
Epistemic Uncertainty for Test-Time Discovery

Anonymous Author(s)

Affiliation

Address

email

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** 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 confirms that the regularizer is essential for sustaining exploration.

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 the knowledge frontier Yuksekgonul 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

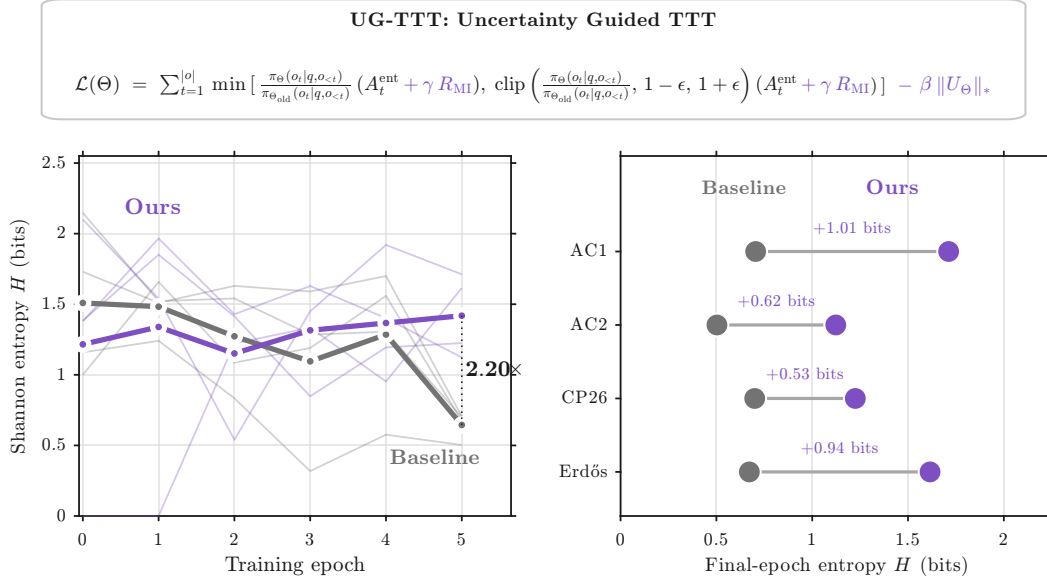


Figure 1: **UG-TTT preserves the diversity that drives discovery.** The training objective (top) augments the TTT-DISCOVER clipped importance-sampling (IS) surrogate with a mutual-information (MI)-shaped advantage and a nuclear-norm regulariser (purple terms). *Left:* Per-epoch Shannon entropy H of the 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.

39 model’s learned distribution primarily performs retrieval or exploitation. However, scientific dis-
 40 covery demands ideas beyond the training data. Therefore, progress hinges on a key question: how
 41 can we push an LLM to navigate its knowledge boundaries, where its initial predictive confidence is
 42 low, but the potential for meaningful epistemic shift is high?

43 **Reinforcement learning as a reward.** Existing test-time training frameworks use Predictor + Upper
 44 Confidence Bound applied to Trees (PUCT) Rosin [2011] to select parent solutions, after which the
 45 policy generates mutations for scoring by a verifier. The failure to improve the maximum reward
 46 is structural, originating in the update rule itself: the standard reinforcement learning objective
 47 maximizes expected reward, which rewards predictive reliability and penalizes variance Cheng et al.
 48 [2026], Li et al. [2025]. At the gradient level, this installs a mechanistic bias toward mutations
 49 with well-estimated outcomes, filtering out uncertain but potentially high-payoff directions. The
 50 consequence is diversity collapse, where the model converges on a narrow set of low-risk mutations,
 51 accumulating exploitative gains in average reward at the expense of the singular breakthrough state
 52 that discovery actually requires Yuksekgonul et al. [2026a], Thomas et al. [2025b].

53 **Why disagreement is the right signal.** Existing fixes for diversity collapse, including entropy
 54 bonuses, confidence-based difficulty weighting, and uncertainty-guided reasoning budgets Cheng
 55 et al. [2026], Li et al. [2025], all derive their signal from a single trained network, which cannot sep-
 56 arate *aleatoric uncertainty* (irreducible input ambiguity) from *epistemic uncertainty* (gaps in what
 57 the model has learned). This is because a single network produces high entropy for both signals;
 58 distinguishing them requires measuring weight-space disagreement, a signal absent in single trajec-
 59 tories Hüllermeier and Waegeman [2021], Balabanov and Linander [2024], Yadkori et al. [2024].
 60 For discovery, this conflation is fatal: the policy spends compute on ambiguous-but-familiar terri-
 61 tory rather than the genuine knowledge frontier where breakthroughs live. AutoDiscovery [Agar-
 62 wal et al., 2025] compounds the problem further, deriving Bayesian surprise from a single frozen
 63 LLM and therefore accumulating no domain knowledge as discoveries proceed. Bayesian theory
 64 resolves the confound: mutual information between predictions and model parameters isolates the
 65 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 Yuksekogonul 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 $\text{MI}_t = H(\bar{p}(\cdot \mid o_{<t})) - \frac{1}{K} \sum_{k=1}^K H(p_k(\cdot \mid o_{<t}))$ 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 $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ 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 $o = (o_1, \dots, o_{|o|})$ 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 $R_{\text{exec}}^{(i)}$, 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 $\mathcal{L}_{\text{PG}}$ 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,

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

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 :

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

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,

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

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 ,

$$\tilde{U}_i = \text{clip}\left(\frac{U_i - \bar{U}}{\sigma_U}, -3, 3\right), \quad \bar{U} = \frac{1}{G} \sum_i U_i, \quad \sigma_U = \sqrt{\frac{1}{G-1} \sum_i (U_i - \bar{U})^2}, \quad (4)$$

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,

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

where α is a base rate, β_{ref} is a reference temperature, and γ_{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

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

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 $\beta \rightarrow \infty$ 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 $z_i = (U_i - \bar{U})/\sigma_U$ and $\mathcal{C} = \{i : |z_i| > 3\}$. Then $|\sum_i A_i^{\text{shaped}} - \sum_i A_i| \leq \gamma_{\text{eff}}(s) \overline{|A|} |\mathcal{C}|(\sqrt{G-1} - 3)$, and the bound vanishes whenever the clip is inactive ($\mathcal{C} = \emptyset$). 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 $\rightarrow 1$), 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

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

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 $k \in \{1, \dots, K\}$. 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 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

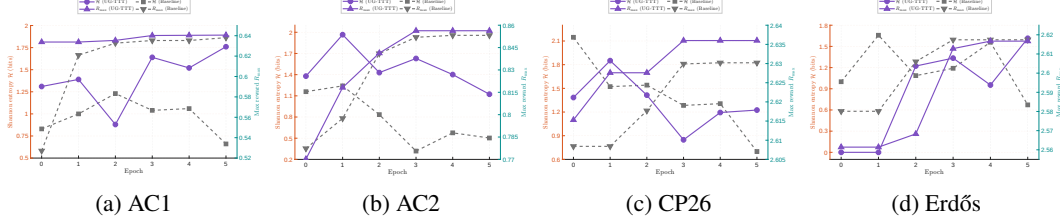


Figure 2: **Training dynamics across all four benchmarks.** Per-epoch Shannon entropy H of the family distribution over correct rollouts (orange, left axis) and cumulative $R_{\max}$ (teal, right axis); dashed is baseline, solid is **UG-TTT**. The baseline contracts H monotonically on every problem while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

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,

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

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, $\lambda_{\text{NNM}}=0.075$, $\alpha=0.1$, 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).

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.

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.

Problem	$R_{\max}$ during training			H at epoch 5 (bits)	
	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).

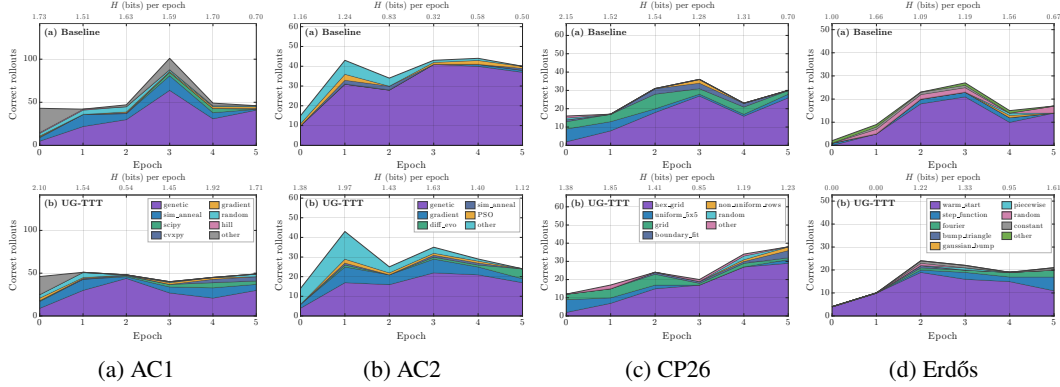


Figure 3: **Diversity collapse is universal and UG-TTT averts it.** Per-epoch family composition of correct rollouts on each benchmark; within each panel, top row is baseline (1 LoRA) and bottom is UG-TTT (5 LoRA + MI + NNM), with $H_{\text{start}} \rightarrow H_{\text{end}}$ reported above. Family vocabularies 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.

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

241 3.2 Diversity is the mechanism

242 Figure 3 tracks the per-epoch family mixture over correct rollouts on every problem. The pattern is
243 uniform across the two family taxonomies (outer search loop on AC1/AC2, structural initialisation
244 on CP26/Erdős). The baseline narrows monotonically to $H \leq 0.71$ bits with one family at 82 to
245 93% of rollouts, whereas UG-TTT either holds a wider portfolio throughout (AC1, CP26) or opens
246 across training (AC2, Erdős) and finishes with ≥ 3 live families on every problem. The collapse
247 is therefore a universal failure mode of TTT-DISCOVER-style RL on discovery, and the diversity
248 regulariser averts it. Mechanistically the β -coupling of Remark 1 (Eq. (5)) drives the effect, since
249 γ_{eff} rises with reward-seeking pressure and sustains exploration on underrepresented families at the
250 moment the baseline contracts. §3.3 confirms the link, as removing NNM drives MI to $\sim 10^{-5}$
251 within two epochs and the family distribution narrows identically to the single-adaptor 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	2.2×10^{-2}	2.33×10^6	2.04×10^6
No NNM (MI bonus collapses to 0)	2.6302	6.6×10^{-6}	2.72×10^6	2.72×10^6

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.

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-adaptor 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.

388 Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu,
389 Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. Teaching
390 large language models to reason with reinforcement learning. *arXiv preprint arXiv:2403.04642*,
391 2024.

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

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

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

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

403 Jinwu Hu, Zhitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li,
404 and Mingkui Tan. Test-time learning for large language models. *arXiv preprint arXiv:2505.20633*,
405 2025.

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

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

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

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

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

420 Mengqi Liao, Xiangyu Xi, Chen Ruinian, Jia Leng, Yangen Hu, Ke Zeng, Shuai Liu, and Huaiyu
421 Wan. Enhancing efficiency and exploration in reinforcement learning for llms. In *Proceedings of*
422 *the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1451–1463,
423 2025.

424 Xiaoou Liu, Tiejun Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. Uncertainty quantifi-
425 cation and confidence calibration in large language models: A survey. In *Proceedings of the 31st*
426 *ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6107–6117,
427 2025.

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

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

435 Binqing Qi, Kaiyan Zhang, Haoxiang Li, Kai Tian, Sihang Zeng, Zhang-Ren Chen, and Bowen Zhou.
436 Large language models are zero shot hypothesis proposers. *arXiv preprint arXiv:2311.05965*,
437 2023.

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

441 Bojana Ranković, Ryan-Rhys Griffiths, and Philippe Schwaller. Large language models as
442 uncertainty-calibrated optimizers for experimental discovery. *arXiv preprint arXiv:2504.06265*,
443 2025.

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

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

449 Linwei Tao, Yi-Fan Yeh, Mingjing Dong, Tao Huang, Philip Torr, and Chang Xu. Revisiting un-
450 certainty estimation and calibration of large language models. *arXiv preprint arXiv:2505.23854*,
451 2025.

452 Morgan Thomas, Albert Bou, and Gianni De Fabritiis. Test-time training scaling laws for chemical
453 exploration in drug design. *Journal of Chemical Information and Modeling*, 65(24):13178–13186,
454 2025a.

455 Morgan Thomas, Albert Bou, and Gianni De Fabritiis. Test-time training scaling laws for chemical
456 exploration in drug design. *Journal of Chemical Information and Modeling*, 65(24):13178–13186,
457 2025b.

458 Christos Thrampoulidis, Sadegh Mahdavi, and Wenlong Deng. Advantage shaping as surrogate
459 reward maximization: Unifying pass@k policy gradients. *arXiv preprint arXiv:2510.23049*,
460 2025.

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

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

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

470 Zhiqiu Xia, Jinxuan Xu, Yuqian Zhang, and Hang Liu. A survey of uncertainty estimation methods
471 on large language models. In *Findings of the Association for Computational Linguistics: ACL*
472 *2025*, pages 21381–21396, 2025.

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

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

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

481 Mingqi Yuan, Man-On Pun, and Dong Wang. Rényi state entropy maximization for exploration
482 acceleration in reinforcement learning. *IEEE transactions on artificial intelligence*, 4(5):1154–
483 1164, 2022.

- 484 Mert Yuksekgonul, Daniel Kocaja, Xinhao Li, Federico Bianchi, Jed McCaleb, Xiaolong Wang, Jan
485 Kautz, Yejin Choi, James Zou, Carlos Guestrin, et al. Learning to discover at test time. *arXiv*
486 *preprint arXiv:2601.16175*, 2026a.
- 487 Mert Yuksekgonul, Daniel Kocaja, Xinhao Li, Federico Bianchi, Jed McCaleb, Xiaolong Wang, Jan
488 Kautz, Yejin Choi, James Zou, Carlos Guestrin, et al. Learning to discover at test time. *arXiv*
489 *preprint arXiv:2601.16175*, 2026b.
- 490 Yuanzhao Zhai, Han Zhang, Yu Lei, Yue Yu, Kele Xu, Dawei Feng, Bo Ding, and Huaimin Wang.
491 Uncertainty-penalized reinforcement learning from human feedback with diverse reward lora en-
492 sembles. *arXiv preprint arXiv:2401.00243*, 2023.
- 493 Yuanzhao Zhai, Yu Lei, Han Zhang, Yue Yu, Kele Xu, Dawei Feng, Bo Ding, and Huaimin Wang.
494 Uncertainty-penalized reinforcement learning from human feedback with diversified reward lora
495 ensembles. *Information Processing & Management*, 63(3):104548, 2026.

496 Appendix

497 A Baseline objective: TTT-DISCOVER in full notation

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

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

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

$$\text{KL}(q_\beta \parallel \text{Unif}) = \ln 2, \quad (10)$$

505 which exists and is unique whenever the within-group rewards are not all identical, since
506 $\text{KL}(q_\beta \parallel \text{Unif})$ is continuous and strictly increasing in β on $(0, \infty)$ with limits 0 and $\ln G$. We
507 solve Eq. (10) by 60 iterations of bisection on $[0, 10^6]$; on the degenerate equal-reward set, the bi-
508 section saturates at the upper bracket and we fall back to $A_i \equiv 0$. Given $\beta(s)$, rollout $i \in \{1, \dots, G\}$
509 receives the leave-one-out entropic weight and advantage

$$w_i = \frac{\exp(\beta R_{\text{exec}}^{(i)})}{\frac{1}{G-1} \sum_{j \neq i} \exp(\beta R_{\text{exec}}^{(j)})}, \quad A_i = w_i - 1, \quad (11)$$

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

$$\mathcal{L}_{\text{PG}}(\theta) = - \sum_{t=1}^{|o|} \min \left[\rho_t(\theta) A_i, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon) A_i \right], \quad (12)$$

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

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

$$\mathcal{L}_{\text{PG}}^{(k)}(\theta_k) = - \sum_{t=1}^{|o|} \min \left[\rho_t^{(k)}(\theta_k) A_i^{\text{shaped}}, \text{clip}(\rho_t^{(k)}, 1-\epsilon, 1+\epsilon) A_i^{\text{shaped}} \right], \quad (13)$$

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

516 **Token-wise KL adjustment.** When $\lambda_{\text{KL}} > 0$, the scalar A_i^{shaped} is broadcast to a per-token ten-
517 sor and the token-wise correction of Yuksekogonul et al. [2026b] is applied before the clipped-IS
518 reduction. Concretely, for each position t the advantage entry is adjusted by $-\lambda_{\text{KL}} (\log \pi_\theta(o_t \mid$
519 $q, o_{<t}) - \log \pi_{\text{base}}(o_t \mid q, o_{<t}))$, which under sampling from π_θ gives the standard k1 single-sample
520 stochastic estimator of $\nabla_\theta \lambda_{\text{KL}} \text{KL}(\pi_\theta \parallel \pi_{\text{base}})$ folded into the per-token PG target—no second auto-
521 grad pass through the policy is needed. The transcription is identical to the reference implementation
522 and is stated here only to fix notation used in the proof of Proposition 1.

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 $\alpha, \beta_{\text{ref}}, \gamma_{\text{max}}, \lambda_{\text{KL}}, \lambda_{\text{NNM}}$.

Ensure: Updated ensemble $\{\theta_k\}_{k=1}^K$.

```

1: // Stage 1: Rollout generation (round-robin over adapters)
2: Build prompt  $q$  from  $(d, s)$ .
3: for  $i = 1, \dots, G$  do
4:    $k_i \leftarrow ((i-1) \bmod K) + 1$   $\triangleright$  1-indexed;  $K \mid G$  so the assignment pattern
       $(1, 2, \dots, K, 1, 2, \dots)$  repeats identically every group
5:    $o_i \sim \pi_{\theta_{k_i}}(\cdot \mid q, s)$ 
6:    $R_{\text{exec}}^{(i)} \leftarrow \text{PROGRAMCHECKER}(s, o_i)$ 
7: end for
8: // Stage 2: Mutual-information scoring (single GPU pass)
9:  $\{p_{\theta_k}\}_{k=1}^K \leftarrow \text{CHUNKEDSCOREALLADAPTERS}(o_{1:G})$ 
10: for  $i = 1, \dots, G$  do
11:   for  $t = 1, \dots, |o_i|$  do
12:      $\bar{p}_t \leftarrow \frac{1}{K} \sum_{k=1}^K p_{\theta_k}(\cdot \mid q, o_{<t}^{(i)})$ 
13:      $\text{MI}_t^{(i)} \leftarrow H(\bar{p}_t) - \frac{1}{K} \sum_{k=1}^K H(p_{\theta_k}(\cdot \mid q, o_{<t}^{(i)}))$   $\triangleright$  Eq. (2)
14:   end for
15:    $U_i \leftarrow \text{top-7\%-mean}_t \text{MI}_t^{(i)}$   $\triangleright$  Eq. (3)
16: end for
17: // Stage 3: Adaptive temperature and LOO advantage (on  $R_{\text{exec}}$  only)
18:  $\beta(s) \leftarrow \text{BISECTIONSOLVE}\{\beta > 0 : \text{KL}(q_\beta \parallel \text{Unif}) = \ln 2\}$  on  $\{R_{\text{exec}}^{(i)}\}$   $\triangleright$  Eq. (10)
19: for  $i = 1, \dots, G$  do
20:    $w_i \leftarrow \exp(\beta R_{\text{exec}}^{(i)}) / \frac{1}{G-1} \sum_{j \neq i} \exp(\beta R_{\text{exec}}^{(j)})$ 
21:    $A_i \leftarrow w_i - 1$   $\triangleright$  Eq. (11)
22: end for
23: // Stage 4: MI-shaped advantage
24:  $\bar{U} \leftarrow \frac{1}{G} \sum_i U_i$ ;  $\sigma_U \leftarrow \text{std}_{\text{unbiased}}(U)$  (cf. Eq. (4));  $|\bar{A}| \leftarrow \frac{1}{G} \sum_j |A_j|$ 
25:  $\gamma_{\text{eff}}(s) \leftarrow \alpha \cdot \min(\beta(s)/\beta_{\text{ref}}, \gamma_{\text{max}})$   $\triangleright$  Eq. (5)
26: for  $i = 1, \dots, G$  do
27:    $\tilde{U}_i \leftarrow \text{clip}((U_i - \bar{U})/\sigma_U, -3, 3)$   $\triangleright$  Eq. (4)
28:    $A_i^{\text{shaped}} \leftarrow A_i + \gamma_{\text{eff}}(s) \cdot |\bar{A}| \cdot \tilde{U}_i$   $\triangleright$  Eq. (6)
29: end for
30: if  $\lambda_{\text{KL}} > 0$  then
31:   Broadcast  $A_i^{\text{shaped}}$  per-token; add token-wise KL correction against  $\pi_{\text{base}}$  (Appendix A)
32: end if
33: // Stage 5: Per-adapter backward, NNM backward, then optimiser step
34: for  $k = 1, \dots, K$  do
35:   Compute  $\frac{1}{K} \mathcal{L}_{\text{PG}}^{(k)}(\theta_k)$  over all  $G$  rollouts and call backward(); gradients accumulate into  $\theta_k.\text{grad}$   $\triangleright$  Eq. (13)
36:   Release adapter  $k$ 's computation graph before advancing
37: end for
   // KL-to-base contribution is already inside  $\mathcal{L}_{\text{PG}}^{(k)}$  via the Stage 4 token-wise correction (App. A)
38: Compute  $\lambda_{\text{NNM}} \mathcal{L}_{\text{NNM}}$  and call backward(); gradients accumulate into all  $\{\theta_k\}$   $\triangleright$  Eq. (7)
39: for  $k = 1, \dots, K$  do
40:   AdamW step on  $\theta_k$  using its accumulated gradient; zero  $\theta_k.\text{grad}$ 
41: end for

```

524 C Proofs

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

526 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
 527 $A_i^{\text{shaped}} = A_i$ identically. Otherwise

$$\sum_{i=1}^G \tilde{U}_i = \sum_{i=1}^G \text{clip}\left(\frac{U_i - \bar{U}}{\sigma_U}, -3, 3\right).$$

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

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

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

$$\left| \sum_{i=1}^G \tilde{U}_i \right| \leq \sum_{i \in \mathcal{C}} (|z_i| - 3).$$

531 By Cauchy–Schwarz, $\sum_{i \in \mathcal{C}} |z_i| \leq \sqrt{|\mathcal{C}| \sum_{i \in \mathcal{C}} z_i^2} = \sqrt{|\mathcal{C}|(G-1)}$ (using σ_U unbiased, so $\sum_i z_i^2 =$
 532 $G-1$); a looser but more interpretable bound uses $|z_i| \leq \sqrt{G-1}$ to give $\sum_{i \in \mathcal{C}} (|z_i| - 3) \leq$
 533 $|\mathcal{C}|(\sqrt{G-1} - 3)$. When $\mathcal{C} = \emptyset$ both sides are exactly 0; note in particular that $|z_i| \leq \sqrt{G-1}$
 534 forces $\mathcal{C} = \emptyset$ whenever $G \leq 10$, so the looser bound is non-vacuous only in the regime $G \geq 11$
 535 (where $\sqrt{G-1} - 3 > 0$). Substituting into Eq. (6) and using that neither $\gamma_{\text{eff}}(s)$ nor $|\bar{A}|$ depends
 536 on i ,

$$\left| \sum_{i=1}^G A_i^{\text{shaped}} - \sum_{i=1}^G A_i \right| = \gamma_{\text{eff}}(s) |\bar{A}| \left| \sum_{i=1}^G \tilde{U}_i \right| \leq \gamma_{\text{eff}}(s) |\bar{A}| |\mathcal{C}| (\sqrt{G-1} - 3),$$

537 which is the statement of the proposition.

538 We emphasise that this is strictly weaker than unbiasedness of the policy-gradient estimator
 539 $\sum_i A_i \nabla \log \pi(o_i)$ in the standard sense [Thrapoulidis et al., 2025]: because $\tilde{U}_i$ depends on roll-
 540 out i through U_i , the substitution $A_i \mapsto A_i^{\text{shaped}}$ does not preserve the expected gradient. The bonus
 541 is deliberately biased; the directional bias toward high-MI rollouts is the exploration mechanism.
 542 Proposition 1 isolates the bias to the gradient direction by bounding the residual perturbation to the
 543 aggregate within-group magnitude. $\square$

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

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

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

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

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

549 with equality in both inequalities iff $m = Kr$ (i.e. W has full row rank, which requires and is
 550 feasible exactly when $Kr \leq d_{\text{in}}$, the standing hypothesis) and all Kr singular values are equal. Let
 551 s denote this common singular value; then $\|W\|_F^2 = Kr s^2 = Kc^2$ gives $s = c/\sqrt{r}$. Under the
 552 full-row-rank hypothesis just established, equal singular values are equivalent to $WW^\top = s^2 I_{Kr}$,
 553 i.e. the rows of W are mutually orthogonal with common norm s . Restricted to a single block A_k ,
 554 this forces A_k to have r equal singular values all equal to $s = c/\sqrt{r}$. Across blocks, mutual row
 555 orthogonality of W forces the row spaces of $A_1, \dots, A_K$ to be mutually orthogonal subspaces of
 556 $\mathbb{R}^{d_{\text{in}}}$; this is feasible iff $Kr \leq d_{\text{in}}$, which holds by hypothesis. The global maximum of $\|W\|_*$ on
 557 the constraint set therefore satisfies both conditions, as claimed. $\square$

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 $\text{MI}_t \equiv 0$. Consequently $U_i \equiv 0$, $\bar{U} = 0$, $\sigma_U = 0$, and by the convention stated below Eq. (4) we set $\tilde{U}_i \equiv 0$. The shaped advantage reduces to $A_i^{\text{shaped}} = A_i$ and Eq. (13) reduces to Eq. (12). The hypothesis $\lambda_{\text{NNM}} = 0$ removes $\mathcal{L}_{\text{NNM}}$ 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 $\mathcal{L}_{\text{PG}}(\theta_1) + \lambda_{\text{KL}} \text{KL}(\pi_\theta \|\pi_{\text{base}})$, 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. $\square$

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 $\text{chunk} = 256$ 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 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 $T = 26\text{k}$ this would peak at ≈ 16 GB without chunking, and we reduce it to ≈ 8.1 GB by streaming logsoftmax over the same $\text{chunk} = 256$ 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
<i>Model</i>		
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	
<i>Optimiser</i>		
Optimiser	AdamW	
Learning rate	4×10^{-5}	
<i>RL training</i>		
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 $\text{KL} = \ln 2$
IS clip ϵ	0.2	
KL penalty λ_{KL}	0.01	
Remove constant-reward groups	yes	
<i>Sampling and context</i>		
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 $K=1$, $\alpha=0$, $\lambda_{\text{NNM}}=0$, and disables streaming MI.

Hyperparameter	Value	Notes
<i>Ensemble</i>		
Number of adapters K	5	round-robin rollout assignment
<i>MI exploration bonus (Sec. 2.3)</i>		
MI metric	true MI, top-7% tokens	Eq. (2), Eq. (3)
MI coefficient α	0.1	base exploration strength
γ_{max} ratio	10.0	clips $\gamma_{\text{eff}}/\alpha$
MI clip range	$[-3, 3]$	on standardised $\tilde{U}_i$
<i>Nuclear-norm regularisation (Sec. 2.4)</i>		
NNM coefficient λ_{NNM}	0.075	applied to $A^{(k)}$ matrices only
Frobenius-normalised variant	no	raw nuclear norm used
<i>Streaming MI early-stop (Appendix F)</i>		
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 25th 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 25th 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:

Problem	Epochs 0–2			All epochs		
	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.

Problem	Winning template	Top-30 (count)		Adoption (%)	
		Baseline	UG-TTT	Baseline	UG-TTT
AC1	<code>target_sum = $\sqrt{2n}$</code> (Cauchy–Schwarz)	1/30	18/30	3.0	28.0
AC2	<code>scipy.diff_evo</code> $\cup$ 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 $h \in \mathbb{R}_{\geq 0}^{1000}$ 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.

659

(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,

```

660

```

    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).

661

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).

662

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: he claims mentioned in the abstract and introduction are well justified in the experimentation of the paper.

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Yes, the authors discuss the limitation of the paper in separate section.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate “Limitations” section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren’t acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All three propositions are stated with explicit assumptions; proofs appear in App. C with the reductions used.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, we mention all baselines, all experimental results and we hope to open source code as well in time.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Yes, all the experimental details are described

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Yes we provide all the training and test details necessary to understand results in our experimentation section.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All experiments are single-run due to the compute constraints of RL training on a single RTX Pro 6000 (96 GB); each 6-epoch run requires approximately 32 GPU-hours. The primary evidence for the mechanism, sustained vs. collapsed ensemble MI (Table 2) is structural and not stochastic-outcome-dependent.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We explicitly explain the GPU resources used for our experiments in the main section of the paper.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We have reviewed the NeurIPS code of ethics and the paper conforms with it.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Discussed in Appendix H.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: There are no such models used in our paper.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All existing assets are cited and used under their respective open-source licenses. Qwen3-8B (<https://huggingface.co/Qwen/Qwen3-8B>) is used under the Apache 2.0 License; mLoRA [Ye et al., 2024] (<https://github.com/TUDB-Labs/mLoRA>) is used under the Apache 2.0 License; the TTT-DISCOVER prompts and verifiers [Yuksekgonul et al., 2026b] (<https://github.com/test-time-training/discover>) are used under the MIT License. No datasets or scraped data are used.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release new datasets or pretrained models. The training code that implements [UG-TTT](#) on top of existing open-source assets will be released upon publication, as stated in the open-access item above.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [No]

Justification: There are no such experiments in the paper.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [No]

Justification: There are no such experiments or studies involved which might inculcate any risks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs (Sonnet 4.6 and Claude Code by Anthropic) were used solely for editing, proofreading, $\LaTeX$ formatting, and assisting with implementation code. They were not part of the core methodology, scientific contribution, or experimental analysis, so declaration is not strictly required by the policy. We disclose this usage in the LLM usage section of the main paper for transparency.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.