Epistemic Uncertainty for Test-Time Discovery

Kainat Riaz[1*] **Muhammad Ahmed Mohsin**[1*] Ahsan Bilal[3]
Muhammad Umer[1] Ayesha Mohsin[2] Aqib Riaz[2] Ali Subhan[2]
John M. Cioffi[1]

[1]Stanford University
[2]National University of Sciences and Technology
[3]University of Oklahoma

[*]Equal contribution (core contributors)

Abstract

Automated scientific discovery using large language models relies on identifying genuinely novel solutions. Standard reinforcement learning penalises high-variance mutations, which leads the policy to prioritise familiar patterns. As a result, the maximum reward plateaus even as the average reward increases. Overcoming this limitation requires a signal that distinguishes unexplored regions from intrinsically difficult problems, which necessitates measuring disagreement across independently adapted weight hypotheses rather than relying on a single network's confidence. UG-TTT[1] addresses this challenge by maintaining a small ensemble of low-rank adapters over a frozen base model. The per-token disagreement, quantified as the mutual information between ensemble predictions and weight hypotheses, isolates epistemic uncertainty and identifies positions where insufficient coverage leads to adapter divergence rather than intrinsic problem difficulty. This measure is incorporated as an exploration bonus into the policy gradient, directing the policy toward positions where persistent adapter disagreement signals low training coverage, the same frontier where genuine discovery is possible. A nuclear norm regularizer ensures the adapters remain distinct from one another, thereby preserving the exploration signal throughout training. Across four scientific discovery benchmarks,UG-TTT increases the maximum reward on three tasks, maintains substantially higher solution diversity, and an ablation study confirm that the regularizer is essential for sustaining .

1 Introduction

Large language models (LLMs) are increasingly deployed not merely as assistants but as active participants in scientific discovery, proposing hypotheses, generating candidate algorithms, and searching vast combinatorial spaces that were previously accessible only through expert intuition Qi et al. [2023], Yuksekgonul et al. [2026a], Ranković et al. [2025]. Systems such as AlphaEvolve Novikov et al. [2025] and TTT-DISCOVER have shown that an LLM coupled with a verification loop can rediscover and occasionally improve on human-designed solutions for open mathematical problems. Nevertheless, these systems consistently exhibit an inherent upper bound: while average performance can be enhanced by optimising the expected reward, the maximum attainable reward, which is critical for novel discovery, tends to plateau prematurely. This phenomenon is a direct consequence of mode collapse, where the model converges on simple, 'safe' solutions found in the training data, and a paucity of dense intrinsic signals to guide the search through the sparse rewards of

[1]Project Website — Code Repository

Preprint.

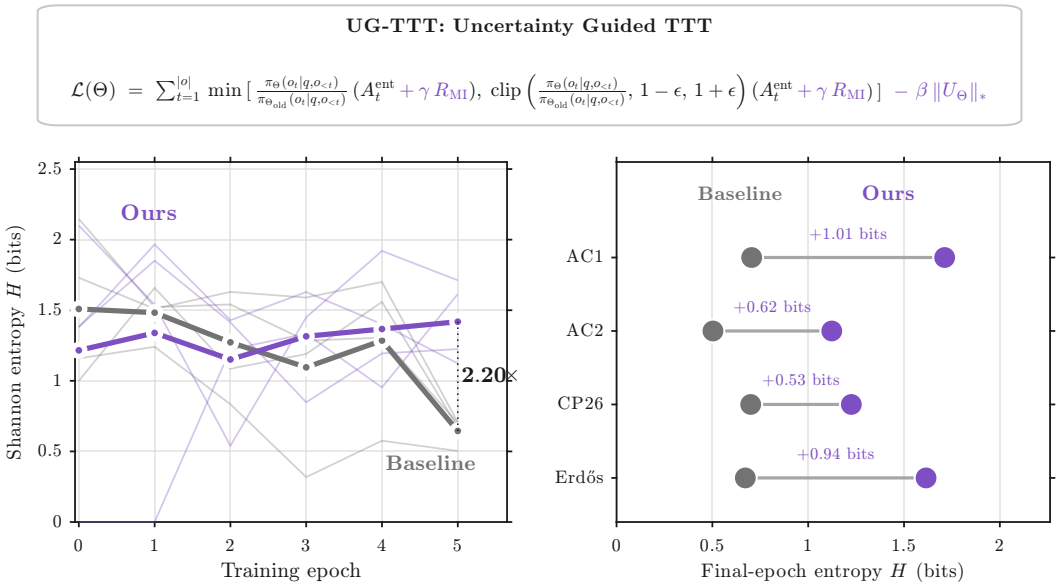


Figure 1: **UG-TTT preserves the diversity that drives discovery.** The training objective (top) augments the TTT-DISCOVER clipped importance-sampling (IS) surrogate with a mutual-information (MI)-shaped advantage and a nuclear-norm regulariser (purple terms). *Left:* Per-epoch Shannon entropy H of the algorithmic family distribution, averaged over the four benchmarks (thin lines show individual tasks); UG-TTT (purple) sustains a cross-task mean $H \approx 1.42$ bits while the TTT-DISCOVER (gray) collapses to 0.65 bits by epoch 5, a $2.20\times$ gap. *Right:* Final-epoch entropy gains: $+1.01$ bits (AC1), $+0.62$ (AC2), $+0.53$ (CP26), $+0.94$ (Erdős), directly enabling the $R_{\max}$ improvements of Sec. 3.

the knowledge frontierYuksekgonul et al. [2026a], Thomas et al. [2025b]. Understanding why this plateau is structurally inherent to the expected-value objective, rather than an incidental failure of search, is the starting point of our work.

Discovery is an out-of-distribution task. A genuine discovery is a solution that lies beyond prior human knowledge and, consequently, outside any pretrained model's training distribution. This framing has an immediate consequence: any method that searches high-probability regions of a model's learned distribution primarily performs retrieval or exploitation. However, scientific discovery demands ideas beyond the training data. Therefore, progress hinges on a key question: how can we push an LLM to navigate its knowledge boundaries, where its initial predictive confidence is low, but the potential for meaningful epistemic shift is high?

Reinforcement learning as a reward. Existing test-time training frameworks use Predictor + Upper Confidence Bound applied to Trees (PUCT) Rosin [2011] to select parent solutions, after which the policy generates mutations for scoring by a verifier. The failure to improve the maximum reward is structural, originating in the update rule itself: the standard reinforcement learning objective maximizes expected reward, which rewards predictive reliability and penalizes variance Cheng et al. [2026], Li et al. [2025]. At the gradient level, this installs a mechanistic bias toward mutations with well-estimated outcomes, filtering out uncertain but potentially high-payoff directions. The consequence is diversity collapse, where the model converges on a narrow set of low-risk mutations, accumulating exploitative gains in average reward at the expense of the singular breakthrough state that discovery actually requires Yuksekgonul et al. [2026a], Thomas et al. [2025b].

Why disagreement is the right signal. Existing fixes for diversity collapse, including entropy bonuses, confidence-based difficulty weighting, and uncertainty-guided reasoning budgets Cheng et al. [2026], Li et al. [2025], all derive their signal from a single trained network, which cannot separate *aleatoric uncertainty* (irreducible input ambiguity) from *epistemic uncertainty* (gaps in what the model has learned).This is because a single network produces high entropy for both signals; distinguishing them requires measuring weight-space disagreement, a signal absent in single trajectories Hüllermeier and Waegeman [2021], Balabanov and Linander [2024], Yadkori et al. [2024].

For discovery, this conflation is fatal: the policy spends compute on ambiguous-but-familiar territory rather than the genuine knowledge frontier where breakthroughs live. AutoDiscovery [Agarwal et al., 2025] compounds the problem further, deriving Bayesian surprise from a single frozen LLM and therefore accumulating no domain knowledge as discoveries proceed. Bayesian theory resolves the confound: mutual information between predictions and model parameters isolates the epistemic component as disagreement among independently adapted weight hypotheses Shorinwa et al. [2025], Yadkori et al. [2024], so ensemble members that agree indicate mere ambiguity, while members that disagree mark a region the model has genuinely not learned. LoRA ensembles realize this decomposition without the cost of full model replication, treating each low-rank adapter as an independent posterior sample over a shared frozen base Wang et al. [2023], Balabanov and Linander [2024]. The remaining obstacle is collapse: jointly trained adapters share data and objective and thus converge toward identical weight configurations, driving mutual information to numerical zero Zhai et al. [2026], a degeneracy our measurements confirm within two epochs.

Contributions. We introduce Uncertainty-Guided Test-Time Training (UG-TTT), an effective framework for autonomous scientific discovery that overcomes the structural limitations of existing test-time systems. Our method extends the test-time discovery objective of the TTT-DISCOVER baseline Yuksekgonul et al. [2026a] with three coupled contributions: (i) An ensemble of K LoRA adapters over a shared frozen base, whose per-token disagreement, quantified as mutual information $\mathrm{MI}_t = H(\bar{p}(\cdot \mid o_{<t})) - \frac{1}{K}\sum_{k=1}^{K} H(p_k(\cdot \mid o_{<t}))$ where H is the Shannon entropy, $p_k(\cdot \mid o_{<t})$ is the predictive distribution of the k-th adapter given previous tokens $o_{<t}$, and $\bar{p}_t = \frac{1}{K}\sum p_k$ is the arithmetic mixture distribution, supplies a Bayesian epistemic-uncertainty signal that localises positions the model has not yet resolved. (ii) A shaped advantage that injects this signal into the policy gradient with a gain coupled to the entropic temperature β, so exploration pressure rises automatically as the policy sharpens. (iii) A nuclear-norm regulariser on the stacked adapter down-projections, chosen because its global maximum is provably attained at configurations in which each adapter occupies a mutually orthogonal input subspace (Prop. 2). This is the precise condition that keeps the BALD decomposition non-trivial, and is a property that prior LoRA regularisers, focused on redundancy reduction, do not provide. Empirically UG-TTT sustains 3 to 4 orders of magnitude higher ensemble mutual information than an unregularised baseline, retains 1.1 to 1.7 bits of solution-family entropy where the baseline converges to a single family, and improves the maximum reward on three of four discovery benchmarks from the TTT-DISCOVER baseline at strictly lower token cost (Fig. 1).

2 Uncertainty-guided test-time training

A discovery-oriented policy must direct its sampling budget toward positions the model has not learned to score, not toward positions that are intrinsically ambiguous; the predictive entropy of a single network does not separate these two regimes [Hüllermeier and Waegeman, 2021, Yadkori et al., 2024]. The Bayesian Active Learning by Disagreement (BALD) identity isolates the epistemic component as the mutual information between the predictive distribution and the weight posterior, which vanishes when ensemble members agree and grows when they place mass on incompatible continuations [Houlsby et al., 2011, Cao and Tsang, 2021]. We estimate this quantity from a small ensemble of low-rank adapters, use it as a per-token exploration term in the policy gradient, and constrain the adapter parameters to keep the estimator non-degenerate throughout training.

2.1 Setup

An instance of LLM-driven discovery is an environment $\mathcal{E} = (d, \mathcal{S}, \mathcal{A}, R, \mathcal{T})$ with natural-language task description d, solution space $\mathcal{S}$, action set $\mathcal{A}$ of next-token continuations, deterministic verifiable reward $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ returned by a program checker, and transition $\mathcal{T}$ that appends an action a to a parent state s. A policy $\pi_\theta(a \mid q, s)$ generates a response $o = (o_1, \ldots, o_{|o|})$ of thinking tokens and executable code, conditioned on a prompt q embedding (d, s). The metric of interest is the per-instance maximum reward $\max_{(s,a)} R(s, a)$: a single breakthrough dominates, and improving the average without moving the maximum has not advanced discovery.

At each training step we sample G rollouts per parent state, score each with $R^{(i)}_{\mathrm{exec}}$, and form leave-one-out advantages A_i through an adaptive temperature $\beta(s)$ computed from R_{exec} alone (parent-state selection, the KL-budgeted bisection for $\beta(s)$, and the clipped importance-sampling loss $\mathcal{L}_{\mathrm{PG}}$

are in Appendix A). The temperature $\beta(s)$ grows monotonically as the within-group reward distribution sharpens and therefore acts as a state-dependent measure of policy confidence, a property Sec. 2.3 uses to couple the exploration coefficient to the onset of mode contraction.

2.2 Token-level epistemic uncertainty from a LoRA ensemble

We replace the single TTT-DISCOVER adapter with K low-rank perturbations sharing a frozen base,

$$\Theta^{(k)} \;=\; \Theta_{\text{base}} + B^{(k)}A^{(k)}, \qquad A^{(k)} \in \mathbb{R}^{r\times d_{\text{in}}},\; B^{(k)} \in \mathbb{R}^{d_{\text{out}}\times r},\; r \ll \min(d_{\text{in}}, d_{\text{out}}), \tag{1}$$

for $k \in \{1, \dots, K\}$. The K adapters are read as samples from a variational posterior $q(\theta \mid \mathcal{D})$ over the shared base, and at every position t the corresponding posterior predictive is the arithmetic mixture $\bar{p}_t = \frac{1}{K}\sum_k p_{\theta_k}(\cdot \mid q, o_{<t})$. With this construction the standard Bayesian Active Learning by Disagreement (BALD) decomposition isolates the epistemic component of the predictive uncertainty at t:

$$\text{MI}_t \;=\; H(\bar{p}_t) \;-\; \frac{1}{K}\sum_{k=1}^{K} H(p_{\theta_k}(\cdot \mid q, o_{<t}))\,, \tag{2}$$

where H is Shannon entropy over the vocabulary [Houlsby et al., 2011, Cao and Tsang, 2021, He et al., 2025a]. MI_t is non-negative, vanishes exactly when every member assigns identical mass to the next token, and grows precisely when members place mass on incompatible alternatives, the operational hallmark of a position the model has not yet resolved. Subtracting the mean per-member entropy strips out the in-member (aleatoric) component, leaving only disagreement attributable to the spread of the posterior over weights. Both terms are evaluated on the full (K, chunk, V) logit tensor inside a chunked LM head; a memory-bounded implementation is described in App. D.

The policy-gradient update operates on whole rollouts, but most positions in a code rollout are syntactic boilerplate, such as indentation, brackets and the keywords, on which the adapters trivially agree. We therefore reduce $\text{MI}_t^{(i)}$ to a single scalar per rollout by averaging only over the most uncertain positions,

$$U_i \;=\; \underset{t\in[1,|o_i|]}{\text{top-7\%-mean}}\, \text{MI}_t^{(i)}, \tag{3}$$

which makes U_i length-robust and concentrates it on the small minority of decoding steps that carries the meaningful signal. He et al. [2025b] report that their entropy-triggered pause-and-rerank mechanism fires on 3.95%–11.88% of decoding steps across eight LLMs spanning 0.6B–8B parameters, identifying these as the high-uncertainty positions where the ground-truth token is most often mis-ranked. We adopt 7% as a heuristic order-of-magnitude choice within this range; full calibration is in App. D.

Two sampling decisions complete the picture. Rollouts are generated *round-robin*, so each adapter produces G/K rollouts per group and the population of generators stays balanced across training. Scoring then evaluates *every* rollout through *all* K adapters in a single chunked forward pass, so MI_t is well-defined for every token of every rollout regardless of which adapter generated it; each adapter therefore receives gradient signal from all G rollouts (full schedule in Alg. 1, App. B).

2.3 Uncertainty-shaped advantage

Two transformations are required before U_i can serve as an additive correction to A_i. First, the raw scale of U_i varies by orders of magnitude across tasks and across training epochs, so a fixed coupling coefficient would over-explore on one benchmark while under-exploring on another; we therefore standardise U_i within each rollout group. Second, a constant exploration weight maintains uniform pressure throughout training, whereas mode collapse occurs precisely in the regime where $\beta(s)$ grows large; we therefore couple the exploration coefficient to $\beta(s)$, so that exploration pressure scales monotonically with policy sharpness.

We address both with a shaped advantage. Within each group we centre and clip U,

$$\tilde{U}_i \;=\; \text{clip}\!\left(\frac{U_i - \overline{U}}{\sigma_U},\; -3,\; 3\right), \qquad \overline{U} = \tfrac{1}{G}\sum_i U_i,\;\; \sigma_U = \sqrt{\tfrac{1}{G-1}\textstyle\sum_i (U_i - \overline{U})^2}, \tag{4}$$

with the convention that $\tilde{U}_i \equiv 0$ whenever $\sigma_U = 0$ (the degenerate case in which all rollouts in the group share an identical U_i). This produces a zero-mean unit-variance signal of common scale

across tasks. The threshold ± 3 follows the standard three-sigma convention for robust advantage normalisation and lies just above the bound $|z_i| \leq \sqrt{G-1}$ that Cauchy–Schwarz imposes on the centred unbiased z-scores, so for $G \leq 10$ the clip is provably inactive (Proposition 1) and acts as a forward-compatible safeguard for $G \geq 11$, where $\sqrt{G-1} > 3$. We then couple the exploration coefficient to the entropic temperature,

$$\gamma_{\text{eff}}(s) \;=\; \alpha \cdot \min\!\left(\frac{\beta(s)}{\beta_{\text{ref}}},\; \gamma_{\max}\right), \tag{5}$$

where α is a base rate, β_{ref} is a reference temperature, and $\gamma_{\max}$ is a safety clip. The shaped advantage that replaces A_i in the per-adapter policy-gradient loss $\mathcal{L}_{\text{PG}}^{(k)}(\theta_k)$ is

$$A_i^{\text{shaped}} \;=\; A_i \;+\; \gamma_{\text{eff}}(s) \cdot \overline{|A|} \cdot \tilde{U}_i\,, \qquad \overline{|A|} \;=\; \frac{1}{G}\sum_j |A_j|, \tag{6}$$

where the prefactor $\overline{|A|}$ rescales the exploration bonus to the same magnitude as the within-group reward signal. A near-uniform group with small $\overline{|A|}$ therefore damps the exploration bonus proportionally, while a decisive group permits it to compete with the executable reward advantage A_{exec}.

Remark 1 (*β-coupling aligns exploration with collapse onset*). $\beta(s)$ increases monotonically as the within-group rewards equalise, and $\beta \to \infty$ marks the limit in which the policy contracts onto a single template. Equation (5) scales γ_{eff} at the same rate, so exploration pressure intensifies with policy sharpening rather than along a fixed schedule, and the bonus remains attenuated while the policy retains mass across modes.

Proposition 1 (*Bounded scale perturbation of the shaped advantage*). *Let $z_i = (U_i - \overline{U})/\sigma_U$ and $\mathcal{C} = \{i : |z_i| > 3\}$. Then $\left|\sum_i A_i^{\text{shaped}} - \sum_i A_i\right| \leq \gamma_{\text{eff}}(s)\,\overline{|A|}\,|\mathcal{C}|(\sqrt{G-1} - 3)$, and the bound vanishes whenever the clip is inactive ($\mathcal{C} = \varnothing$).* The bonus is intentionally biased toward high-MI rollouts; this directional bias *is* the exploration mechanism. Proposition 1 (proof in App. C) bounds only the residual scale perturbation, which is empty in practice: U_i rarely concentrates beyond three group-standard-deviations from the mean for the group sizes we use ($G \leq 64$).

2.4 Subspace-diversity regulariser

The objective so far does not preclude the K adapters from converging to a common solution. Each adapter receives identical rollouts and shaped advantages, so identical gradients drive their parameters toward a shared optimum and any initial disagreement erodes. In pilot runs the row spaces of the down-projections aligned within two to three epochs (pairwise cosine $\to 1$), U_i fell below 10^{-3} (Table 2), and the entire MI bonus quietly collapsed to zero. The exploration mechanism of the previous subsection then degraded silently into ordinary RL on K tied adapters.

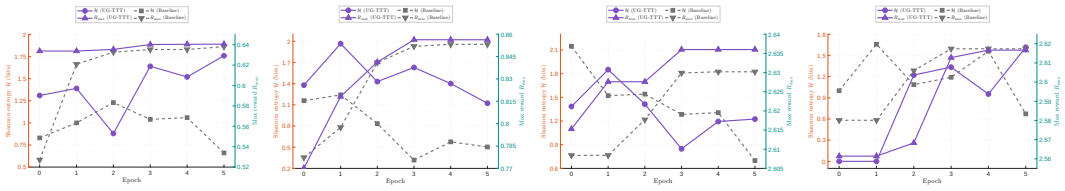
To prevent this we maximise the nuclear norm of the per-layer stacked down-projections (Nuclear-Norm Maximisation, NNM). For each tracked linear layer ℓ, write

$$W_\ell \;=\; \left[A_\ell^{(1)};\, A_\ell^{(2)};\, \ldots;\, A_\ell^{(K)}\right] \;\in\; \mathbb{R}^{Kr \times d_{\text{in}}}, \qquad \mathcal{L}_{\text{NNM}} \;=\; -\,\frac{1}{L}\sum_{\ell=1}^{L} \left\|W_\ell\right\|_*, \tag{7}$$

where $A_\ell^{(k)}$ is the down-projection of adapter k at layer ℓ, $\|\cdot\|_*$ is the nuclear norm (sum of singular values), and L is the number of tracked layers; the negative sign converts minimisation into a maximisation of $\sum_\ell \|W_\ell\|_*$. We apply the regulariser to the down-projections $A^{(k)}$ only and leave the up-projections $B^{(k)}$ unconstrained, because $A^{(k)}$ controls *which* input directions an adapter attends to while $B^{(k)}$ must remain free to deliver the output magnitude the RL signal demands; joint regularisation oscillates in pilot runs and in Zhai et al. [2023] (App. D).

What makes this the right choice rather than any other low-rank penalty is geometric:

Proposition 2 (*The global maximum is subspace-orthogonal*). *Fix a common adapter-wise Frobenius norm $\|A_\ell^{(k)}\|_F = c$ for all $k \in \{1, \ldots, K\}$. Whenever $Kr \leq d_{\text{in}}$, the global maximum of $\|W_\ell\|_*$ on this constraint set consists of K adapter blocks $\{A_\ell^{(k)}\}_{k=1}^K$ whose row spaces are mutually orthogonal in $\mathbb{R}^{d_{\text{in}}}$, and each block has r equal singular values, all equal to $c/\sqrt{r}$.*



(a) AC1 (b) AC2 (c) CP26 (d) Erdős

Figure 2: **Training dynamics across all four benchmarks.** Per-epoch Shannon entropy H of the family distribution over correct rollouts (orange, left axis) and cumulative $R_{\max}$ (teal, right axis); dashed is baseline, solid is UG-TTT. The baseline contracts H monotonically on every problem while UG-TTT sustains higher H and matches or exceeds the baseline reward ceiling, so diversity preservation does not trade off against $R_{\max}$.

A proof appears in App. C. In the equal-norm regime that AdamW with shared LoRA initialisation approximately maintains, the approximate maximisers are configurations where each adapter attends to a distinct rank-r input subspace. This is the geometric condition under which the BALD decomposition in Eq. (2) remains non-trivial: members trained on overlapping input subspaces drift back to identical predictive distributions and zero out MI_t, which is exactly the failure mode we documented above. Existing LoRA regularisers, including those targeting redundancy reduction or low-rank stability, do not provide this guarantee because their stated objectives concern individual matrices rather than the joint geometry of the ensemble.

2.5 Training objective

The per-adapter policy-gradient loss with shaped advantage, a frozen-base KL anchor, and the diversity regulariser of Eq. (7) combine into a single per-step objective,

$$\mathcal{L}_{\text{total}} = \underbrace{\frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_{\text{PG}}^{(k)}(\theta_k)}_{\text{ensemble RL}} + \underbrace{\lambda_{\text{KL}}\,\mathrm{KL}\big(\pi_\theta \,\|\, \pi_{\text{base}}\big)}_{\text{base anchor}} + \underbrace{\lambda_{\text{NNM}}\,\mathcal{L}_{\text{NNM}}}_{\text{diversity}}. \tag{8}$$

Algorithm 1 iterates over the K adapters sequentially using the parallel multi-LoRA training framework of Ye et al. [2024] to bound activation memory (App. B), and the per-token reward-shaping estimator used for the KL anchor follows Yuksekgonul et al. [2026b] (App. A).

Proposition 3 (TTT-DISCOVER *as a special case*). *If either (a) $K=1$, or (b) all K adapters share identical weights at every step, and additionally $\lambda_{\text{NNM}}=0$, then $\mathcal{L}_{\text{total}}$ in Eq.* (8) *reduces to the policy-gradient objective of Yuksekgonul et al. [2026a].* A proof appears in App. C.

3 Experiments

We evaluate UG-TTT on four TTT-DISCOVER benchmarks [Yuksekgonul et al., 2026b], namely a step-function autocorrelation inequality (**AC1**, 1000-element sequence, minimise C_1), the autocorrelation lower-bound problem (**AC2**, maximise C_2), circle packing of n=26 unit-square circles (**CP26**, maximise the radius sum) and the Erdős minimum-overlap integral (**Erdős**, minimise C_5). We ask whether UG-TTT discovers stronger solutions (§3.1), whether preserved ensemble diversity drives those gains (§3.2) and which components carry that mechanism (§3.3).

Setup. All runs use Qwen3-8B Yang et al. [2025] with K=5 rank-16 LoRA adapters, λ_{NNM}=0.075, α=0.1, group size G=8 with 8 groups per batch (64 rollouts per epoch), for 6 epochs on one RTX Pro 6000 (96 GB; ≈32 h per run), with TTT-DISCOVER prompts and verifiers unchanged. A rollout is correct when the verifier returns reward >0, and each correct rollout receives one algorithmic family label from the first matching rule in an ordered regular-expression list (otherwise `other`).

Table 1: **Main results across the four TTT-DISCOVER benchmarks.** Higher $R_{\max}$ is better; R is $1/$upper bound on AC1 and Erdős and a lower bound on AC2 and CP26. The CP26 ceiling matches the published TTT-DISCOVER state of the art, first reached at epoch 3.

	$R_{\max}$ during training			H at epoch 5 (bits)	
Problem	Baseline	UG-TTT	Δ	Baseline	UG-TTT
AC1	0.6381	**0.6406**	+0.0025	0.70	**1.71**
AC2	0.8532	**0.8563**	+0.0031	0.50	**1.12**
CP26	2.6302	**2.6359**	+0.0058	0.70	**1.23**
Erdős	**2.6174**	2.6167	−0.0007	0.67	**1.62**

Streaming MI: enabled on AC1 and CP26; disabled on AC2 and Erdős (paired calibration in §3.3, full diagnostic in Appendix F).

3.1 Main results

Table 1 reports training-time $R_{\max}$ and final-epoch family entropy H. UG-TTT improves $R_{\max}$ on three of four problems, matches the fourth to within 0.03% of the bound ($\Delta = -0.0007$ on Erdős), and retains 1.1 to 1.7 bits of H in every case while the baseline collapses below 0.71 bits.

Absolute Δs mask where each gain sits on the discovery curve. On AC1, UG-TTT crosses $R \geq 0.63$ at step 12 versus step 189 for the baseline ($\approx 16\times$ faster), and the baseline never reaches 0.64 within budget. The CP26 $+0.0058$ matches the published TTT-DISCOVER ceiling of 2.6359, which UG-TTT matches in 384 rollouts (6 gradient steps $\times$ 8 groups $\times$ G=8) versus the 25,600 rollouts (50 steps of 512) used by TTT-DISCOVER on the same Qwen3-8B model (§4.1.2 and Tab. 9 of Yuksekgonul et al. [2026a]). On AC2, the $+0.0031$ flip comes from the `differential_evolution` family, which yields zero correct rollouts under the unregularised ensemble (Tab. 5). Erdős ties on the scalar bound, yet UG-TTT logs 13 new-best events versus 7 for the baseline ($+86\%$) and nearly doubles the discretisation resolution (mean $n_{\text{points}} = 170$ vs. 89), surfacing the Fourier initialisation revisited in §3.3.

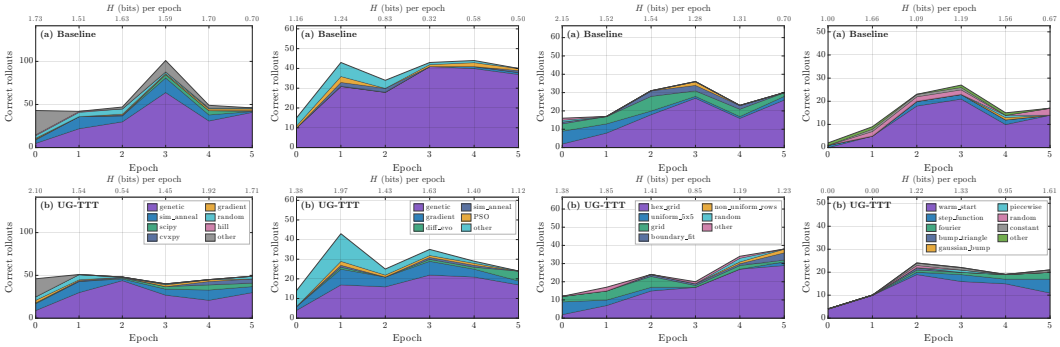
3.2 Diversity is the mechanism

Figure 3 tracks the per-epoch family mixture over correct rollouts on every problem. The pattern is uniform across the two family taxonomies (outer search loop on AC1/AC2, structural initialisation on CP26/Erdős). The baseline narrows monotonically to $H \leq 0.71$ bits with one family at 82 to 93% of rollouts, whereas UG-TTT either holds a wider portfolio throughout (AC1, CP26) or opens across training (AC2, Erdős) and finishes with ≥ 3 live families on every problem. The collapse is therefore a universal failure mode of TTT-DISCOVER-style RL on discovery, and the diversity regulariser averts it. Mechanistically the β-coupling of Remark 1 (Eq. (5)) drives the effect, since γ_{eff} rises with reward-seeking pressure and sustains exploration on underrepresented families at the moment the baseline contracts. §3.3 confirms the link, as removing NNM drives MI to $\sim 10^{-5}$ within two epochs and the family distribution narrows identically to the single-adapter baseline.

The families that survive under UG-TTT are exactly those that yield the winning solutions, namely the Cauchy-Schwarz target $\sqrt{2n}$ on AC1 (18/30 best rollouts versus 1/30 for the baseline), the Fourier ansatz $h(x) = a + b\sin(kx)$ on Erdős and `scipy.differential_evolution` on AC2. Each motif sits inside a family the baseline extinguishes by the final epoch (Fig. 3), so preserved diversity is a precondition for discovery. Full per-problem adoption rates appear in App. G.

3.3 Ablation and computational cost

Table 2 ablates NNM against the full method on CP26 (the same 5-LoRA comparator as Tab. 1). Without NNM, row spaces align within two epochs, ensemble MI falls to $\sim 10^{-5}$ (3 to 4 orders of magnitude below the regularised run; Table 2), the MI bonus zeroes out, and the run reduces to unregularised 5-LoRA RL; $R_{\max}$ stalls at 2.6302, while the full method matches that ceiling on 25% fewer training tokens (2.04×10^6 versus 2.72×10^6). NNM is therefore the load-bearing component, keeping the ensemble informative enough for the MI bonus to deliver real exploration pressure to the policy gradient.



(a) AC1 (b) AC2 (c) CP26 (d) Erdős

Figure 3: **Diversity collapse is universal and UG-TTT averts it.** Per-epoch family composition of correct rollouts on each benchmark; within each panel, top row is baseline (1 LoRA) and bottom is UG-TTT (5 LoRA + MI + NNM), with $H_{\text{start}} \to H_{\text{end}}$ reported above. Family vocabularies are problem-specific and identical across conditions, so the contrast lies in which families remain populated. Terminal $H = 1.71/1.12/1.23/1.62$ bits under UG-TTT versus $0.70/0.50/0.70/0.67$ under the baseline.

Table 2: **CP26 component ablation.** Removing NNM collapses ensemble MI within two epochs, zeroing the MI bonus and reverting the run to unregularised 5-LoRA RL. Mean MI is per-rollout, averaged over epoch 5. "Tokens to own $R_{\max}$" is cumulative `rollout/num_tokens` until the run first hits its own ceiling, while "Tokens to $R=2.6302$" is the cost of matching the no-NNM ceiling, giving a direct compute comparison at fixed reward.

Configuration	$R_{\max}$	Mean MI (ep. 5)	Tokens to own $R_{\max}$	Tokens to R=2.6302
Full UG-TTT (NNM + MI bonus)	**2.6359**	$\mathbf{2.2\times10^{-2}}$	$\mathbf{2.33\times10^{6}}$	$\mathbf{2.04\times10^{6}}$
No NNM (MI bonus collapses to 0)	2.6302	6.6×10^{-6}	2.72×10^{6}	2.72×10^{6}

Streaming MI, with paired calibration on AC2 and Erdős. Streaming early-stop (App. F) caps the thinking phase and re-checks ensemble MI before continuing, and we run it by default. Rollouts above the MI threshold continue past the cap to natural termination; only below-threshold rollouts are truncated, so the firing rate is the MI gate, not a hard length cut. It fires in 50.0% of AC1 rollouts and 55.6% of CP26 rollouts (saving 569 and 757 tokens, roughly $5\times$ the unregularised CP26 baseline rate of 10.7%), driving the 25% token-budget reduction in Tab. 2; the Tab. 1 $R_{\max}$ rollouts on AC1 and CP26 are themselves streamed. On AC2 and Erdős, paired controls show streaming reducing $R_{\max}$ by 0.0148 and 0.0017 respectively, so we disable it there and report no-streaming numbers in Tab. 1. No length-versus-quality diagnostic predicts this asymmetry in advance, so the calibration is empirical. The Erdős tie reflects UG-TTT matching the bound through a distinct family (Fourier basis with SLSQP at $n_{\text{points}} = 400$) that the single-adapter baseline never enters.

Limitations

We evaluate on Qwen3-8B with $K = 5$, one GPU configuration and a single seed across four benchmarks, so entropy gains and per-problem streaming behaviour may not transfer to other base models, larger K, or different initialisations. The ensemble incurs a roughly K-fold parameter overhead and a chunked K-way scoring pass, partially offset on AC1 and CP26 by streaming early-stop. The three coupled hyperparameters ($\alpha, \beta_{\text{ref}}, \gamma_{\max}$) and the fixed top-7% token threshold remain heuristic, and streaming calibration required paired pilots on AC2 and Erdős. UG-TTT inherits a deterministic program-checker reward from TTT-DISCOVER, so porting to learned or noisy verifiers would require recalibrating the MI bonus.

4 Background and related work

RLHF and reward model optimization. Reinforcement learning from human feedback (RLHF) has become the standard mechanism for aligning LLMs with human preferences Havrilla et al. [2024], Hong et al. [2025], Rafailov et al. [2024], producing gains in reasoning, commonsense, and code generation through Proximal Policy Optimization (PPO)-style optimization against a learned reward model Guo et al. [2025], Havrilla et al. [2024], Hong et al. [2025]. Its central pathology is reward hacking: scalar rewards compress nuanced objectives into a single dimension, inherently suppressing exploration and enabling the exploitation of proxy loopholes while inducing overconfidence, hallucination, and distributional fragility Khalaf et al. [2025], Rafailov et al. [2024], Xia et al. [2025], Shorinwa et al. [2025], Farquhar et al. [2024], Tao et al. [2025].

Test-time training. Test-time training (TTT) adapts model parameters at inference using self-supervised signals or retrieved neighbors Hu et al. [2025], Hardt and Sun [2023], Akyürek et al. [2024], often with parameter-efficient adapters to mitigate forgetting Hu et al. [2025], Moradi et al. [2025], and has delivered substantial gains on the Abstraction and Reasoning Corpus (ARC) and BIG-Bench Hard Akyürek et al. [2024]. Like RLHF, it inherits mode collapse under sparse outcome rewards Thomas et al. [2025a]: without dense intrinsic signals, adaptation converges prematurely on familiar solution paths rather than advancing toward the knowledge frontier.

Uncertainty quantification. Foundational work on uncertainty quantification (UQ) separates aleatoric from epistemic uncertainty and proposes Bayesian networks, deep ensembles, Monte Carlo dropout, and evidential methods as estimation strategies Abdar et al. [2021], Charpentier et al. [2022], Hüllermeier and Waegeman [2021], Caldeira and Nord [2021]. In LLMs, UQ has been deployed for hallucination detection through token probabilities, verbalized confidence, ensemble disagreement, and conformal intervals Xia et al. [2025], Shorinwa et al. [2025], Farquhar et al. [2024], Tao et al. [2025], Liu et al. [2025], Vashurin et al. [2025], Huang et al. [2025], with production systems such as LM-Polygraph Fadeeva et al. [2023] and entropy-based confabulation detectors Farquhar et al. [2024]. These methods remain post-hoc: they flag unreliable outputs but do not integrate into the training signal, and most fail to cleanly separate epistemic from aleatoric sources at LLM scale Liu et al. [2025], Abdar et al. [2021].

Intrinsic motivation and MI-based exploration. Curiosity bonuses and entropy regularization have been proposed to counter exploitation bias in RL with LLMs Du et al. [2023], Liao et al. [2025], Gao et al. [2025], Dwaracherla et al. [2024], Levy et al. [2025], Cheng et al. [2026], though classical novelty rewards suffer vanishing incentives in large action spaces Dai et al. [2025], Yuan et al. [2022]. Recent work uses ensemble disagreement, including LoRA-based variants, as an intrinsic signal tied to the knowledge boundary Shorinwa et al. [2025], Hüllermeier and Waegeman [2021], with actor-wise perplexity bonuses and critic-wise value variance integrated into advantage functions Dai et al. [2025]. Frameworks such as Curiosity-Driven Exploration (CDE) and i-MENTOR demonstrate that maintained entropy prevents premature convergence in sparse-reward reinforcement learning with verifiable rewards (RLVR) Dai et al. [2025], Gao et al. [2025], but balancing intrinsic against outcome rewards risks calibration collapse Dai et al. [2025]. None of these approaches target epistemic uncertainty at the granularity required for frontier exploration, which is the gap UG-TTT addresses.

5 Conclusion

We argue that diversity collapse in test-time discovery systems is a structural consequence of single-model, scalar-reward optimization: policy gradients suppress high-variance candidates, and a single network cannot disentangle epistemic from aleatoric uncertainty. UG-TTT addresses both by introducing an ensemble of low-rank adapters that provides a token-level mutual information signal, combined with a nuclear norm regularizer to maintain subspace diversity and a temperature-coupled exploration bonus.

UG-TTT improves maximum reward on three of four benchmarks and matches the fourth while discovering qualitatively distinct solutions, sustaining substantially higher mutual information and late-stage entropy. Ablations identify the nuclear norm regularizer as critical, with its removal causing rapid collapse of the uncertainty signal and reversion to standard reinforcement learning behavior.

LLM usage

`Sonnet 4.6` by `Anthropic` was used for editing, proofreading, and resolving LaTeX formatting issues in this document. `Claude Code` by `Anthropic` was used to assist with code writing in the implementation of `UG-TTT`.

References

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021.

Dhruv Agarwal, Bodhisattwa Prasad Majumder, Reece Adamson, Megha Chakravorty, Satvika Reddy Gavireddy, Aditya Parashar, Harshit Surana, Bhavana Dalvi Mishra, Andrew McCallum, Ashish Sabharwal, et al. Autodiscovery: Open-ended scientific discovery via bayesian surprise. *arXiv preprint arXiv:2507.00310*, 2025.

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. *arXiv preprint arXiv:2411.07279*, 2024.

Oleksandr Balabanov and Hampus Linander. Uncertainty quantification in fine-tuned llms using lora ensembles. *arXiv preprint arXiv:2402.12264*, 2024.

João Caldeira and Brian Nord. Deeply uncertain: comparing methods of uncertainty quantification in deep learning algorithms. *Machine Learning: Science and Technology*, 2(1):015002, 2021.

Xiaofeng Cao and Ivor W Tsang. Bayesian active learning by disagreements: A geometric perspective. *arXiv preprint arXiv:2105.02543*, 2021.

Bertrand Charpentier, Ransalu Senanayake, Mykel Kochenderfer, and Stephan Günnemann. Disentangling epistemic and aleatoric uncertainty in reinforcement learning. *arXiv preprint arXiv:2206.01558*, 2022.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 30377–30385, 2026.

Runpeng Dai, Linfeng Song, Haolin Liu, Zhenwen Liang, Dian Yu, Haitao Mi, Zhaopeng Tu, Rui Liu, Tong Zheng, Hongtu Zhu, et al. Cde: Curiosity-driven exploration for efficient reinforcement learning in large language models. *arXiv preprint arXiv:2509.09675*, 2025.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning*, pages 8657–8677. PMLR, 2023.

Vikranth Dwaracherla, Seyed Mohammad Asghari, Botao Hao, and Benjamin Van Roy. Efficient exploration for llms. *arXiv preprint arXiv:2402.00396*, 2024.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, et al. Lm-polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 446–461, 2023.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Jingtong Gao, Ling Pan, Yejing Wang, Rui Zhong, Chi Lu, Maolin Wang, Qingpeng Cai, Peng Jiang, and Xiangyu Zhao. Navigate the unknown: Enhancing llm reasoning with intrinsic motivation guided exploration. *arXiv preprint arXiv:2505.17621*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. *arXiv preprint arXiv:2305.18466*, 2023.

Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. Teaching large language models to reason with reinforcement learning. *arXiv preprint arXiv:2403.04642*, 2024.

Jianfeng He, Linlin Yu, Changbin Li, Runing Yang, Fanglan Chen, Kangshuo Li, Min Zhang, Shuo Lei, Xuchao Zhang, Mohammad Beigi, et al. Survey of uncertainty estimation in llms-sources, methods, applications, and challenges. *Information Fusion*, page 104057, 2025a.

Kaifeng He, Mingwei Liu, Chong Wang, Zike Li, Yanlin Wang, Xin Peng, and Zibin Zheng. Adadec: Uncertainty-guided adaptive decoding for llm-based code generation. *arXiv e-prints*, pages arXiv–2506, 2025b. Lookahead reranking at high-entropy tokens.

Haitao Hong, Yuchen Yan, Xingyu Wu, Guiyang Hou, Wenqi Zhang, Weiming Lu, Yongliang Shen, and Jun Xiao. Cooper: Co-optimizing policy and reward models in reinforcement learning for large language models. *arXiv preprint arXiv:2508.05613*, 2025.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Jinwu Hu, Zhitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-time learning for large language models. *arXiv preprint arXiv:2505.20633*, 2025.

Yuheng Huang, Jiayang Song, Zhijie Wang, Shengming Zhao, Huaming Chen, Felix Juefei-Xu, and Lei Ma. Look before you leap: An exploratory study of uncertainty analysis for large language models. *IEEE Transactions on Software Engineering*, 51(2):413–429, 2025.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

Hadi Khalaf, Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio du Pin Calmon. Inference-time reward hacking in large language models. *arXiv preprint arXiv:2506.19248*, 2025.

Guillaume Levy, Cédric Colas, Pierre-Yves Oudeyer, Thomas Carta, and Clément Romac. Worldllm: Improving llms' world modeling using curiosity-driven theory-making. *arXiv preprint arXiv:2506.06725*, 2025.

Ang Li, Zhihang Yuan, Yang Zhang, Shouda Liu, and Yisen Wang. Know when to explore: Difficulty-aware certainty as a guide for llm reinforcement learning. *arXiv preprint arXiv:2509.00125*, 2025.

Mengqi Liao, Xiangyu Xi, Chen Ruinian, Jia Leng, Yangen Hu, Ke Zeng, Shuai Liu, and Huaiyu Wan. Enhancing efficiency and exploration in reinforcement learning for llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1451–1463, 2025.

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6107–6117, 2025.

Mohammad Mahdi Moradi, Hossam Amer, Sudhir Mudur, Weiwei Zhang, Yang Liu, and Walid Ahmed. Continuous self-improvement of large language models by test-time training with verifier-driven sample selection. *arXiv preprint arXiv:2505.19475*, 2025.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Biqing Qi, Kaiyan Zhang, Haoxiang Li, Kai Tian, Sihang Zeng, Zhang-Ren Chen, and Bowen Zhou. Large language models are zero shot hypothesis proposers. *arXiv preprint arXiv:2311.05965*, 2023.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sushil Sikchi, Joey Hejna, Brad Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. *Advances in Neural Information Processing Systems*, 37:126207–126242, 2024.

Bojana Ranković, Ryan-Rhys Griffiths, and Philippe Schwaller. Large language models as uncertainty-calibrated optimizers for experimental discovery. *arXiv preprint arXiv:2504.06265*, 2025.

Christopher D Rosin. Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence*, 61(3):203–230, 2011.

Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z Ren, and Anirudha Majumdar. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *ACM Computing Surveys*, 58(3):1–38, 2025.

Linwei Tao, Yi-Fan Yeh, Minjing Dong, Tao Huang, Philip Torr, and Chang Xu. Revisiting uncertainty estimation and calibration of large language models. *arXiv preprint arXiv:2505.23854*, 2025.

Morgan Thomas, Albert Bou, and Gianni De Fabritiis. Test-time training scaling laws for chemical exploration in drug design. *Journal of Chemical Information and Modeling*, 65(24):13178–13186, 2025a.

Morgan Thomas, Albert Bou, and Gianni De Fabritiis. Test-time training scaling laws for chemical exploration in drug design. *Journal of Chemical Information and Modeling*, 65(24):13178–13186, 2025b.

Christos Thrampoulidis, Sadegh Mahdavi, and Wenlong Deng. Advantage shaping as surrogate reward maximization: Unifying pass@ k policy gradients. *arXiv preprint arXiv:2510.23049*, 2025.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, et al. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248, 2025.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

Xi Wang, Laurence Aitchison, and Maja Rudolph. Lora ensembles for large language model fine-tuning. *arXiv preprint arXiv:2310.00035*, 2023.

Zhiqiu Xia, Jinxuan Xu, Yuqian Zhang, and Hang Liu. A survey of uncertainty estimation methods on large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21381–21396, 2025.

Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvári. To believe or not to believe your llm. *arXiv preprint arXiv:2406.02543*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhengmao Ye, Dengchun Li, Zetao Hu, Tingfeng Lan, Jian Sha, Sicong Zhang, Lei Duan, Jie Zuo, Hui Lu, Yuanchun Zhou, and Mingjie Tang. mLoRA: Fine-tuning LoRA adapters via highly-efficient pipeline parallelism in multiple GPUs. *arXiv:2312.02515*, 2024.

Mingqi Yuan, Man-On Pun, and Dong Wang. Rényi state entr opy maximization for exploration acceleration in reinforcement learning. *IEEE transactions on artificial intelligence*, 4(5):1154–1164, 2022.

Mert Yuksekgonul, Daniel Koceja, Xinhao Li, Federico Bianchi, Jed McCaleb, Xiaolong Wang, Jan Kautz, Yejin Choi, James Zou, Carlos Guestrin, et al. Learning to discover at test time. *arXiv preprint arXiv:2601.16175*, 2026a.

Mert Yuksekgonul, Daniel Koceja, Xinhao Li, Federico Bianchi, Jed McCaleb, Xiaolong Wang, Jan Kautz, Yejin Choi, James Zou, Carlos Guestrin, et al. Learning to discover at test time. *arXiv preprint arXiv:2601.16175*, 2026b.

Yuanzhao Zhai, Han Zhang, Yu Lei, Yue Yu, Kele Xu, Dawei Feng, Bo Ding, and Huaimin Wang. Uncertainty-penalized reinforcement learning from human feedback with diverse reward lora ensembles. *arXiv preprint arXiv:2401.00243*, 2023.

Yuanzhao Zhai, Yu Lei, Han Zhang, Yue Yu, Kele Xu, Dawei Feng, Bo Ding, and Huaimin Wang. Uncertainty-penalized reinforcement learning from human feedback with diversified reward lora ensembles. *Information Processing & Management*, 63(3):104548, 2026.

Appendix

A Baseline objective: TTT-DISCOVER in full notation

We collect here, for reference, the TTT-DISCOVER training signal that UG-TTT builds on. TTT-DISCOVER optimises a state-conditional entropic objective with an adaptive temperature $\beta(s)$,

$$J_\beta(\theta) \;=\; \mathbb{E}_{s\sim\mathrm{reuse}(\mathcal{H})}\!\left[\frac{1}{\beta(s)}\log \mathbb{E}_{a\sim\pi_\theta(\cdot|q,s)}\!\left[e^{\beta(s)\,R_{\mathrm{exec}}(s,a)}\right]\right], \tag{9}$$

which interpolates between expected reward ($\beta\to 0$) and the per-state maximum ($\beta\to\infty$). The outer expectation is over parent states s drawn from the reuse distribution over the PUCT history $\mathcal{H}$, and the inner expectation is over rollouts a sampled from the policy at prompt q and parent s. For every group, $\beta(s)$ is selected as the unique $\beta>0$ such that the softmax $q_\beta(a)\propto\exp(\beta R_{\mathrm{exec}}(a))$ has a fixed $\ln 2$ KL budget against uniform,

$$\mathrm{KL}\big(q_\beta \,\big\|\, \mathrm{Unif}\big) \;=\; \ln 2, \tag{10}$$

which exists and is unique whenever the within-group rewards are not all identical, since $\mathrm{KL}(q_\beta\|\mathrm{Unif})$ is continuous and strictly increasing in β on $(0,\infty)$ with limits 0 and $\ln G$. We solve Eq. (10) by 60 iterations of bisection on $[0,10^6]$; on the degenerate equal-reward set, the bisection saturates at the upper bracket and we fall back to $A_i\equiv 0$. Given $\beta(s)$, rollout $i\in\{1,\dots,G\}$ receives the leave-one-out entropic weight and advantage

$$w_i \;=\; \frac{\exp\big(\beta\, R^{(i)}_{\mathrm{exec}}\big)}{\frac{1}{G-1}\sum_{j\neq i}\exp\big(\beta\, R^{(j)}_{\mathrm{exec}}\big)}, \qquad A_i \;=\; w_i - 1, \tag{11}$$

which centres the advantage within each group without a learned value function. The rollouts are trained with the clipped importance-sampling loss

$$\mathcal{L}_{\mathrm{PG}}(\theta) \;=\; -\sum_{t=1}^{|o|}\min\Big[\,\rho_t(\theta)\,A_i,\;\mathrm{clip}\big(\rho_t(\theta),\,1{-}\epsilon,\,1{+}\epsilon\big)\,A_i\Big], \tag{12}$$

with likelihood ratio $\rho_t(\theta) = \pi_\theta(o_t \mid q, o_{<t})/\pi_{\theta_{\mathrm{old}}}(o_t \mid q, o_{<t})$, reducing to plain importance sampling when ϵ is disabled. A KL-to-base penalty $\lambda_{\mathrm{KL}}\,\mathrm{KL}(\pi_\theta\|\pi_{\mathrm{base}})$ is added when $\lambda_{\mathrm{KL}}>0$.

The per-adapter form used inside UG-TTT replaces A_i with A_i^{shaped} (Eq. (6)):

$$\mathcal{L}^{(k)}_{\mathrm{PG}}(\theta_k) \;=\; -\sum_{t=1}^{|o|}\min\Big[\,\rho^{(k)}_t(\theta_k)\,A_i^{\mathrm{shaped}},\;\mathrm{clip}\big(\rho^{(k)}_t,\,1{-}\epsilon,\,1{+}\epsilon\big)\,A_i^{\mathrm{shaped}}\Big], \tag{13}$$

with per-adapter ratio $\rho^{(k)}_t(\theta_k) = \pi_{\theta_k}(o_t \mid q, o_{<t})/\pi_{\theta_{k,\mathrm{old}}}(o_t \mid q, o_{<t})$.

Token-wise KL adjustment. When $\lambda_{\mathrm{KL}}>0$, the scalar A_i^{shaped} is broadcast to a per-token tensor and the token-wise correction of Yuksekgonul et al. [2026b] is applied before the clipped-IS reduction. Concretely, for each position t the advantage entry is adjusted by $-\lambda_{\mathrm{KL}}\big(\log\pi_\theta(o_t \mid q, o_{<t}) - \log\pi_{\mathrm{base}}(o_t \mid q, o_{<t})\big)$, which under sampling from π_θ gives the standard k1 single-sample stochastic estimator of $\nabla_\theta\,\lambda_{\mathrm{KL}}\mathrm{KL}(\pi_\theta\|\pi_{\mathrm{base}})$ folded into the per-token PG target—no second autograd pass through the policy is needed. The transcription is identical to the reference implementation and is stated here only to fix notation used in the proof of Proposition 1.

B Full training algorithm

Algorithm 1 UG-TTT training step (one group of G rollouts).

Require: Ensemble $\{\theta_k\}_{k=1}^{K}$; parent state s from PUCT buffer; group size G.
Require: Coefficients α, β_{ref}, $\gamma_{\max}$, λ_{KL}, λ_{NNM}.
Ensure: Updated ensemble $\{\theta_k\}_{k=1}^{K}$.

// Stage 1: Rollout generation (round-robin over adapters)
Build prompt q from (d, s).
for $i = 1, \ldots, G$ **do**
  $k_i \leftarrow \big((i-1) \bmod K\big) + 1$ ▷ 1-indexed; $K \mid G$ so the assignment pattern $(1, 2, \ldots, K, 1, 2, \ldots)$ repeats identically every group
  $o_i \sim \pi_{\theta_{k_i}}(\cdot \mid q, s)$
  $R_{\text{exec}}^{(i)} \leftarrow \textsc{ProgramChecker}(s, o_i)$
end for
// Stage 2: Mutual-information scoring (single GPU pass)
$\{p_{\theta_k}\}_{k=1}^{K} \leftarrow \textsc{ChunkedScoreAllAdapters}(o_{1:G})$
for $i = 1, \ldots, G$ **do**
  for $t = 1, \ldots, |o_i|$ **do**
    $\bar{p}_t \leftarrow \frac{1}{K} \sum_{k=1}^{K} p_{\theta_k}(\cdot \mid q, o_{<t}^{(i)})$
    $\text{MI}_t^{(i)} \leftarrow H(\bar{p}_t) - \frac{1}{K} \sum_{k=1}^{K} H\big(p_{\theta_k}(\cdot \mid q, o_{<t}^{(i)})\big)$ ▷ Eq. (2)
  end for
  $U_i \leftarrow \text{top-7\%-mean}_t \, \text{MI}_t^{(i)}$ ▷ Eq. (3)
end for
// Stage 3: Adaptive temperature and LOO advantage (on R_{exec} only)
$\beta(s) \leftarrow \textsc{BisectionSolve}\big\{\beta > 0 : \text{KL}(q_\beta \,\|\, \text{Unif}) = \ln 2\big\}$ on $\{R_{\text{exec}}^{(i)}\}$ ▷ Eq. (10)
for $i = 1, \ldots, G$ **do**
  $w_i \leftarrow \exp(\beta R_{\text{exec}}^{(i)}) \big/ \frac{1}{G-1} \sum_{j \neq i} \exp(\beta R_{\text{exec}}^{(j)})$
  $A_i \leftarrow w_i - 1$ ▷ Eq. (11)
end for
// Stage 4: MI-shaped advantage
$\overline{U} \leftarrow \frac{1}{G} \sum_i U_i$; $\sigma_U \leftarrow \text{std}_{\text{unbiased}}(U)$ (cf. Eq. (4)); $\overline{|A|} \leftarrow \frac{1}{G} \sum_j |A_j|$
$\gamma_{\text{eff}}(s) \leftarrow \alpha \cdot \min\big(\beta(s)/\beta_{\text{ref}},\ \gamma_{\max}\big)$ ▷ Eq. (5)
for $i = 1, \ldots, G$ **do**
  $\tilde{U}_i \leftarrow \text{clip}\big((U_i - \overline{U})/\sigma_U,\ -3,\ 3\big)$ ▷ Eq. (4)
  $A_i^{\text{shaped}} \leftarrow A_i + \gamma_{\text{eff}}(s) \cdot \overline{|A|} \cdot \tilde{U}_i$ ▷ Eq. (6)
end for
if $\lambda_{\text{KL}} > 0$ **then**
  Broadcast A_i^{shaped} per-token; add token-wise KL correction against π_{base} (Appendix A)
end if
// Stage 5: Per-adapter backward, NNM backward, then optimiser step
for $k = 1, \ldots, K$ **do**
  Compute $\frac{1}{K} \mathcal{L}_{\text{PG}}^{(k)}(\theta_k)$ over all G rollouts and call `backward()`; gradients accumulate into $\theta_k.\text{grad}$ ▷ Eq. (13)
  Release adapter k's computation graph before advancing
end for
// KL-to-base contribution is already inside $\mathcal{L}_{\text{PG}}^{(k)}$ via the Stage 4 token-wise correction (App. A)
Compute $\lambda_{\text{NNM}} \mathcal{L}_{\text{NNM}}$ and call `backward()`; gradients accumulate into all $\{\theta_k\}$ ▷ Eq. (7)
for $k = 1, \ldots, K$ **do**
  AdamW step on θ_k using its accumulated gradient; zero $\theta_k.\text{grad}$
end for

C Proofs

C.1 Proof of Proposition 1 (bounded scale perturbation of the shaped advantage)

Fix a parent state s and its group of G rollouts. If $\sigma_U = 0$, Eq. (4) sets $\tilde{U}_i \equiv 0$ by convention, and $A_i^{\text{shaped}} = A_i$ identically. Otherwise

$$\sum_{i=1}^{G} \tilde{U}_i \;=\; \sum_{i=1}^{G} \operatorname{clip}\Big(\frac{U_i - \overline{U}}{\sigma_U},\, -3,\, 3\Big)\,.$$

Let $z_i := (U_i - \overline{U})/\sigma_U$ and $\mathcal{C} = \{i : |z_i| > 3\}$. By centring, $\sum_{i=1}^{G} z_i = 0$, so

$$\sum_{i=1}^{G} \tilde{U}_i \;=\; \sum_{i \notin \mathcal{C}} z_i \;+\; \sum_{i \in \mathcal{C}} \operatorname{sign}(z_i)\cdot 3 \;=\; -\sum_{i \in \mathcal{C}} z_i \;+\; \sum_{i \in \mathcal{C}} \operatorname{sign}(z_i)\cdot 3 \;=\; \sum_{i \in \mathcal{C}} \big(\operatorname{sign}(z_i)\cdot 3 - z_i\big).$$

For $i \in \mathcal{C}$ we have $|z_i| > 3$, and a direct case split on $\operatorname{sign}(z_i)$ gives $|\operatorname{sign}(z_i)\cdot 3 - z_i| = |z_i| - 3$, so by the triangle inequality

$$\Big|\sum_{i=1}^{G} \tilde{U}_i\Big| \;\le\; \sum_{i \in \mathcal{C}} \big(|z_i| - 3\big).$$

By Cauchy–Schwarz, $\sum_{i\in\mathcal{C}} |z_i| \le \sqrt{|\mathcal{C}|\,\sum_i z_i^2} = \sqrt{|\mathcal{C}|(G-1)}$ (using σ_U unbiased, so $\sum_i z_i^2 = G-1$); a looser but more interpretable bound uses $|z_i| \le \sqrt{G-1}$ to give $\sum_{i\in\mathcal{C}} \big(|z_i| - 3\big) \le |\mathcal{C}|(\sqrt{G-1} - 3)$. When $\mathcal{C} = \varnothing$ both sides are exactly 0; note in particular that $|z_i| \le \sqrt{G-1}$ forces $\mathcal{C} = \varnothing$ whenever $G \le 10$, so the looser bound is non-vacuous only in the regime $G \ge 11$ (where $\sqrt{G-1} - 3 > 0$). Substituting into Eq. (6) and using that neither $\gamma_{\text{eff}}(s)$ nor $\overline{|A|}$ depends on i,

$$\Big|\sum_{i=1}^{G} A_i^{\text{shaped}} - \sum_{i=1}^{G} A_i\Big| \;=\; \gamma_{\text{eff}}(s)\,\overline{|A|}\,\Big|\sum_{i=1}^{G} \tilde{U}_i\Big| \;\le\; \gamma_{\text{eff}}(s)\,\overline{|A|}\,|\mathcal{C}|\big(\sqrt{G-1} - 3\big),$$

which is the statement of the proposition.

We emphasise that this is strictly weaker than unbiasedness of the policy-gradient estimator $\sum_i A_i \nabla \log \pi(o_i)$ in the standard sense [Thrampoulidis et al., 2025]: because $\tilde{U}_i$ depends on rollout i through U_i, the substitution $A_i \mapsto A_i^{\text{shaped}}$ does not preserve the expected gradient. The bonus is deliberately biased; the directional bias toward high-MI rollouts is the exploration mechanism. Proposition 1 isolates the bias to the gradient direction by bounding the residual perturbation to the aggregate within-group magnitude. □

C.2 Proof of Proposition 2 (the global maximum is subspace-orthogonal)

Fix a layer ℓ and write $A_k := A_\ell^{(k)} \in \mathbb{R}^{r \times d_{\text{in}}}$. The problem is

$$\max_{A_1,\dots,A_K} \big\|W\big\|_*, \qquad W = \big[A_1; \dots; A_K\big] \in \mathbb{R}^{Kr \times d_{\text{in}}}, \qquad \|A_k\|_F = c\;\forall k.$$

Let $\sigma_1, \dots, \sigma_m$ denote the nonzero singular values of W, where $m = \operatorname{rank}(W) \le \min(Kr, d_{\text{in}}) = Kr$ (using the hypothesis $Kr \le d_{\text{in}}$). By Cauchy–Schwarz applied to $(\sigma_1, \dots, \sigma_m)$ and the all-ones vector,

$$\|W\|_*^2 \;=\; \Big(\sum_{i=1}^{m} \sigma_i\Big)^2 \;\le\; m \sum_{i=1}^{m} \sigma_i^2 \;\le\; Kr \cdot \|W\|_F^2 \;=\; Kr \sum_{k=1}^{K} c^2 \;=\; K^2 r\, c^2,$$

with equality in both inequalities iff $m = Kr$ (i.e. W has full row rank, which requires and is feasible exactly when $Kr \le d_{\text{in}}$, the standing hypothesis) and all Kr singular values are equal. Let s denote this common singular value; then $\|W\|_F^2 = Kr\, s^2 = Kc^2$ gives $s = c/\sqrt{r}$. Under the full-row-rank hypothesis just established, equal singular values are equivalent to $WW^\top = s^2 I_{Kr}$, i.e. the rows of W are mutually orthogonal with common norm s. Restricted to a single block A_k, this forces A_k to have r equal singular values all equal to $s = c/\sqrt{r}$. Across blocks, mutual row orthogonality of W forces the row spaces of $A_1, \dots, A_K$ to be mutually orthogonal subspaces of $\mathbb{R}^{d_{\text{in}}}$; this is feasible iff $Kr \le d_{\text{in}}$, which holds by hypothesis. The global maximum of $\|W\|_*$ on the constraint set therefore satisfies both conditions, as claimed. □

C.3 Proof of Proposition 3 (TTT-DISCOVER as a special case)

Under either branch the ensemble's mixture $\bar{p}_t$ in Eq. (2) coincides with each member's distribution: in (a) trivially because $K = 1$, and in (b) because identical weights yield identical p_{θ_k}. Each term of the decomposition $H(\bar{p}_t) - \frac{1}{K}\sum_k H(p_{\theta_k})$ then equals the same quantity, so $\mathrm{MI}_t \equiv 0$. Consequently $U_i \equiv 0$, $\overline{U} = 0$, $\sigma_U = 0$, and by the convention stated below Eq. (4) we set $\tilde{U}_i \equiv 0$. The shaped advantage reduces to $A_i^{\mathrm{shaped}} = A_i$ and Eq. (13) reduces to Eq. (12). The hypothesis $\lambda_{\mathrm{NNM}} = 0$ removes $\mathcal{L}_{\mathrm{NNM}}$ from Eq. (8) outright (this assumption is required even in branch (a), since for $K = 1$ the regulariser $-\|A_\ell^{(1)}\|_*$ is a non-trivial function of θ_1 that would otherwise perturb the policy update). What remains of Eq. (8) is $\mathcal{L}_{\mathrm{PG}}(\theta_1) + \lambda_{\mathrm{KL}}\mathrm{KL}(\pi_\theta \| \pi_{\mathrm{base}})$, which is the TTT-DISCOVER objective of Yuksekgonul et al. [2026b]. In branch (b), identical-init plus shared-data training (Sec. 2.5) keeps the K adapters tied at every step, so the per-adapter update is byte-identical to baseline's single-adapter update. □

D Implementation notes

Chunked MI and base-logprob scoring. Two distinct logit tensors arise during scoring. (i) The *ensemble MI* path materialises an (K, chunk, V) fp32 logit tensor inside the chunked LM head; for Qwen3-8B with $K = 5$, $V = 152{,}064$, and $\mathrm{chunk} = 256$ this peaks at ≈ 0.78 GB per chunk. The MI decomposition of Eq. (2) is summed into a running per-rollout buffer that reuses the same chunk's $\mathrm{logsoftmax}$ (already computed for target-token gather), so the MI contribution costs $< 0.1\%$ of the LM-head matmul. (ii) The *base-model logprob* path needed for the per-token KL-to-base correction (App. A) materialises a $(1, T, V)$ fp32 logit tensor; for $T = 26$k this would peak at ≈ 16 GB without chunking, and we reduce it to ≈ 8.1 GB by streaming $\mathrm{logsoftmax}$ over the same $\mathrm{chunk} = 256$ granularity and releasing each chunk's allocation before the next is loaded.

Top-7% threshold. Wang et al. [2025] shows that the RL signal in LLMs should concentrate on the high-entropy minority of tokens, the pivotal decision points, rather than be diluted across boilerplate. The top-7% choice is motivated in §2.2; it adapts to sequence length so an n-token response contributes $\lceil 0.07n \rceil$ positions, which remains well-calibrated across the 7k–15k-token range typical of reasoning outputs.

Sequential per-adapter backward. A single joint K-adapter forward–backward retains K parallel computation graphs and scales as $\mathcal{O}(K \cdot T \cdot L \cdot D)$ activation memory, exceeding 80 GB on our setup. We therefore iterate over adapters in Stage 5 of Algorithm 1 (the mLoRA scheme cited in §2.5): each adapter's forward and backward are computed in turn, the computation graph is freed before the next adapter's forward, and gradients are accumulated into a shared buffer for the final AdamW step.

Restricting regularisation to $A^{(k)}$. In pilot runs, regularising both $A^{(k)}$ and $B^{(k)}$ produced oscillatory optimisation: a penalty applied to $B^{(k)}$ was absorbed by a compensating rescaling of $A^{(k)}$, and vice versa, giving no net increase in subspace diversity while destabilising the RL loss. This matches the empirical observation of Zhai et al. [2023]. Leaving $B^{(k)}$ free removes the degeneracy.

E Hyperparameters

Tables 3 and 4 list every hyperparameter used in all reported runs. Values are identical across problems unless noted; per-problem deviations are listed in the notes column.

Table 3: **Shared hyperparameters** (identical for TTT-DISCOVER and UG-TTT on all four problems).

Hyperparameter	Value	Notes
Model		
Base model	Qwen3-8B	fp16 precision
LoRA target modules	q, k, v, o projections	all attention projections
LoRA rank r	16	
LoRA alpha	32	effective scale $= \alpha/r = 2$
LoRA dropout	0.05	
Optimiser		
Optimiser	AdamW	
Learning rate	4×10^{-5}	
RL training		
Training epochs	6	
Group size G	8	rollouts per PUCT node
Groups per batch	8	gradient accumulation
Rollout temperature	1.0	
Advantage estimator	entropic adaptive-β	Eq. (9)
β_{ref}	2.0	bisection target for KL $= \ln 2$
IS clip ϵ	0.2	
KL penalty λ_{KL}	0.01	
Remove constant-reward groups	yes	
Sampling and context		
Sampler	PUCT backprop	
Context window	32 768 tokens	Qwen3-8B native
Phase-1 (thinking) token cap	26 000	hard cap on thinking tokens
Phase-2 wrap budget	6 000 tokens	code phase; streaming runs only

Table 4: **UG-TTT-specific hyperparameters.** The baseline sets K=1, α=0, λ_{NNM}=0, and disables streaming MI.

Hyperparameter	Value	Notes
Ensemble		
Number of adapters K	5	round-robin rollout assignment
MI exploration bonus (Sec. 2.3)		
MI metric	true MI, top-7% tokens	Eq. (2), Eq. (3)
MI coefficient α	0.1	base exploration strength
$\gamma_{\max}$ ratio	10.0	clips $\gamma_{\text{eff}}/\alpha$
MI clip range	$[-3, 3]$	on standardised $\tilde{U}_i$
Nuclear-norm regularisation (Sec. 2.4)		
NNM coefficient λ_{NNM}	0.075	applied to $A^{(k)}$ matrices only
Frobenius-normalised variant	no	raw nuclear norm used
Streaming MI early-stop (Appendix F)		
Enabled	AC1, CP26	disabled on AC2 and Erdős
Window size W	4 096 tokens	
Check interval C	2 048 tokens	
Min generation before check	4 096 tokens	
MI threshold percentile	25th	of historical window MI
Warmup epochs	3	no streaming before epoch 3

F Streaming MI early-stop and length-reward diagnostic

Mechanism as logged. The streaming early-stop is implemented as a hard cap on the thinking phase: when a rollout reaches 4096 thinking tokens (occasionally 6144 on Erdős where the cap was widened), the windowed ensemble MI is compared to the running 25^{th} percentile of historical MI; if the rollout is below threshold, the thinking phase is closed and the rollout is forced into a bounded code phase. The MI check is evaluated once, at the cap: rollouts below the historical 25^{th} percentile are truncated to a bounded code phase, while rollouts above threshold continue past the cap to natural termination. The recorded `streaming_mi_stop_step` therefore coincides with the cap (it is the step at which the gate is queried), but the gate itself is the MI threshold, not the cap; the $50.0\%/55.6\%$ firing rates on AC1 and CP26 are the rates at which the MI check chose to truncate, not a fixed length cut. Non-streamed rollouts within the same run either emitted `</think>` before the cap or were above threshold at the cap and continued to natural termination.

Per-problem firing and savings.

Problem	Firing rate	Mean tokens saved per rollout	Mean tokens saved per firing	Streaming setting in Table 1
AC1	50.0%	569	1,138	enabled
CP26	55.6%	757	1,363	enabled
AC2	disabled	—	—	disabled
Erdős	disabled	—	—	disabled

Paired AC2/Erdős comparison (epochs 3–5). We forked AC2 from a shared epoch-2 checkpoint and ran the remaining three epochs with and without streaming; on Erdős we ran the two configurations as separate full runs and aligned epochs 3–5 for the comparison.

	$R_{\max}$ (stream)	$R_{\max}$ (no stream)	Mean rollout length (stream)	Mean rollout length (no stream)
AC2	0.8415	**0.8563**	5,312	8,711
Erdős	2.6150	**2.6167**	5,255	8,720

Length–reward diagnostic. We test whether the within-run rank correlation between rollout length and execution reward separates the four problems. Restricting to correct rollouts and parsing the `</think>` marker to split thinking from code tokens (streamed rollouts use the logged `phase1_tokens` / `phase2_tokens` fields directly), we report Spearman ρ for epochs 0–2 and for all epochs:

	Epochs 0–2			All epochs		
Problem	ρ_{think}	ρ_{code}	ρ_{total}	ρ_{think}	ρ_{code}	ρ_{total}
AC1	-0.30	$+0.52$	-0.19	-0.51	$+0.48$	-0.43
CP26	$+0.16$	-0.10	$+0.15$	-0.34	-0.14	-0.41
AC2	-0.28	$+0.39$	-0.21	-0.18	$+0.36$	-0.11
Erdős	-0.18	$+0.12$	-0.16	-0.11	$+0.07$	-0.10

The thinking-phase correlation is not significantly positive on any problem and is significantly negative on AC1; the code-phase correlation is positive on AC1 and AC2, weakly positive on Erdős, and null on CP26. No single test on length predicts whether streaming will help or hurt $R_{\max}$ on a given problem; the per-problem calibration in this paper is therefore empirical (paired pilot on AC2/Erdős) rather than diagnosable in advance from a length–reward test.

What streaming-truncated rollouts look like in practice. On AC1 and CP26 the rollouts that achieve the Table 1 $R_{\max}$ are themselves streaming-truncated: AC1 $R_{\max}$=0.6406 from a rollout of length $5{,}116$ (`streaming_mi_stopped`=1); CP26 $R_{\max}$=2.6359 from a rollout of length $5{,}318$ (also truncated). On AC2 and Erdős the Table 1 entries are produced from no-streaming runs (lengths $5{,}642$ and $7{,}316$ respectively). We report this asymmetry directly because it is recoverable from the trajectory logs.

Table 5: **Method-exclusive discoveries (idea emergence).** For each problem, the algorithmic family used by the highest-reward rollout, its adoption among the 30 best rollouts, and its adoption among all correct rollouts during training. Top-30: number of the 30 highest-reward rollouts that contain the motif. Adoption: motif uses divided by total correct rollouts. The baseline never crosses 1/30 on any problem.

		Top-30 (count)		Adoption (%)	
Problem	Winning template	Baseline	UG-TTT	Baseline	UG-TTT
AC1	`target_sum` $= \sqrt{2n}$ (Cauchy–Schwarz)	1/30	**18**/30	3.0	**28.0**
AC2	`scipy diff_evo` $\cup$ gradient-based	0/30	**13**/30	0.9	**21.6**
CP26	non-uniform rows, e.g. `[5,5,5,5,6]`	1/30	**13**/30	17.6	**22.4**
Erdős	Fourier init $h(x) = a + b\sin(kx)$	0/30	**6**/30	0	**6.9**

G Method-exclusive discoveries

Preserving entropy is only useful if the surviving families produce *better* programs. The surviving families translate into winning motifs the baseline reaches at most 1/30 times on any problem (Table 5); two are mathematically substantive—the Cauchy–Schwarz target $\sqrt{2n}$ on AC1 (a closed-form symmetry bounding C_1 from below) and the Fourier ansatz $h(x) = a + b\sin(kx)$ on Erdős—and both lie inside families the baseline extinguishes by the final epoch (Fig. 3). Preserved ensemble diversity is the precondition for entering them at all.

H Broader impact

UG-TTT reduces the compute required for LLM-driven scientific discovery, reaching competitive discovery ceilings in 6 training epochs where TTT-DISCOVER requires 50, on an identical Qwen3-8B base model on a single GPU. The mechanism is dual use under any task admitting a cheap programmatic verifier, since rewarding verifier-uncertain regions under a loosely specified verifier (for instance, automated exploit search) would actively seek loopholes. Two mitigations follow directly from the method. First, the verifier should be audited prior to scaling compute. Second, ensemble mutual information should be monitored as a leading indicator of stalled exploration.

I Qualitative case study on AC2

The $+0.0031$ flip on AC2 (Tab. 1) and the 0/30 versus **13**/30 adoption gap on the `scipy.optimize.differential_evolution` family (Tab. 5) admit a concrete reading at the program level. We compare the highest-reward correct rollout produced by TTT-DISCOVER and by UG-TTT over identical training budgets on the autocorrelation lower-bound problem. Both panels report the verbatim algorithmic core of the model-emitted Python program, abridged where boilerplate (clipping, time guards, fallback rebuilds of the initial guess set) is repeated across rollouts.

Problem statement (AC2)

Construct a sequence $h \in \mathbb{R}^{1000}_{\geq 0}$ of step-function heights whose autocorrelation lower bound C_2 (returned by the closed-form checker `evaluate_sequence`) is as large as possible, subject to $\sum_i h_i \geq 0.01$ and $h_i \in [0, 1000]$. Reward equals C_2 truncated at correctness; a ground-truth Cauchy–Schwarz construction `height_sequence_1` is exposed as a global hint. Higher is better.

TTT-DISCOVER (best of 384 rollouts, $R_{\max} = 0.8532$, hand-rolled genetic algorithm)

```
POPULATION_SIZE = 200; MAX_GENERATIONS = 500
MUTATION_RATE_INITIAL, MUTATION_RATE_FINAL = 0.25, 0.05

# half: perturb height_sequence_1; half: random uniform over [0, 200]
```

```
population = generate_initial_population()

for gen in range(MAX_GENERATIONS):
    fitness = [evaluate_sequence(s) for s in population]
    if max(fitness) > best_value:
        best_value, best_seq, stamina = max(fitness), argmax_seq, 0
    else:
        stamina += 1
    if stamina > 80:                          # restart on stagnation
        population = generate_initial_population(); stamina = 0

    parents = [tournament_selection(population, fitness, size=5)
               for _ in range(POPULATION_SIZE)]
    new_pop = [parents[0]]                    # elitism
    while len(new_pop) < POPULATION_SIZE:
        p1, p2 = random.sample(parents, 2)
        w = random.uniform(0.3, 0.7)          # blend crossover
        child = [p1[i]*w + p2[i]*(1-w) for i in range(SEQ_LENGTH)]
        rate = MUTATION_RATE_INITIAL - (gen/MAX_GENERATIONS)*(
                  MUTATION_RATE_INITIAL - MUTATION_RATE_FINAL)
        for i in range(SEQ_LENGTH):
            if random.random() < rate:
                child[i] = clip(child[i] + random.gauss(0, 1.5))
        new_pop.append(child)

    best_seq = local_search(best_seq)         # +/- delta sweep on elite
    population = new_pop
```

Algorithmic family. Pure-Python evolutionary loop. Tournament selection of size 5, blend crossover with weight in $[0.3, 0.7]$, Gaussian mutation with rate decaying linearly from 0.25 to 0.05, restart-on-stagnation after 80 generations without improvement, and a coordinate-wise delta sweep as elite refinement. No external optimiser is invoked. Across all 384 baseline rollouts, 0 entered the top-30 via `scipy.optimize.differential_evolution` (Tab. 5); the family is reachable in principle but not in practice under the unregularised ensemble.

(best of 384 rollouts, $R_{\max} = 0.8563$, structured-seed differential evolution)

```
from scipy.optimize import differential_evolution

length = 1000
bounds = [(0.0, 1000.0)] * length

initial_guesses = []
if 'height_sequence_1' in globals():          # exploit the hint
    initial_guesses.append(np.clip(np.array(height_sequence_1), 0, 1000)
        )

for _ in range(50):                           # uniform low-magnitude
    initial_guesses.append(np.clip(np.random.rand(length)*0.1, 0, 1000))
for _ in range(50):                           # gaussian, small mean
    initial_guesses.append(np.clip(np.random.normal(0.03, 0.1, length),
        0, 1000))
for _ in range(50):                           # sorted gaussian (monotone
    prior)
    s = np.sort(np.clip(np.random.normal(0.05, 0.1, length), 0, 1000))
    initial_guesses.append(s)

popsize = 100
initial_guesses = initial_guesses[:popsize]

result = differential_evolution(
    lambda x: -evaluate_sequence(x.tolist()),
    bounds,
    popsize       = popsize,
```

```
    maxiter        = 500,
    mutation       = (0.5, 1.0),      # dithered scale factor
    recombination  = 0.8,
    tol            = 1e-5,
    strategy       = 'best1bin',
    init           = np.array(initial_guesses),
)
return np.clip(result.x, 0, 1000).tolist()
```

Algorithmic family. A single call to `scipy.optimize.differential_evolution` replaces the hand-rolled loop. The contribution sits in the initialisation, namely a structured pool of 151 seeds spanning the closed-form Cauchy–Schwarz hint, three statistically distinct random priors (clamped uniform, low-mean Gaussian, monotone sorted Gaussian), and a population size matched to the seed pool. The search uses dithered mutation in $[0.5, 1.0]$ and the `best1bin` strategy. This program belongs to the `scipy diff_evo` family that the unregularised ensemble emits in 0/30 top rollouts and the UG-TTT ensemble emits in **13**/30 (Tab. 5).

Why UG-TTT reaches the better program

The two programs differ less in their search effort than in the algorithmic family they invoke. The TTT-DISCOVER rollout reimplements the components of differential evolution (population, mutation, recombination, elitism) inside a 130-line Python loop, while the UG-TTT rollout delegates the loop to the `scipy` solver and concentrates novelty in a structured seed pool. Reaching that delegation requires the policy to explore a code region the unregularised ensemble has effectively extinguished; the family-entropy curve on AC2 (Fig. 3b) shows TTT-DISCOVER contracting onto hand-rolled evolutionary loops by epoch 3, whereas UG-TTT retains weight on `scipy` solvers and gradient-based families throughout training. The mutual-information bonus of Sec. 2.3 pays out precisely on the tokens where the policy chooses `from scipy.optimize import differential_evolution` over `def construct_function` with a manual loop, since adapter disagreement is highest on those import-and-API-name positions and lowest inside the body of either choice once committed. The nuclear-norm regulariser of Sec. 2.4 is what keeps that disagreement available on AC2, and removing it collapses adoption to the TTT-DISCOVER pattern (Tab. 2).