) \circ \tau^\star(T(s,a^\star(s)))$, with $\tau^\star(y) = y$ at terminals. The map $s \mapsto a^\star(s)$ is a deterministic policy on the graph and $\tau^\star(s)$ is its rollout from s.*

Lemma 8 (The greedy path attains the value and has substructure)**.** *For every state s with $V(s) > -\infty$: the greedy path $\tau^\star(s)$ ends at a terminal y with $R_x(y) = 0$ and $W(\tau^\star(s)) = V(s)$. For every state $u \in S_{\tau^\star(s)}$ the action taken there is $a^\star(u)$ and the suffix of $\tau^\star(s)$ from u is $\tau^\star(u)$. In particular, if two greedy paths visit a common state they take the same action there and coincide from that state on.*

Proof. Induct in reverse topological order. At a terminal $V(y) = R_x(y)$, and $V(y) > -\infty$ forces $R_x(y) = 0 = W(y)$. At a nonterminal s with $V(s) > -\infty$, the maximum $V(s) = \max_a Q(s,a)$ is attained by $a^\star(s)$, so $V(T(s,a^\star(s))) > -\infty$ and by induction $W(\tau^\star(T(s,a^\star(s)))) = V(T(s,a^\star(s)))$ with an accepted terminal. Then $W(\tau^\star(s)) = w(s,a^\star(s)) + W(\tau^\star(T(s,a^\star(s)))) = Q(s,a^\star(s)) = V(s)$. The substructure statements are immediate from the recursive definition: at each state visited, the path takes $a^\star$ of that state and continues with the greedy path of the successor, so the suffix from any visited u is by construction $\tau^\star(u)$, which depends on u alone. □

Definition 4 (Tropical greedy update)**.** *Let $\tau = \tau^\star(s_0)$ with nonterminal state set S_τ, and let $\eta > 0$. The tropical greedy update is one gradient step on the log-likelihood of τ with respect to the logits,*

$$z' = z + \eta \nabla_z \sum_{(s,a)\in\tau} \log \pi_z(a \mid s).$$

Because a path visits each state once and logits at different states are independent, this reads $z'_{s,\cdot} = z_{s,\cdot}$ for $s \notin S_\tau$ and

$$z'_{s,a} = z_{s,a} + \eta\big(\mathbf{1}\{a = a^\star(s)\} - \pi(a \mid s)\big) \qquad \text{for } s \in S_\tau,\ a \in \mathcal{A}(s).$$

It is the $K = 1$ update of Equation equation 14 for one problem, without the token normalization. Primes denote quantities under z', and $\Delta w(s,a) = w'(s,a) - w(s,a)$.

Lemma 9 (Local effect of the update)**.** *(a) $\Delta w(s,a) = 0$ for every $s \notin S_\tau$ and every a.*

(b) For $s \in S_\tau$ and $a \neq a^\star(s)$, $\Delta w(s,a^\star(s)) - \Delta w(s,a) = \eta\big(1 - \pi(a^\star(s) \mid s) + \pi(a \mid s)\big) > 0$.

(c) For $s \in S_\tau$, $\Delta w(s,a^\star(s)) > 0$.

(d) Consequently $\Delta w(s,a) \le \Delta w(s,a^\star(s))$ for all $s \in S_\tau$ and all a, with $\Delta w(s,a^\star(s)) \ge 0$ for all s on or off τ.

Proof. (a) The logits at s are unchanged, hence so is $\pi(\cdot \mid s)$. (b) At any state, $w(s,a) = z_{s,a} - \log \sum_{a'} e^{z_{s,a'}}$, so $\Delta w(s,a) = \Delta z_{s,a} - \big[\log \sum_{a'} e^{z'_{s,a'}} - \log \sum_{a'} e^{z_{s,a'}}\big]$, and the bracket does not depend on a. Subtracting the expressions for $a^\star(s)$ and a, $\Delta w(s,a^\star(s)) - \Delta w(s,a) = \Delta z_{s,a^\star(s)} - \Delta z_{s,a} = \eta(1 - \pi(a^\star(s) \mid s)) + \eta\pi(a \mid s)$, which is positive since $\pi(a^\star(s) \mid s) < 1$ when s has at least two actions. (c) Write $\pi(a^\star \mid s) = \big(1 + \sum_{a \neq a^\star} e^{z_{s,a} - z_{s,a^\star}}\big)^{-1}$ and likewise for z'. By (b)

each exponent $z'_{s,a} - z'_{s,a^\star}$ is strictly smaller than $z_{s,a} - z_{s,a^\star}$, so the sum strictly decreases and $\pi'(a^\star \mid s) > \pi(a^\star \mid s)$, i.e. $\Delta w(s, a^\star(s)) > 0$. (d) is (b), (c) and (a) together. □

Part (b) is the exact statement of blindness: whatever normalization does to the probabilities of the other actions at a trained state, the greedy action gains strictly more than any of them, in log units, by an amount that does not depend on the value function at all. Part (c) says the greedy action's own log-probability rises. Nothing is claimed about the other actions individually; with three or more actions some of them may gain probability from the dominant one. The proofs below never need them to lose.

Lemma 10 (Gain bound). *For every path τ' in the certified graph,*

$$W'(\tau') - W(\tau') \;\le\; \sum_{x \in S_{\tau'} \cap S_\tau} \Delta w(x, a^\star(x)),$$

and for every state v,

$$V'(v) \;\le\; V(v) + \sum_{x \in S_\tau \cap \mathrm{Desc}(v)} \Delta w(x, a^\star(x)).$$

Proof. $W'(\tau') - W(\tau') = \sum_{(x,a) \in \tau'} \Delta w(x, a)$. For $x \notin S_\tau$ the term is 0 by Lemma 9(a); for $x \in S_\tau$ it is at most $\Delta w(x, a^\star(x))$ by Lemma 9(d). This gives the first inequality. For the second, if $V(v) = -\infty$ then no certified path leaves v and $V'(v) = -\infty$ as well, since the graph and the boundary are unchanged. Otherwise every path τ' from v has $S_{\tau'} \subseteq \mathrm{Desc}(v)$, and since each $\Delta w(x, a^\star(x)) \ge 0$,

$$V'(v) = \max_{\tau'} W'(\tau') \le \max_{\tau'} \Big[W(\tau') + \sum_{x \in S_\tau \cap \mathrm{Desc}(v)} \Delta w(x, a^\star(x)) \Big] = V(v) + \sum_{x \in S_\tau \cap \mathrm{Desc}(v)} \Delta w(x, a^\star(x)).$$

□

Lemma 11 (Stability along the trained path, with exact gain). *Write $\tau = \big((u_0, a_0), \ldots, (u_{m-1}, a_{m-1}), u_m\big)$ with $u_0 = s_0$, $a_k = a^\star(u_k)$ and u_m an accepted terminal, and set $\Delta_{\ge k} = \sum_{j=k}^{m-1} \Delta w(u_j, a_j)$ for $0 \le k \le m$, so $\Delta_{\ge m} = 0$. Then for every k:*

$$V'(u_k) = V(u_k) + \Delta_{\ge k},$$

and for $k < m$ the action a_k is the unique maximizer of $Q'(u_k, \cdot)$. In particular the greedy policy is unchanged on S_τ, the greedy path from s_0 under z' is again τ, and $\Delta_{\ge k} > 0$ for $k < m$.

Proof. Backward induction on k. For $k = m$, $V'(u_m) = R_x(u_m) = 0 = V(u_m)$ and $\Delta_{\ge m} = 0$. Assume the claim for $k+1$ and consider u_k.

For the greedy action, since a_k attains $V(u_k)$ we have $V(u_k) = w(u_k, a_k) + V(u_{k+1})$, and therefore

$$Q'(u_k, a_k) = w'(u_k, a_k) + V'(u_{k+1}) = w(u_k, a_k) + \Delta w(u_k, a_k) + V(u_{k+1}) + \Delta_{\ge k+1} = V(u_k) + \Delta_{\ge k}.$$

For any other action $a \ne a_k$, let $v = T(u_k, a)$. The states $u_0, \ldots, u_k$ are ancestors of v: u_k is a parent of v, and each u_j with $j < k$ is an ancestor of u_k. By acyclicity none of them lies in $\mathrm{Desc}(v)$, so $S_\tau \cap \mathrm{Desc}(v) \subseteq \{u_{k+1}, \ldots, u_{m-1}\}$ and Lemma 10 gives $V'(v) \le V(v) + \Delta_{\ge k+1}$. Hence, using $Q(u_k, a) = w(u_k, a) + V(v) \le V(u_k)$,

$$Q'(u_k, a) = w'(u_k, a) + V'(v) \le w(u_k, a) + \Delta w(u_k, a) + V(v) + \Delta_{\ge k+1} \le V(u_k) + \Delta w(u_k, a) + \Delta_{\ge k+1}.$$

Subtracting,

$$Q'(u_k, a_k) - Q'(u_k, a) \;\ge\; \Delta w(u_k, a_k) - \Delta w(u_k, a) = \eta\big(1 - \pi(a_k \mid u_k) + \pi(a \mid u_k)\big) > 0$$

by Lemma 9(b). (If $V(v) = -\infty$ the inequality is trivial.) So a_k is the unique maximizer of $Q'(u_k, \cdot)$ and $V'(u_k) = Q'(u_k, a_k) = V(u_k) + \Delta_{\ge k}$. Since the first maximizer is the unique maximizer, $a'^\star(u_k) = a_k$; applying this along τ shows the greedy path from s_0 is unchanged. Finally $\Delta_{\ge k} > 0$ for $k < m$ because each summand is positive by Lemma 9(c). □

Theorem 12 (Compositional policy improvement)**.** *Let $\tau = \tau^\star(s_0)$ and let z' be the tropical greedy update along τ with any step $\eta > 0$.*

(i) (Pointwise improvement from one path.) *For every state s of the certified graph, $V'(s) \geq V(s)$.*

(ii) (Additivity over shared fragments.) *For every s with $V(s) > -\infty$,*

$$V'(s) - V(s) \;\geq \sum_{x \in S_{\tau^\star(s)} \cap S_\tau} \Delta w(x, a^\star(x)).$$

(iii) (Undamped propagation to interfaces.) *If $\tau^\star(s)$ visits a state of τ and u_k is the first such state, then $\tau^\star(s)$ coincides with τ from u_k on and*

$$V'(s) - V(s) \;\geq\; V'(u_k) - V(u_k) = \Delta_{\geq k} > 0.$$

(iv) (Iteration.) *Repeating the update produces, at every state, a non-decreasing sequence of values, with the same trained path at every step.*

Proof. (i)–(ii). If $V(s) = -\infty$ then $V'(s) = -\infty$ and (i) holds with equality. Otherwise let $\tau^\star(s)$ be the greedy path from s under z; by Lemma 8 it is certified and $W(\tau^\star(s)) = V(s)$. By the join property $V'(s) \geq W'(\tau^\star(s))$, and $W'(\tau^\star(s)) - W(\tau^\star(s)) = \sum_{(x,a) \in \tau^\star(s)} \Delta w(x, a)$. For $x \notin S_\tau$ the term is 0 (Lemma 9(a)). For $x \in S_\tau$, Lemma 8 says the action taken by $\tau^\star(s)$ at x is $a^\star(x)$, so the term is $\Delta w(x, a^\star(x)) > 0$ (Lemma 9(c)). Hence $V'(s) \geq V(s) + \sum_{x \in S_{\tau^\star(s)} \cap S_\tau} \Delta w(x, a^\star(x)) \geq V(s)$, which is (ii) and (i).

(iii). The states of $\tau^\star(s)$ before u_k lie outside S_τ, by the choice of u_k. From u_k on, $\tau^\star(s)$ equals $\tau^\star(u_k)$ by Lemma 8, and so does the suffix of τ from u_k, since $\tau = \tau^\star(s_0)$ visits u_k. Therefore $S_{\tau^\star(s)} \cap S_\tau = \{u_k, \ldots, u_{m-1}\}$ and the sum in (ii) equals $\Delta_{\geq k}$, which is $V'(u_k) - V(u_k)$ by Lemma 11 and is positive.

(iv). By Lemma 11 the greedy path from s_0 under z' is again τ, so the next update is a tropical greedy update along the same path and (i) applies again. □

Read (iii) as a statement about composition. The state s was never trained from and the path $\tau^\star(s)$ was never sampled; it is a prefix from s to u_k, discovered in some rollout, composed at the interface u_k with a suffix that was trained as part of a different path. The theorem says that all of the training on that suffix is inherited by s, in full, in log units. Under expected return the change at s from training at u_k is scaled by the share of s's success mass that flows through u_k, and is further reduced by the mass lost on suppressed alternatives; under the maximum neither attenuation exists, because the value at s is the score of one path and that path does not use any suppressed alternative.

The next proposition establishes that the improvement step of Theorem 12 is genuinely a different step from classical policy improvement, in two senses: the action it reinforces cannot be determined from expected-return information, and acting on expected-return information can destroy the tropical value while the tropical step can destroy expected return. The example is the one from Proposition **??**, lifted from values to decisions.

Proposition 4 (Incompatibility with expected-return improvement)**.** *(a)* (Non-identifiability.) *There exist two instances that agree on a state u, on $\pi(\cdot \mid u)$, on $V^\pi(T(u, a))$ for every action, and on $V^\pi(u)$, yet have different tropical greedy actions at u. Hence no rule that selects an action from these quantities selects the tropical greedy action on all instances.*

(b) (Mutual harm.) *Let u have two actions a, a' with a the unique tropical greedy action and a' the unique expected-return greedy action, $V^\pi(T(u, a')) > V^\pi(T(u, a))$. Any update that increases $\pi(a \mid u)$ and changes no other state strictly decreases $V^\pi(u)$. Any update that increases $\pi(a' \mid u)$ and changes no other state strictly decreases $V^\pi_{\mathrm{trop}}(u)$ provided $\log \frac{\pi'(a'|u)}{\pi(a'|u)} < -A^\pi_{\mathrm{trop}}(u, a')$, in particular for all sufficiently small steps.*

Proof. (a) Take $\pi(a \mid u) = \pi(a' \mid u) = \frac{1}{2}$ in both instances. In the first, $T(u, a)$ has a single certified continuation of probability $\frac{1}{2}$ and $T(u, a')$ has three of probability $\frac{1}{4}$ each. In the second, $T(u, a)$ has

two of probability $\frac{1}{4}$ and $T(u, a')$ one of probability $\frac{3}{4}$. In both, $V^{\bar{\pi}}(T(u, a)) = \frac{1}{2}$, $V^{\bar{\pi}}(T(u, a')) = \frac{3}{4}$ and $V^{\bar{\pi}}(u) = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{3}{4} = \frac{5}{8}$. The tropical Q-values are $\log \pi(a \mid u) + \max \log P$ over the subtree: in the first instance $\log(\frac{1}{2} \cdot \frac{1}{2}) = \log \frac{1}{4}$ for a against $\log(\frac{1}{2} \cdot \frac{1}{4}) = \log \frac{1}{8}$ for a', so the tropical greedy action is a; in the second, $\log \frac{1}{8}$ for a against $\log(\frac{1}{2} \cdot \frac{3}{4}) = \log \frac{3}{8}$ for a', so it is a'. A rule that is a function of the listed quantities returns the same action in both instances and is therefore wrong in one of them.

(b) With two actions, $\pi(a' \mid u) = 1 - \pi(a \mid u)$ and downstream values are unchanged by an update at u, so

$$V^{\bar{\pi}}(u) = \pi(a \mid u) V^{\bar{\pi}}(T(u, a)) + \big(1 - \pi(a \mid u)\big) V^{\bar{\pi}}(T(u, a'))$$
$$= V^{\bar{\pi}}(T(u, a')) - \pi(a \mid u) \big[V^{\bar{\pi}}(T(u, a')) - V^{\bar{\pi}}(T(u, a))\big].$$

The bracket is positive by hypothesis, so increasing $\pi(a \mid u)$ strictly decreases $V^{\bar{\pi}}(u)$.

For the second claim write $A = w(u, a) + V^{\bar{\pi}}_{\text{trop}}(T(u, a))$ and $B = w(u, a') + V^{\bar{\pi}}_{\text{trop}}(T(u, a'))$, so $V^{\bar{\pi}}_{\text{trop}}(u) = \max\{A, B\} = A > B$ and $A^{\bar{\pi}}_{\text{trop}}(u, a') = B - A < 0$. An update increasing $\pi(a' \mid u)$ decreases $\pi(a \mid u)$, so $w'(u, a) < w(u, a)$ and $A' < A$, while $B' = B + \log \frac{\pi'(a'|u)}{\pi(a'|u)}$. Then $V^{\bar{\pi}}_{\text{trop}}(u)' = \max\{A', B'\} < A = V^{\bar{\pi}}_{\text{trop}}(u)$ as soon as $B' < A$, i.e. as soon as $\log \frac{\pi'(a'|u)}{\pi(a'|u)} < A - B = -A^{\bar{\pi}}_{\text{trop}}(u, a')$. Since the left side tends to 0 with the step size and the right side is a fixed positive number, the condition holds for all sufficiently small steps. □

Remark 1 (On the coverage path). *Theorem 12 concerns the $K = 1$ update. The coverage path τ_2 of Section 4.3 may take a non-greedy action at a state shared with τ_1, in which case Lemma 11 need not hold at that state and strict monotonicity may fail there. This is a deliberate trade: the second path is chosen for fragment coverage, not accessibility, and the price is a possible local decrease in the best-witness accessibility in exchange for rehearsing fragments that would otherwise erode. The theorem tells us exactly what is given up and where.*

A.5 The tropical performance difference lemma

The performance difference lemma expresses the change in value between two policies as an expectation, under the new policy's trajectory distribution, of the old policy's advantage. Everything that is hard in classical policy optimization lives in that expectation: it is over a distribution one does not have, its gradient carries the visitation weight $d^{\pi}(s)\pi(a \mid s)$ that makes rare good paths invisible, and convergence rates inherit a distribution mismatch coefficient from it. This section derives the tropical analogue and shows that the expectation is replaced by a sum over a single path. In this section $\lambda > 0$ and the internal reward r are general and terminal scores $g(y)$ may be graded; the binary boundary is the case $g = R_x$.

Lemma 13 (Path form and path identity). *For an actor π let $Q^{\pi}, V^{\pi}, A^{\pi}$ be given by Equation equation 6 with $V^{\pi}(y) = g(y)$ at terminals, and let $W_{\pi}(\tau) = \sum_t \big[\lambda \log \pi(a_t \mid s_t) + r(s_t, a_t)\big] + g(y)$.*

(a) $V^{\pi}(s) = \max_{\tau \text{ from } s} W_{\pi}(\tau)$ for every s.

(b) For every path τ' from s to a terminal y with $g(y) > -\infty$,

$$W_{\pi}(\tau') = V^{\pi}(s) + \sum_{(s_t, a_t) \in \tau'} A^{\pi}(s_t, a_t).$$

Proof. (a) Reverse topological induction. At terminals both sides equal $g(y)$. At a nonterminal s, every path from s is (s, a) followed by a path from $T(s, a)$, and additivity gives $\max_{\tau} W_{\pi}(\tau) = \max_a \big[\lambda \log \pi(a \mid s) + r(s, a) + \max_{\tau'' \text{ from } T(s,a)} W_{\pi}(\tau'')\big] = \max_a Q^{\pi}(s, a) = V^{\pi}(s)$ by the induction hypothesis.

(b) Along τ' every value $V^{\pi}(s_t)$ is finite, since the suffix of τ' from s_t is a path with finite score and (a) gives $V^{\pi}(s_t) \geq$ that score. Then, with $s_H = y$,

$$\sum_{t=0}^{H-1} A^{\pi}(s_t, a_t) = \sum_{t=0}^{H-1} \big[\lambda \log \pi(a_t \mid s_t) + r(s_t, a_t) + V^{\pi}(s_{t+1}) - V^{\pi}(s_t)\big]$$

$$= \big[W_\pi(\tau') - g(y)\big] + V^\pi(y) - V^\pi(s),$$

by telescoping the value terms. Since $V^\pi(y) = g(y)$, this equals $W_\pi(\tau') - V^\pi(s)$. □

Part (b) has no classical counterpart. It says that the deficit of *any* certified path relative to the value is exactly the sum of tropical advantages along it: a deterministic accounting, path by path, with no expectation. Every path has non-positive deficit because every tropical advantage is non-positive, and the witness has deficit zero because every action on it is a maximizer. The lemma is the reason the tropical advantage never needs to be estimated: it is a local quantity computable by the dynamic program, and on the path one trains it vanishes identically.

Theorem 14 (Tropical performance difference lemma)**.** *For actors π, π' and a state s with $V^{\pi'}(s) > -\infty$, let $\tau^\star_{\pi'}(s)$ be a path attaining $V^{\pi'}(s)$. Then*

$$V^{\pi'}(s) - V^\pi(s) = \sum_{(s_t,a_t)\in\tau^\star_{\pi'}(s)} \Big[A^\pi(s_t,a_t) + \lambda \log \frac{\pi'(a_t \mid s_t)}{\pi(a_t \mid s_t)}\Big]. \tag{16}$$

Proof. Rewards and terminal scores do not depend on the actor, so for any path $W_{\pi'}(\tau') = W_\pi(\tau') + \lambda \sum_t \log \frac{\pi'(a_t|s_t)}{\pi(a_t|s_t)}$. Apply this to $\tau' = \tau^\star_{\pi'}(s)$, which by Lemma 13(a) satisfies $W_{\pi'}(\tau') = V^{\pi'}(s)$ and has an accepted terminal, and expand $W_\pi(\tau')$ by Lemma 13(b). □

Set the two lemmas side by side. Classically, $V^{\pi'}(s) - V^\pi(s) = \mathbb{E}_{\tau\sim\pi'}[\sum_t A^\pi(s_t,a_t)]$: the old advantage, averaged over the new policy's trajectories. Tropically the average is over one path, and a second term appears, the log-ratio of the two actors along it. It appears because in tropical RL the per-step reward is the actor's own log-likelihood, so changing the actor changes the rewards; there is no analogue in classical RL, where rewards are exogenous. The log-ratio that trust-region methods introduce by hand as an importance weight is here part of the identity, additively, along a single path.

Corollary 2 (Sandwich, and a witness for the slack)**.** *Under the hypotheses of Theorem 14, with $\tau^\star_\pi(s)$ a path attaining $V^\pi(s)$,*

$$\lambda \sum_{(s_t,a_t)\in\tau^\star_\pi(s)} \log \frac{\pi'(a_t \mid s_t)}{\pi(a_t \mid s_t)} \;\le\; V^{\pi'}(s) - V^\pi(s) \;\le\; \lambda \sum_{(s_t,a_t)\in\tau^\star_{\pi'}(s)} \log \frac{\pi'(a_t \mid s_t)}{\pi(a_t \mid s_t)}. \tag{17}$$

The slack of the left inequality is $W_{\pi'}(\tau^\star_{\pi'}(s)) - W_{\pi'}(\tau^\star_\pi(s)) \ge 0$, the amount by which the new witness beats the old witness under the new actor; the slack of the right inequality is $-\sum_{\tau^\star_{\pi'}(s)} A^\pi(s_t,a_t) \ge 0$, the old-actor deficit of the new witness.

Proof. Lower bound: by Lemma 13(a), $V^{\pi'}(s) \ge W_{\pi'}(\tau^\star_\pi(s)) = W_\pi(\tau^\star_\pi(s)) + \lambda \sum_{\tau^\star_\pi(s)} \log \frac{\pi'}{\pi} = V^\pi(s) + \lambda \sum_{\tau^\star_\pi(s)} \log \frac{\pi'}{\pi}$, and the slack is $V^{\pi'}(s) - W_{\pi'}(\tau^\star_\pi(s)) = W_{\pi'}(\tau^\star_{\pi'}(s)) - W_{\pi'}(\tau^\star_\pi(s))$, nonnegative since $\tau^\star_{\pi'}(s)$ attains the maximum under π'. Upper bound: drop the non-positive advantages in equation 16; the slack is their negative sum. □

The lower bound is what TROPIC's improvement step maximizes with $\pi = \bar\pi_t$: the log-likelihood gain on the current witness. It is a valid lower bound on the true improvement for *any* π', with no penalty term. A penalty-free bound of this kind also exists for the hull distance, where it is the evidence lower bound; what is specific to the vertex distance is that both ends of the sandwich are single-path log-likelihood ratios and that the slack is not a divergence between posteriors but a concrete path, the new witness. When discovery or composition has made a new path the best one under the new actor, that path is the slack, and the slack is the contribution of composition to improvement. Nothing in the classical lemma isolates this quantity.

Corollary 3 (Tropical policy gradient, and the classical weights)**.** *Let the actor be tabular softmax and let the maximizer $\tau^\star$ of $V^{\pi_z}(s_0)$ be unique. Then*

$$\frac{\partial V^{\pi_z}(s_0)}{\partial z_{s,a}} = \begin{cases} \lambda\big(\mathbf{1}\{a = a^\star(s)\} - \pi_z(a \mid s)\big), & s \in S_{\tau^\star}, \\ 0, & \textit{otherwise}, \end{cases} \tag{18}$$

where $a^\star(s)$ is the action of $\tau^\star$ at s. For comparison, with zero internal reward and terminal reward $R \in \{0,1\}$, the expected-return value $V^{\bar\pi}(s_0) = \sum_\tau P_z(\tau)R(\tau)$ satisfies

$$\frac{\partial V^{\bar\pi}(s_0)}{\partial z_{s,a}} = d^\pi(s)\,\pi_z(a \mid s)\,A_{\mathrm{ER}}(s,a), \qquad A_{\mathrm{ER}}(s,a) = V^{\bar\pi}(T(s,a)) - V^{\bar\pi}(s), \tag{19}$$

where $d^\pi(s)$ is the probability that a rollout from s_0 visits s.

Proof. For equation 18, Corollary 1(b) applies to $W_z(\tau) = \sum_t[\lambda \log \pi_z(a_t \mid s_t) + r(s_t,a_t)] + g(y)$, which is continuously differentiable in z, and gives $\nabla_z V^{\pi_z}(s_0) = \lambda \sum_{(s_t,a_t)\in\tau^\star} \nabla_z \log \pi_z(a_t \mid s_t)$. Since $\partial \log \pi_z(a_t \mid s_t)/\partial z_{s,a} = \mathbf{1}\{s = s_t\}\big(\mathbf{1}\{a = a_t\} - \pi_z(a \mid s)\big)$ and a path visits each state at most once, the stated formula follows.

For equation 19, $\partial \log P_z(\tau)/\partial z_{s,a} = \mathbf{1}\{s \in S_\tau\}\big(\mathbf{1}\{a_\tau(s) = a\} - \pi_z(a \mid s)\big)$, so

$$\frac{\partial V^{\bar\pi}(s_0)}{\partial z_{s,a}} = \sum_{\tau \ni s} P_z(\tau)R(\tau)\big(\mathbf{1}\{a_\tau(s) = a\} - \pi_z(a \mid s)\big).$$

Every path through s factors as a prefix from s_0 to s, an action at s, and a suffix from $T(s, a_\tau(s))$, with $P_z(\tau) = P_z(\text{prefix})\pi_z(a_\tau(s) \mid s)P_z(\text{suffix})$ and $R(\tau)$ determined by the suffix. Summing over prefixes gives $d^\pi(s)$ and summing over suffixes gives $V^{\bar\pi}$ of the successor, so $\sum_{\tau\ni s,\, a_\tau(s)=a} P_z(\tau)R(\tau) = d^\pi(s)\pi_z(a \mid s)V^{\bar\pi}(T(s,a))$ and $\sum_{\tau\ni s} P_z(\tau)R(\tau) = d^\pi(s)V^{\bar\pi}(s)$. Substituting,

$$\frac{\partial V^{\bar\pi}(s_0)}{\partial z_{s,a}} = d^\pi(s)\pi_z(a \mid s)V^{\bar\pi}(T(s,a)) - \pi_z(a \mid s)d^\pi(s)V^{\bar\pi}(s) = d^\pi(s)\pi_z(a \mid s)A_{\mathrm{ER}}(s,a).$$

□

The two formulas differ in exactly the way the two performance difference lemmas differ. The classical gradient at (s,a) carries the probability of reaching s, the probability of taking a, and, inside A_{ER}, the probability of completing the task afterwards. On a path whose steps each have probability ρ, every one of these factors is a power of ρ and the gradient at every logit on the path is of order ρ^d. The tropical gradient on the witness carries none of them: it is $\lambda(\mathbf{1} - \pi)$, of order one however improbable the path. Natural policy gradient removes the upstream factor by preconditioning and keeps the completion factor; the tropical gradient removes both, because the path it differentiates is certified and its advantage is identically zero. The next theorem turns this into a rate separation on the simplest instance, with exact gradients, so that neither sampling noise nor value estimation is involved.

Theorem 15 (Exponential versus logarithmic escape on a chain)**.** *Let the graph be a chain $s_0 \to s_1 \to \cdots \to s_d$, where at each s_k ($k < d$) the action $a^\star$ leads to s_{k+1} and the action a' leads to a rejected terminal, and s_d is an accepted terminal. Let the actor be tabular softmax with one free logit per state, $\pi_k = \pi(a^\star \mid s_k) = \sigma(z_k)$, initialized at $\pi_k^{(0)} = \rho \le \frac{1}{2}$ for all k, and take $\lambda = 1$, $r \equiv 0$. Consider gradient ascent with step $\eta > 0$ on the logits.*

(a) (Expected return.) *For $J_{\mathrm{ER}}(z) = V^{\bar\pi}(s_0) = \prod_{k<d} \pi_k$, the first iteration N at which $J_{\mathrm{ER}} \ge 2\rho^d$ satisfies*

$$N > \frac{\log 2}{2\,d\,\eta\,\rho^d}.$$

(b) (Tropical.) *For $J_{\mathrm{trop}}(z) = V^{\bar\pi}_{\mathrm{trop}}(s_0) = \sum_{k<d} \log \pi_k$, after $N_0 = \lceil \frac{2}{\eta} \log \frac{1}{\rho} \rceil$ iterations every $\pi_k \ge \frac{1}{2}$, so $J_{\mathrm{trop}} \ge -d \log 2$, and for every $n > N_0$,*

$$-J^{(n)}_{\mathrm{trop}} < \frac{2d}{\eta\,(n - N_0)}.$$

The bound in (a) is exponential in d; the bound in (b) is logarithmic in $1/\rho$ and independent of d, followed by a $1/n$ rate.

Proof. (a) Since $\log \pi_k = \log \sigma(z_k)$ and $\frac{d}{dz} \log \sigma(z) = 1 - \sigma(z)$,

$$\frac{\partial J_{\mathrm{ER}}}{\partial z_k} = J_{\mathrm{ER}} \frac{\partial \log J_{\mathrm{ER}}}{\partial z_k} = J_{\mathrm{ER}}(1 - \pi_k) \in (0, J_{\mathrm{ER}}).$$

Hence each z_k, each π_k and J_{ER} are non-decreasing along the iteration, and at any iteration n with $J_{\mathrm{ER}}^{(n)} < 2\rho^d$ every logit moves by less than $2\eta\rho^d$. The function $\log \sigma$ is concave, so for $\Delta > 0$, $\log \sigma(z + \Delta) - \log \sigma(z) \le \Delta\,(1 - \sigma(z)) \le \Delta$. Therefore, at any such iteration,

$$\log J_{\mathrm{ER}}^{(n+1)} - \log J_{\mathrm{ER}}^{(n)} = \sum_{k<d} \left[\log \pi_k^{(n+1)} - \log \pi_k^{(n)} \right] < 2\,d\,\eta\,\rho^d.$$

All iterations before N have $J_{\mathrm{ER}} < 2\rho^d$, so summing, $\log J_{\mathrm{ER}}^{(N)} - \log J_{\mathrm{ER}}^{(0)} < 2d\eta\rho^d N$. Since $J_{\mathrm{ER}}^{(0)} = \rho^d$ and $J_{\mathrm{ER}}^{(N)} \ge 2\rho^d$, the left side is at least $\log 2$, which gives the claim.

(b) Here $\partial J_{\mathrm{trop}}/\partial z_k = 1 - \pi_k = \sigma(-z_k)$, so $z_k^{(n+1)} = z_k^{(n)} + \eta\,\sigma(-z_k^{(n)})$ and each z_k is increasing. The coordinates evolve independently and identically, so fix k.

Phase one. While $z_k^{(n)} < 0$ we have $\sigma(-z_k^{(n)}) > \frac{1}{2}$, so $z_k^{(n+1)} - z_k^{(n)} > \eta/2$. The initial logit is $z_k^{(0)} = \log \frac{\rho}{1-\rho} \ge \log \rho = -\log \frac{1}{\rho}$, since $\frac{\rho}{1-\rho} \ge \rho$. Hence z_k reaches 0 within at most $\lceil \frac{\log(1/\rho)}{\eta/2} \rceil = N_0$ iterations, and, being increasing, stays nonnegative thereafter. Thus $\pi_k^{(n)} \ge \frac{1}{2}$ for all $n \ge N_0$ and $J_{\mathrm{trop}}^{(N_0)} \ge -d \log 2$.

Phase two. For $z \ge 0$, $1 + e^z \le 2e^z$, so $\sigma(-z) = \frac{1}{1+e^z} \ge \frac{1}{2}e^{-z}$. Let $y_n = e^{z_k^{(n)}}$ for $n \ge N_0$, so $y_{N_0} \ge 1$. Using $e^x \ge 1 + x$,

$$y_{n+1} = y_n e^{z_k^{(n+1)} - z_k^{(n)}} \ge y_n\big(1 + \eta\,\sigma(-z_k^{(n)})\big) \ge y_n\Big(1 + \frac{\eta}{2y_n}\Big) = y_n + \frac{\eta}{2}.$$

Hence $y_n \ge 1 + \frac{\eta}{2}(n - N_0)$ for $n \ge N_0$. Finally $-\log \pi_k^{(n)} = \log(1 + e^{-z_k^{(n)}}) \le e^{-z_k^{(n)}} = 1/y_n \le \big(1 + \frac{\eta}{2}(n - N_0)\big)^{-1} < \frac{2}{\eta(n-N_0)}$ for $n > N_0$. Summing over the d coordinates gives the bound. □

Remark 2 (Where the separation lies)**.** *On this chain there is a single certified path, so* $J_{\mathrm{trop}} = \log J_{\mathrm{ER}}$ *exactly, and part (b) also describes gradient ascent on* $\log J_{\mathrm{ER}}$*, which is the maximum-likelihood objective of MaxRL. The separation in Theorem 15 is therefore between expected return and its logarithm, and the obstruction in (a) is the factor* J_{ER} *in the gradient, which is the weight* $\omega_{\mathrm{ER}} = P_\theta(\tau)$ *of Proposition 3. What distinguishes the tropical objective from maximum likelihood is not this rate but the object it moves toward, as established in Section A.2 and in Proposition 4: on a problem with several certified paths the maximum-likelihood gradient is spread over them by posterior weight, while the tropical gradient is the score function of one. The theorem is stated with exact gradients to make clear that the exponential cost in (a) is a property of the objective's landscape, not of sampling; no critic, baseline or additional data removes it. With sampled gradients every expected-return method also pays* ρ^{-d} *to observe the path at all, and that cost is the one the archive removes.*

A.6 How TROPIC realizes these results

The algorithm of Section 4.3 is the loop these results describe. The archive is the set of certified vertices of Section A.2; discovery adds vertices and composition adds vertices assembled from existing ones, and by Theorem 3 the set never shrinks. Rescoring under the frozen actor recomputes the distances from the current actor to each vertex. Corollary 1(a) is why the prefix and suffix passes may be run separately and joined. The improvement step is maximum likelihood on that nearest vertex, which Corollary 1(b) identifies as the exact gradient of the tropical value, Corollary 2 identifies as maximizing a penalty-free lower bound on the true improvement, and Theorem 12 shows to be a pointwise improvement step on the whole retained graph. By Theorem 7 and Proposition 3 the step depends on the archive only through its support, which is why acquisition may be as off-policy as the frontier rule makes it, with no correction. By Proposition 4 it is not a step that expected-return information could have selected. What the actor is pulled toward, at every iteration and at every interface, is the nearest computation it has been shown to be capable of; what it is never pulled toward is a distribution over computations, and what it is never allowed to lose is a computation it has already been shown.

B RL ENVIRONMENTS

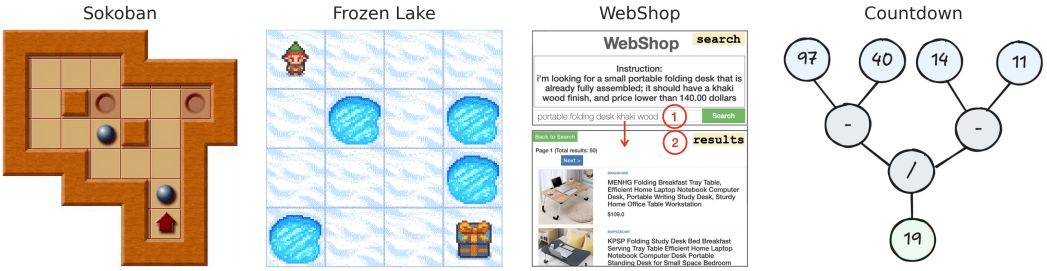

Figure 5: **Environments.** From left to right: **Sokoban**, push the box onto the target without creating a dead end; **FrozenLake**, reach the goal without stepping into a hole; **WebShop**, search for and purchase a product that matches a natural-language instruction; and **Countdown**, combine the given numbers with arithmetic operations to reach the target, here $(97 - 40)/(14 - 11) = 19$.

All four environments in Fig. 5 are multi-turn agentic tasks from RAGEN-2 (Wang et al., 2026). In each, the agent observes a text rendering of the current environment state, emits an action, and receives a binary verifier signal only at the end of the episode. As described in the main body, TROPIC prompts the actor with the problem and the current environment state only, and includes the step count in the state so that the graph of states is acyclic. Two attempts therefore share a state whenever they reach the same environment configuration at the same step, even if they got there by different actions. These shared states are the interfaces at which TROPIC joins a prefix from one attempt with a suffix from another. For every task, we describe below what a state is, what the verifier checks, and why such interfaces arise.

Sokoban. The agent moves on a 6×6 grid and must push a single box onto a target cell. A state is the board configuration, i.e., the positions of the agent and the box. An episode succeeds when the box rests on the target. Pushing the box against a wall or into a corner can make the puzzle unsolvable, so many attempts fail late after a sequence of correct early pushes. For expected-return methods, such attempts yield zero reward and their correct prefixes receive no credit. TROPIC keeps these prefixes in its graph: if another attempt, started from the same board configuration, reaches the target, the two are joined into a verified solution that neither attempt produced alone. Because different move orders often lead to the same board, such shared configurations are frequent.

Countdown. The agent is given three to six input numbers and a target, and must combine the numbers with $+$, $-$, $\times$, and $\div$, using each number exactly once, to reach the target. A state is the multiset of numbers still available, and each step replaces two numbers by the result of an operation on them. The verifier checks that the final number equals the target. Because arithmetic steps on disjoint numbers commute, different attempts frequently reach the same intermediate multiset, e.g., $\{57, 3\}$ from $97 - 40$ and $14 - 11$ in either order. An attempt that computes a useful intermediate result but then fails, and another that finishes correctly from the same intermediate multiset, can therefore be joined into a verified derivation. Training and evaluation use disjoint problem splits.

FrozenLake. The agent navigates a 4×4 grid from a start cell to a goal cell while avoiding holes, which end the episode. Each cell is safe with probability 0.9 when the map is generated, and transitions are deterministic in our experiments. A state is the agent's position. Success requires an entire sequence of safe moves, and a single misstep anywhere ends the episode with zero reward, so failed attempts often contain long safe prefixes. TROPIC retains these prefixes and restarts new attempts from the frontier of safe cells, so a safe route to the middle of the grid found in one attempt can be completed by a suffix found in another.

WebShop. The agent receives a natural-language shopping instruction, e.g., a portable folding desk with a given finish below a price limit, and must search the catalog, navigate result and product pages, select the required attributes, and purchase a matching item. We use the RAGEN-2 `webshop-minimal` adaptation. A state is the current page together with the selections made so far. An episode succeeds only if the purchased item satisfies all requested attributes. Episodes are long,

and a mistake late in the episode, such as a wrong attribute, discards an otherwise correct search and navigation. TROPIC keeps the correct search and navigation steps and can join them with a correct selection-and-purchase suffix found in another attempt from the same product page.

Why these Tasks Favor Composition. The four environments share the structure that makes expected return inefficient and tropical composition effective: the reward is sparse and terminal, failures typically occur after many correct steps, and different attempts pass through the same intermediate states. On-policy baselines must sample a complete successful episode before any of its steps is reinforced. TROPIC instead retains every valid step, including steps from failed attempts, and assembles verified solutions at shared states. This is the mechanism measured directly by FIRST-SOLVE in Sec. 6.1.

C TRAINING AND EVALUATION SETUP

Our implementation builds on RAGEN-2 (Wang et al., 2026) and its VERL training backend (Sheng et al., 2025). We follow the default RAGEN-2 configuration where applicable and describe all task- and method-specific settings below.

Models. We initialize all policies from Qwen2.5-3B-Instruct (Qwen et al., 2025), with 3.1B parameters, and Gemma-4-E4B-it (Gemma Team, 2026), with 4B effective and 8B total parameters including embeddings, i.e., more than twice as many parameters in total. The two models also differ in family, architecture, and pretraining data, which lets us test whether TROPIC's gains depend on a particular model. All models are fully fine-tuned without parameter-efficient adapters.

Baselines. We compare against PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), and RAGEN-2's SNR-Aware Filtering (Wang et al., 2026). All use AdamW with an actor learning rate of 10^{-6}, $(\beta_1, \beta_2) = (0.9, 0.999)$, and weight decay 0.01. PPO and SNR-Aware use a learned critic with learning rate 10^{-5} and Generalized Advantage Estimation (Schulman et al., 2018) with $\gamma = 1$ and $\lambda_{\text{GAE}} = 1$. GRPO and DAPO use group-relative advantages without a critic. For DAPO we use the standard implementation provided by RAGEN-2. The policy-ratio clipping interval is $[0.8, 1.28]$, entropy regularization uses coefficient 0.001, and no reference-policy KL penalty is applied. SNR-Aware Filtering applies the RAGEN-2 rollout-group filter with cumulative probability threshold $p = 0.9$, using the within-group reward standard deviation as its selection score. The other baselines apply no rollout filtering. The additional methods in Table 3 (MaxRL, GiGPO, and GiGPO + T-STAR) use the same backbone, optimizer, and rollout budget, with the remaining hyperparameters set as recommended by their authors.

TROPIC. TROPIC uses the same AdamW optimizer and actor learning rate (10^{-6}) as the baselines, and performs maximum-likelihood updates on verified paths. It uses no critic, policy-ratio clipping, entropy regularization, or reference-policy KL penalty. Each active training problem receives $G = 2$ sequential groups of $n = 8$ attempts per iteration. The first group starts from the initial state; subsequent groups start from a selected frontier state when available, otherwise falling back to the initial state. The resulting $Gn = 16$ rollout starts per problem match the baselines' nominal attempt budget; this does not imply equal generated-token counts or verification costs. Valid transitions are retained up to the graph capacity even when the originating attempt fails, but only complete paths replayed from the initial state and accepted by the verifier are eligible as training targets. Graph memory persists across iterations over a recurring pool of 128 training problems, with eight problems visited per iteration. Table 4 lists the TROPIC-specific settings for the main experiments.

Composition is performed after each attempt group, giving two refreshes per iteration and at most $2 \times 64 = 128$ composition replays per active problem per iteration. This limit does not include verification of sampled successful paths or any additional restart-prefix checks.

Training and Infrastructure. All experiments run on 8 NVIDIA A100 GPUs using Fully Sharded Data Parallel (FSDP) training with BF16 mixed precision. Rollouts are generated with vLLM, and distributed execution is coordinated through Ray. The optimization mini-batch size is 32 sequences, i.e., four sequences per GPU, with gradient accumulation where necessary. All methods train for at

Table 4: **TROPIC hyperparameters.** Shared across tasks and models unless stated otherwise.

Hyperparameter	Description	Value
G	Attempt groups per problem/iteration	2
n	Attempts per group	8
κ	Frontier exploration weight	1.0
–	Training-rollout temperature	1.0
L	Prefixes and suffixes per shared state	4 each
–	Composition candidates per problem/refresh	64
–	Maximum retained edges per problem	1,024
–	Maximum archived solutions per problem	64
–	Edge-rescoring batch size	64
K	Maximum training paths per problem	2
–	Coverage-based path selection	Enabled
–	Update epochs per iteration	2
–	Recurring training-problem pool size	128

most 200 rollout–update iterations on Sokoban, Countdown, and FrozenLake, and at most 100 on WebShop. Validation is performed before training and every 25 iterations.

Evaluation Protocol. Final evaluation uses a fixed set of 512 held-out instances per environment that are not seen during training. Decoding uses temperature $T = 0.5$, top-$p = 1$, and no top-k truncation. At evaluation time, TROPIC uses no archive, restarts, or composition: the trained policy must solve each instance in a single attempt, exactly like the baselines, so the score measures what training has transferred into the model. Each method is evaluated at its final checkpoint, and we report the mean and standard deviation across three independent training seeds.

Metrics. We report the following metrics.

1. **Success rate.** The fraction of evaluation instances solved in a single attempt. Success means placing the box on the target in Sokoban, satisfying the arithmetic verifier in Countdown, reaching the goal in FrozenLake, and completing a fully correct purchase in WebShop.
2. **pass@k.** The probability that at least one of k attempts solves an evaluation instance (see Fig. 1).
3. **Generated tokens per episode.** The number of tokens generated across all turns of an evaluation episode, averaged over successful and unsuccessful episodes. We use each model's tokenizer and exclude prompts, environment observations, and padding. This metric measures inference cost only.
4. **Training wall-clock time.** The elapsed time from the start to the end of training, including rollout generation, optimization, scheduled validation, and checkpointing, and excluding external environment setup and checkpoint preparation.
5. **FIRST-SOLVE** (ablations only). The fraction of solved training instances whose first verified solution was assembled by composition before any complete successful trajectory had been sampled. By definition, it is zero for *No Composition*, which does not compose, and for *Successful-Only*, which composes only from already successful trajectories.

What the Evaluation Measures. We use *compositional* as described in the main body: a solution is compositional when its fragments have been produced in separate attempts but never together in a single one. TROPIC targets exactly this situation during training, and FIRST-SOLVE measures it directly. On Sokoban, 31.6% of solved training problems are first solved by a composition never sampled end to end. The held-out evaluation asks a separate question: whether the policy trained on these assembled solutions solves new instances on its own, in one attempt, without any of TROPIC's search machinery. Since every held-out instance is unseen during training, success requires producing a new solution rather than recalling a stored one. We do not claim systematic generalization in the sense of Lake & Baroni (2018), i.e., to problems drawn from a different distribution of compositions, and leave such splits to future work.

On the Fairness of the Comparison. TROPIC differs from the baselines in three respects that could affect the comparison: it restarts attempts from frontier states, it replays and verifies composed paths, and it uses its own stopping rule. We address each in turn.

Frontier restarts. Restarting from a frontier state requires a deterministic, resettable environment. We regard this as the setting TROPIC is designed for, not an advantage granted to it. To check that restarts alone do not explain the gains, the ablations in Table 3 separate them from composition. TROPIC (No Composition) keeps the archive and frontier restarts and trains by maximum likelihood on sampled verified paths, which makes it closely related to archive-based exploration combined with rejection-sampling fine-tuning. It reaches 40.8% on Sokoban, on par with the strongest baseline (SNR-Aware, 42.2%). Conversely, TROPIC (Root-Only) removes frontier restarts but keeps composition, and reaches 51.8%, above every baseline. The improvement therefore comes from composing fragments across attempts, not from restarts.

Replay and verification. Replaying a composed path executes its stored actions in the environment and queries the verifier. Accordingly, it generates no tokens. Since all methods generate the same number of rollouts per iteration, replay adds only environment calls, which are inexpensive relative to generation in all four tasks.

Stopping and Wall-Clock Time. All methods share the same iteration cap. The baselines stop according to RAGEN-2's criteria, while TROPIC stops early when no training problem yields a new update. Reported wall-clock times therefore reflect the end-to-end cost of each method under its standard protocol, not a comparison at matched compute. This does not favor TROPIC. Since no method can exceed the shared cap, early stopping can only shorten TROPIC's training, never extend it, and each baseline trains for as long as its standard protocol allows. Running every method under its standard protocol is also the setting in which a practitioner would compare them. Success rates are unaffected by this difference: all methods are evaluated at their final checkpoint on the same held-out instances.

D LIMITATIONS

Tropical RL changes a single operation of reinforcement learning: alternative trajectories are combined by a maximum instead of a sum. This one change gives the value three properties that expected return lacks. It names a concrete path, it never decreases as new verified paths are found, and it splits into a best prefix and a best suffix at any shared state. TROPIC uses these properties to assemble solutions from fragments the actor never produced together. On four agentic tasks, this translates into higher single-attempt success than on-policy baselines.

It is worth being precise about the guarantee. The archive, and the capability it certifies, never shrink. For an actor with separate parameters per state, one step on the best path raises the tropical value of every state in the graph (Theorem 4). A language model, however, shares parameters across states and problems, so a step on one problem can still lower the likelihood of a verified path on another. The coverage paths of Section 4.3 mitigate this, but they are a heuristic, not a proof. TROPIC therefore guarantees that no verified computation is lost from *memory*, not that the *actor* never forgets.

This distinction points to the direction we find most promising: continual self-improvement, in which a capability, once verified, is kept and becomes reusable by later ones. Expected return has no mechanism for this, since it stores discoveries only in current probabilities and normalization makes them compete (Proposition 1). The tropical value provides the missing bookkeeping: a set of verified computations that only grows and whose best elements compose. Whether this record can be transferred into shared weights without interference, at scale, remains open.