# AdaPO: Towards Principled Efficient Rollout Allocation in Test-Time Training

**Youkang Wang**[1], **Jian Wang**[2], **Tianyi Zeng**[1], **Xiao-Yong Wei**[1,2]*, **Qing Li**[1]
[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[2]College of Computer Science, Sichuan University

## Abstract

Test-time training and test-time policy optimization have emerged as effective tools for improving the robustness of LLMs under distribution shift. A common approach constructs label-free reward signals from multiple sampled rollouts using majority voting, but existing methods typically rely on a fixed, large rollout budget for every query. This fixed allocation is inefficient because many queries reach consensus well before the rollout budget is exhausted. In this work, we introduce **AdaPO**, which replaces fixed-budget majority voting with an adaptive two-stage voting procedure that moves from electing to confirming when a leader is statistically dominant, and stops once the posterior error of accepting that leader is sufficiently low. AdaPO is plug-and-play with standard policy optimization algorithms such as PPO and GRPO, and it is also compatible with supervised fine-tuning at test time. Our theoretical results show that, under the symmetric categorical noise model with known likelihood parameters and on paths where the leading pair of answers is stable, AdaPO's electing-stage trigger reduces to the classical sequential probability ratio test (SPRT), while the confirming-stage rule yields a principled posterior-threshold stopping mechanism. Across representative reasoning benchmarks, AdaPO substantially reduces compute, achieving over 40% token savings on GPQA while maintaining competitive accuracy. The source code is available at `https://github.com/GeometricMLMechanism/AdaPO`.

## 1 Introduction

LLMs have made major progress in language understanding [1, 2], but their performance can drop sharply under test-time distribution shifts (e.g., new domains, reasoning patterns, or linguistic variation) [3]. This motivates test-time adaptation methods that adjust behavior on the fly. Test-time learning (TTL) adapts models during inference, initially via lightweight, parameter-free mechanisms such as entropy minimization or batch-norm updates [4, 5]. More recent test-time training (TTT) performs explicit parameter updates at inference to improve task-specific robustness, such as few-shot shift [6] and selective fine-tuning [7].

Within the TTT paradigm, test-time policy optimization has recently gained traction. Casting inference as an RL problem, methods such as one-shot RL with verifiable rewards (RLVR) [8] and TTRL [9] use consensus signals (e.g., majority voting) to produce label-free rewards that guide policy updates, linking reasoning consistency to likely correctness. A key limitation is *rollout inefficiency*: most test-time RL methods allocate a fixed, large number of rollouts per query and use Majority Voting to obtain stable reward estimates. This ignores variation in query difficulty, since easy cases often reach consensus after only a few samples while hard cases require many more, making inference cost a major obstacle to deployment.

*Corresponding Author: Xiao-Yong Wei. Email: cs007.wei@polyu.edu.hk

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

This paper presents **AdaPO**, a general framework that addresses this challenge by casting reward estimation as an adaptive two-stage voting automaton, illustrated in Figure 1. Rather than fixing rollout budget in advance, AdaPO monitors the partial vote and applies SPRT-grounded trigger and posterior-threshold stopping rules [10, 11]. More specifically, AdaPO explores by accumulating voting evidence in the *Electing* stage and moves to *Confirming* once the current leader answer becomes statistically dominant over the remaining candidates, which happens earlier on easier instances. In the *Confirming* stage, AdaPO exploits the leader, testing whether the dominance persists, and stops sampling when the posterior confidence in the leader exceeds a prescribed target [12]. Under the symmetric categorical noise model with known likelihood parameters, and as long as the identities of the leading two answers stay stable, the electing-stage trigger reduces to the classical SPRT, which minimizes expected sample size subject to error constraints. AdaPO is compatible with common policy optimization methods such as PPO [13] and GRPO [14], and it also supports test-time supervised fine-tuning.

Our contributions are threefold. 1) We formalize the inefficiency of fixed-budget majority voting in test-time training and characterize adaptive rollout allocation through principled perspectives. 2) We introduce AdaPO, a plug-and-play framework replacing heuristic fixed-budget voting with a finite automaton decision procedure. 3) We show empirically that AdaPO delivers substantial compute savings while preserving competitive accuracy across representative reasoning benchmarks.

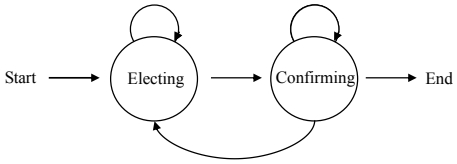

Figure 1: Finite-state procedure of AdaPO.

## 2 Related Work

**Test-Time Training with LLMs.** Traditional inference assumes fixed parameters, yet Large Language Models (LLMs) often suffer from distribution shifts in complex reasoning tasks. While early *test-time adaptation* (TTA) [4, 5] focused on lightweight updates like entropy minimization, recent *test-time training* (TTT) [15, 6, 7] paradigm enables more expressive adaptation through full parameter updates. Prevalent TTT approaches range from retrieving nearest-neighbor contexts [16] to continuous self-supervised refinement via perplexity-based objectives [3]. These methods demonstrate that dynamic learning at inference is crucial for the self-improvement of LLMs.

**Test-Time Reinforcement Learning.** Integrating reinforcement learning (RL) into the test-time training paradigm has emerged as a powerful tool for reasoning, often leveraging external verifiers or feedback [17]. Specifically, Test-Time Reinforcement Learning (TTRL) [9] utilizes multiple rollouts and consensus-based rewards to update policies. However, existing approaches rely on a fixed, redundant number of rollouts to stabilize updates, which makes them costly. Recent work, such as RL with one training example [8] and Compute as Teacher (CaT) [18], yields gains from minimal examples but does not optimize rollout allocation. Inspired by efficient test-time scaling methods [19–21], we bridge this gap with a statistically grounded early-stopping mechanism for adaptive rollout allocation.

**Policy Optimization and Efficiency.** The effectiveness of TTT often hinges on the underlying optimization algorithm. Proximal Policy Optimization (PPO) [13] and Group Relative Policy Optimization (GRPO) [14] are widely used for stabilizing updates under limited feedback. Despite their success, these algorithms treat rollout generation uniformly. As a plug-and-play mechanism that sets rollout budgets dynamically, AdaPO improves the efficiency of standard surrogates like GRPO without compromising optimization stability.

## 3 Methodology

### 3.1 Preliminaries

Given an input prompt $x \in \mathcal{X}$, the model $\pi_\theta(\cdot \mid x)$, parameterized by $\theta$, introduces a stochastic policy over completions $y \in \mathcal{Y}$. One can perform test-time training by updating the policy using $N$ rollouts, with rewards computed relative to a reference answer. Because ground-truth labels are unavailable

during testing, this reference is typically chosen as the best among $M$ sampled completions generated at test time, via a practical and widely adopted heuristic: majority voting.

**Rollouts and Majority Voting:** Let $\mathcal{E}(y) : \mathcal{Y} \to \mathcal{A}$ denote an *answer extractor* that maps any completion $y \in Y$ to an answer $a \in \mathcal{A}$, where $\mathcal{A}$ denotes the set of $m$ distinct candidate answers for the prompt $x$. Majority voting proceeds by sampling $M$ rollouts to obtain votes $D_{1:M} = \{D_1, \ldots, D_M\}$, where $D_t = \mathcal{E}(y_t) \in \mathcal{A}$ for $1 \le t \le M$ and $y_t \sim \pi_\theta(\cdot \mid x)$. Let $V(a)$ be the count (frequency) of answer $a \in \mathcal{A}$ over $D_{1:M}$. The best answer selected with $M$ budget is then $a^\star_M = \arg\max_{a_j \in \mathcal{A}} V(a_j \mid D_{1:M})$.

**Dilemma of Majority Voting:** A key limitation of majority voting is that its cost scales directly with the rollout budget $M$. Using a larger $M$ yields a more reliable selection by stabilizing the vote statistics, but it can be prohibitively expensive in test-time training, since a large amount of rollouts will be needed just to determine the reference label for reward. Conversely, reducing $M$ lowers the cost but makes the resulting statistics much less reliable.

### 3.2 A Finite Voting Automaton

To balance robustness and efficiency, we learn a prompt-dependent stopping time $\tau \le M$ that halts voting early once we have high confidence that the partial winner matches the full-$M$ majority vote. Let $\delta \in \mathcal{A}$ denote the selected answer at stopping. At a high level, Our optimization objective is

$$\min_{\tau,\delta} \mathbb{E}[\tau] \qquad \text{s.t.} \qquad \mathbb{P}(\delta \neq a^\star_M) \le \epsilon,$$

where $\epsilon \in (0, 1)$ is the allowable selection error. Note that $a^\star_M$ is the best proxy reward signal available at test-time to approximate ground-truth when true label is absent.

We represent the sequential voting procedure as a deterministic finite-state automaton (Figure 1) with state set $\mathcal{Q} = \{\mathit{Start}, \mathit{Elc}, \mathit{Cfm}, \mathit{End}\}$, where $\mathit{Elc}$ and $\mathit{Cfm}$ are shorthand for $\mathit{Electing}$ and $\mathit{Confirming}$. Let $\Phi_t \in \mathcal{Q}$ denote the stage after processing $t$ rollouts. The update is governed by explicit rules, which will be derived and introduced in Sections 3.3 and 3.4.

### 3.3 Trigger Rule at *Electing* Stage

We now instantiate the high-level objective above for the transition from $\mathit{Electing}$ to $\mathit{Confirming}$. Let $\alpha, \beta \in (0, 1)$ denote the allowable type-I and type-II error rates. At time $t$, let $a_L = \arg\max_{a_j \in \mathcal{A}} V(a_j | D_{1:t})$ be the current leader, let $a_R = \arg\max_{a_j \in \mathcal{A}, a_j \neq a_L} V(a_j | D_{1:t})$ be the runner-up, and define the vote gap $\Delta_t = V(a_L) - V(a_R)$. We denote by $p_0 = \mathbb{P}(D_t = a^\star_M)$ the probability that a rollout produces the majority answer $a^\star_M$. To obtain a tractable reduction of the multi-class voting process into a sequential test, we adopt the *symmetric categorical noise* model: votes are conditionally independent given $a^\star_M$, and each of the other $m - 1$ answers is produced with probability $q_0 = \frac{1-p_0}{m-1}$, with $p_0 > 1/m$.

Because $a_L$ and $a_R$ are defined from the running votes, the hypotheses tested below are data-dependent. We therefore state the optimality result on the *identity-stable path* $\mathcal{G}$, the event that the leader/runner-up pair $(a_L, a_R)$ does not change between the minimum rollout count $t_0$ and the trigger time; off $\mathcal{G}$ the result is an approximation whose error we quantify below.

**Proposition 3.1** (Electing-Stage Trigger Rule)**.** *Under the symmetric categorical noise model and on the identity-stable path $\mathcal{G}$, the transition from Electing to Confirming is the acceptance boundary of Wald's SPRT for $H_1 : a^\star_M = a_L$ against $H_0 : a^\star_M = a_R$, i.e., it triggers at the first time*

$$\Delta_t \ge \frac{\log A}{\log \kappa}, \qquad \textit{where} \quad \kappa := \frac{p_0(m-1)}{1-p_0}, \quad A := \frac{1-\beta}{\alpha}.$$

*Equivalently, with integer boundary $\Delta_A := \left\lceil \frac{\log A}{\log \kappa} \right\rceil$, the trigger time is $\tau = \inf\{t \ge t_0 : \Delta_t \ge \Delta_A\}$, where $t_0$ is the minimum rollout count ($t_0 = N$ in AdaPO).*

***Derivation*** *(full proof in Appendix E.1). Step 1: reduction to a binary test.* We compare the two hypotheses

$$H_1 : \ a^\star_M = a_L \ \text{(the leader wins)} \qquad \text{vs.} \qquad H_0 : \ a^\star_M = a_R \ \text{(the runner-up wins).}$$

Under the symmetric model every other candidate has the same per-rollout probability $q_0$ as $a_R$ but strictly fewer votes, so its posterior probability of being $a_M^\star$ is dominated by that of $a_R$; the dominant error event $\{a_L \neq a_M^\star\}$ is therefore governed by $a_R$, and comparing $(a_L, a_R)$ is the worst case.

*Step 2: likelihood ratio.* Under $H_j$ the votes are i.i.d., so $\mathbb{P}(D_{1:t} \mid H_j) = \prod_{k=1}^{t} \mathbb{P}(D_k \mid H_j) = p_0^{V(a_j^\star)} \prod_{a \neq a_j^\star} q_0^{V(a)}$ with $a_1^\star = a_L$ and $a_0^\star = a_R$. All factors of candidates $a \notin \{a_L, a_R\}$ appear identically in both hypotheses and cancel, which gives

$$\Lambda_t = \frac{\mathbb{P}(D_{1:t} \mid H_1)}{\mathbb{P}(D_{1:t} \mid H_0)} = \frac{p_0^{V(a_L)}\, q_0^{V(a_R)}}{p_0^{V(a_R)}\, q_0^{V(a_L)}} = \left(\frac{p_0}{q_0}\right)^{V(a_L)-V(a_R)} = \kappa^{\Delta_t}.$$

Each rollout thus multiplies $\Lambda_t$ by $\kappa$ (vote for $a_L$), by $1/\kappa$ (vote for $a_R$), or by 1 (any other answer), and $\kappa > 1$ because $p_0 > 1/m$. With a uniform prior, $\mathbb{P}(H_1 \mid D_{1:t}) = \Lambda_t/(1+\Lambda_t)$.

*Step 3: Wald's boundaries.* Wald's SPRT stops at the first time the likelihood ratio leaves the interval $(B, A)$:

$$\text{accept } H_1 \text{ if } \Lambda_t \geq A = \tfrac{1-\beta}{\alpha}, \qquad \text{accept } H_0 \text{ if } \Lambda_t \leq B = \tfrac{\beta}{1-\alpha};$$

for a fixed pair of simple hypotheses it attains error rates $(\alpha, \beta)$ (up to overshoot) with the minimum expected sample size. Substituting $\Lambda_t = \kappa^{\Delta_t}$ and taking logarithms ($\log \kappa > 0$), the acceptance condition becomes

$$\kappa^{\Delta_t} \geq A \iff \Delta_t \geq \frac{\log A}{\log \kappa} \iff \Delta_t \geq \Delta_A,$$

where the last step uses that $\Delta_t$ is an integer. On $\mathcal{G}$ the pair $(a_L, a_R)$ is fixed, so $\tau$ is exactly the SPRT acceptance time for $H_1$. □

**Scope of optimality.** Two features of AdaPO differ from Wald's setting of a fixed pair of hypotheses with both boundaries active from the start. (i) *Data-defined hypotheses.* $a_L$ and $a_R$ are the running arg-max and second arg-max, so the tested pair can change before the trigger. The reduction is exact on $\mathcal{G}$; leaving $\mathcal{G}$ requires a competitor to overtake a leader with positive drift, whose probability decays exponentially in the vote gap by the large-deviation argument behind the $e^{-cM}$ term in Proposition 3.3 (Appendix E). This adds an exponentially small correction, and Appendix B verifies the top-2 reduction empirically. (ii) *One-sided trigger.* The electing stage only checks the upper boundary $A$. The lower boundary $B$ is not discarded but relocated: it is the condition that sends *Confirming* back to *Electing* (Theorem 3.4), which lets AdaPO withdraw from a false leader instead of committing once. Hence the classical optimality guarantee applies to the electing sub-procedure conditional on path stability; the two-stage automaton as a whole is a new construction built on these SPRT components rather than a classical SPRT.

**Modeling assumptions.** The symmetric categorical noise model is not meant to describe real rollout distributions, which are often skewed and dependent. It is the minimal homogeneous structure under which the multi-class voting problem reduces to a binary sequential test whose likelihood ratio depends only on the leader/runner-up gap; without it, the likelihood ratio depends on the full vote histogram and the decision rule becomes intractable. Appendix B shows empirically that AdaPO is robust to violations of this model.

**Efficiency Analysis.** We assess efficiency by examining the expected time to trigger the transition.

**Theorem 3.2** (Expected Triggering Time under the SPRT-Induced Rule)**.** *Consider the untruncated process ($M = \infty$). Let $\tau$ denote the confirmation trigger time from Proposition 3.1. Then the expected trigger time satisfies*

$$\mathbb{E}[\tau] = O\left(\frac{1}{\mu} \cdot \frac{\log(1/\alpha)}{\log \kappa}\right), \qquad \mu = \frac{m p_0 - 1}{m - 1}.$$

*Moreover, if $p_0 = \frac{1}{m} + \xi$ with $\xi \ll 1$, then $\mathbb{E}[\tau] = O\left(\frac{\log(1/\alpha)}{\xi^2}\right)$, where $\xi$ is the assumed small probability advantage.*

Theorem 3.2 (proof in Appendix E.2) implies that the expected rollout complexity grows only logarithmically with the target error probability, and scales quadratically with the inverse of the accuracy gap. Consequently, easy instances (with larger $p_0$) tend to terminate quickly, whereas harder instances consume more budget in a controlled manner.

**Accuracy Analysis.** We analyze the accuracy of the proposed voting procedure by bounding its *decision regret*, measuring the probability of disagreeing with the full-budget majority vote. For a trigger at time $\tau \le M$ and outputs $a_\tau$, we define the decision regret as

$$\mathcal{R}_M \;:=\; \mathbb{P}\big(a_\tau \neq a_M^\star\big).$$

**Proposition 3.3** (Decision Regret at Confirmation Trigger)**.** *The decision regret at confirmation trigger is bounded by*

$$\mathcal{R}_M \;\le\; \max\{\alpha,\beta\} \;+\; e^{-cM},$$

*for a constant $c > 0$ depending only on $(p_0, m)$.*

The dominant term $\max\{\alpha,\beta\}$ is controlled by the allowable error rates in the *Electing* stage (Proposition 3.1), ensuring statistical correctness at the confirmation trigger. The second term $e^{-cM}$ accounts for truncation at the maximum rollout budget $M$ and decreases exponentially with $M$. Full derivations are provided in Appendix E.3.

### 3.4 Posterior-Threshold Stopping Rule at *Confirming* Stage

The *Confirming* stage collects additional evidence for the triggered leader before termination, guarding against a false consensus at the trigger. Let $\gamma \in (0,1)$ be the target posterior confidence. For $t \ge \tau$, keep $a_L$ fixed to the leader at the trigger, let $\Delta_t$ be its vote gap to the strongest other answer, and let $Z_t$ count the time steps in $[\tau, t]$ at which the trigger condition $\Delta_t \ge \Delta_A$ holds (so $Z_\tau = 1$). We write $E_Z$ for the event that $Z$ such confirmations have been observed.

**Theorem 3.4** (Posterior-Threshold Confirming-Stage Stopping Rule)**.** *With $A = \frac{1-\beta}{\alpha}$ and $B = \frac{\beta}{1-\alpha}$, while in Confirming at time $t$ the next state is*

$$\begin{cases} \textit{End} & \textit{if } \mathbb{P}(a_M^\star = a_L \mid E_{Z_t}) \ge \dfrac{A\kappa^{Z_t-1}}{A\kappa^{Z_t-1}+1} \ge \gamma, \\ \textit{Electing} & \textit{if } \Delta_t < \log B / \log \kappa, \\ \textit{Confirming} & \textit{otherwise.} \end{cases}$$

*The stopping condition is equivalent to $Z_t \ge Z^\star(\gamma) := \left\lceil 1 + \frac{\log(\gamma/(1-\gamma)) - \log A}{\log \kappa} \right\rceil$.*

***Proof sketch*** *(full proof in Appendix F.1). Bayes factor.* At the trigger, the electing-stage SPRT guarantees a Bayes factor $\Lambda_\tau = \kappa^{\Delta_\tau} \ge A$ for $a_L$. Each subsequent confirmation requires another informative rollout in favor of $a_L$, whose likelihood ratio between $H_1$ and $H_0$ is $\kappa$, so after $Z_t$ confirmations the effective Bayes factor $\Lambda_{\text{eff}}(Z_t) = \mathbb{P}(E_{Z_t} \mid H_1)/\mathbb{P}(E_{Z_t} \mid H_0)$ satisfies $\Lambda_{\text{eff}}(Z_t) \ge A\kappa^{Z_t-1}$. *Posterior threshold.* Under a uniform prior the posterior of $a_L$ is $\Lambda_{\text{eff}}/(1+\Lambda_{\text{eff}})$, which is increasing in $\Lambda_{\text{eff}}$. Requiring it to exceed $\gamma$ therefore gives

$$\frac{A\kappa^{Z_t-1}}{1+A\kappa^{Z_t-1}} \ge \gamma \iff \log A + (Z_t-1)\log\kappa \ge \log\frac{\gamma}{1-\gamma} \iff Z_t \ge Z^\star(\gamma).$$

*Return.* If instead the gap falls below the lower SPRT boundary $\log B/\log\kappa$, the evidence now favors a competitor and the procedure returns to *Electing*. □

When $\gamma/(1-\gamma) \le A$, e.g. $\alpha = \beta = 1-\gamma = 0.05$ as in our main experiments, $Z^\star(\gamma) = 1$ and AdaPO stops at the first trigger; larger $\gamma$ adds confirmation rounds (Table 6). The extra cost is small and the error is controlled:

**Theorem 3.5** (Confirmation Time)**.** *For the untruncated process, $\mathbb{E}[\tilde{t}] = O\Big(\frac{Z^\star(\gamma)-1}{\mu}\Big)$ with $\mu = \frac{mp_0-1}{m-1}$, where $\tilde{t}$ is the number of rollouts after $\tau$ before reaching End (Appendix F.2).*

**Corollary 3.6** (Decision Regret at Termination)**.** *With maximum rollout budget $M$, the decision regret satisfies $\mathcal{R}_M^{(\gamma)} \le (1-\gamma) + e^{-cM}$ for a constant $c > 0$.*

The confidence level $\gamma$ thus becomes an explicit accuracy guarantee: conditioned on termination before truncation, the probability of selecting an answer other than $a_M^\star$ is at most $1-\gamma$, and the $e^{-cM}$ term accounts for prompts that are forcibly terminated at the budget $M$ (Appendix F.3).

**Algorithm 1:** AdaPO rollout allocation and policy update for a batch of prompts. The sequential variant is the case $B_{\text{par}} = 1$; $\hat{p}_0$ and $m$ are estimated once at $n_p = N$ and then frozen.

**Input:** policy $\pi_\theta$, prompts $\{x_p\}_{p\in\mathcal{P}}$, extractor $\mathcal{E}$, budgets $N \le M$, $\alpha, \beta, \gamma, \eta$, round size $B_{\text{par}}$

$A \leftarrow \frac{1-\beta}{\alpha}$, $B \leftarrow \frac{\beta}{1-\alpha}$; for every $p$: $V_p \leftarrow \mathbf{0}$, $n_p \leftarrow 0$, $\Phi_p \leftarrow$ ELC

**while** $\mathcal{P} \neq \emptyset$ **do**

  $\mathbf{Y} \leftarrow$ sample $\min(B_{\text{par}}, M - n_p)$ rollouts of $x_p$ for every $p \in \mathcal{P}$ ▷ *one parallel call*

  **foreach** $p \in \mathcal{P}$ **and** $y \in \mathbf{Y}_p$ (in order, while $\Phi_p \neq$ END) **do**

    $V_p(\mathcal{E}(y)) \mathrel{+}= 1$, $n_p \mathrel{+}= 1$; $a_L, a_R \leftarrow$ top-2 answers of $V_p$, $\Delta \leftarrow V_p(a_L) - V_p(a_R)$

    **if** $n_p < N$ **then continue**

    **if** $n_p = N$ **then** ▷ *estimate once, then freeze*

      $\hat{p}_0 \leftarrow \eta\, V_p(a_L)/N$, $m \leftarrow \max(|\{a : V_p(a) > 0\}|, 5)$, $\kappa \leftarrow \frac{\hat{p}_0(m-1)}{1-\hat{p}_0}$, $\Delta_A \leftarrow \lceil \frac{\log A}{\log \kappa} \rceil$, $\Delta_B \leftarrow \lfloor \frac{\log B}{\log \kappa} \rfloor$, $Z^\star \leftarrow Z^\star(\gamma)$

    **if** $\Phi_p =$ ELC **and** $\Delta \ge \Delta_A$ **then** ▷ *Electing*

      $\Phi_p \leftarrow$ CFM, $c_p \leftarrow a_L$, $Z_p \leftarrow 1$

    **else if** $\Phi_p =$ CFM **then** ▷ *Confirming*

      $\Delta_c \leftarrow V_p(c_p) - \max_{a \neq c_p} V_p(a)$

      **if** $\Delta_c \le \Delta_B$ **then** $\Phi_p \leftarrow$ ELC ▷ *evidence favors a competitor*

      **else if** $\Delta_c \ge \Delta_A$ **then** $Z_p \mathrel{+}= 1$

    **if** $\Phi_p =$ CFM **and** $Z_p \ge Z^\star$ **then** $\Phi_p \leftarrow$ END

  remove from $\mathcal{P}$ every $p$ with $\Phi_p =$ END or $n_p = M$

$a_p^\star \leftarrow c_p$ if $\Phi_p =$ END, else $\arg\max_a V_p(a)$ ▷ *budget-exhausted fallback*

**RL:** reward $R(y_i; a_p^\star) = \mathbb{I}[\mathcal{E}(y_i) = a_p^\star]$ on the first $N$ rollouts of each prompt, update $\theta$ (PPO/GRPO/RF++)

**SFT:** update $\theta$ on $\mathcal{L}(\theta) = -\frac{1}{|\mathcal{P}|} \sum_p \log \pi_\theta(G_p \mid x_p)$ with pseudo-labels $G_p$ built from $a_p^\star$

### 3.5 Test-Time Policy Optimization

Once a confident pseudo-label $a^\star$ is produced at the $End$ state, the retained $N$ rollouts are used to optimize the policy parameters $\theta$ at test time. (1) When applied to test-time reinforcement learning **(RL)**, each rollout receives a label-free reward indicating agreement with the consensus, i.e., $R(Y_i; a^\star) = \mathbb{I}[a_i = a^\star]$. Policy updates are then performed by maximizing the expected reward under $\pi_\theta(\cdot \mid x)$ using standard on-policy policy-gradient methods, applicable to all RL algorithms including PPO [13], GRPO [14], Reinforce++ [22], etc. (2) When applied to test-time supervised fine-tuning **(SFT)**, AdaPO determines a pseudo-label $G_i$ at a data-dependent stopping time $\tau_i$ for each test input $x_i$. The model is then updated via the standard cross-entropy objective $\mathcal{L}(\theta) = -\frac{1}{B} \sum_{i=1}^{B} \log \pi_\theta(G_i \mid x_i)$.

In practice, we instantiate stopping rollout rules of AdaPO in two variants, both summarized in Algorithm 1: (1) The **sequential variant** samples one rollout at a time and stops immediately once the stopping criterion is met, maximizing token efficiency; it corresponds to $B_{\text{par}} = 1$. (2) The **batch-sequential variant** generates $B_{\text{par}}$ rollouts per unfinished prompt in each parallel generation round and updates the prompt states only after the round, trading a small amount of extra token usage for parallelism and better hardware utilization. Both variants apply the same rule to the votes of a prompt one at a time and differ only in the rollouts generated after the stopping point within a round. Appendix A gives a more detailed listing (Algorithm 2).

## 4 Experiments

### 4.1 Experimental Setup

**Benchmarks.** We utilize a suite of challenging math and science benchmarks: (1) **MATH-500** [23], consisting of 500 representative high-school level math problems; (2) **AIME 2024** [24], a collection of 30 problems, representing Olympiad-level reasoning tasks; (3) **AMC 2023** [25], comprising 83 intermediate-level problems from the American Mathematics Competitions; (4) **GPQA** [26], a graduate-level science Q&A benchmark for biology, physics, and chemistry.

**Backbone Models.** To verify cross-architecture generalization and scalability, we evaluate AdaPO on a diverse set of open-weight LLMs, including Qwen2.5-Math-1.5B [27], Qwen2.5-7B [28], and

Table 1: Performance comparison between AdaPO and SOTA test-time reinforcement learning baseline. **Bold** values indicate the best performance for each backbone model per metric.

| | AIME 2024 | | | | | AMC 2023 | | | | | MATH-500 | | | | | GPQA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mean @16 | Pass @16 | Pass @1 | Toks. (↓) | Toks. Saving | Mean @16 | Pass @16 | Pass @1 | Toks. (↓) | Toks. Saving | Mean @16 | Pass @16 | Pass @1 | Toks. (↓) | Toks. Saving | Mean @16 | Pass @16 | Pass @1 | Toks. (↓) | Toks. Saving |
| **Qwen2.5-Math-1.5B** | | | | | | | | | | | | | | | | | | | | |
| TTRL w/ GRPO | 15.6 | 43.3 | 20.0 | 142M | **17.25%** | 47.7 | **83.1** | 54.2 | 159M | 26.93% | **72.3** | 90.6 | 73.6 | 218M | 32.57% | 26.5 | 87.8 | 30.5 | 251M | 43.20% |
| AdaPO w/ GRPO | 14.6 | **46.7** | 20.0 | **117M** | | **48.5** | 81.9 | 54.2 | **116M** | | **72.3** | 90.8 | **73.8** | **147M** | | **26.8** | 87.8 | **34.5** | **143M** | |
| TTRL w/ PPO | **18.5** | **46.7** | **23.3** | 143M | 15.20% | 48.1 | **83.1** | **55.4** | 160M | 26.80% | 71.9 | **91.2** | **73.8** | 231M | **35.01%** | 25.9 | 87.8 | 29.9 | 262M | **45.06%** |
| AdaPO w/ PPO | 16.0 | **46.7** | **23.3** | 121M | | 48.1 | **83.1** | 53.0 | 117M | | **72.3** | 91.0 | **73.8** | 150M | | 26.6 | 89.8 | 32.5 | 144M | |
| TTRL w/ RF++ | 11.9 | 43.3 | 20.0 | 198M | 14.35% | 39.3 | 81.9 | 49.4 | 242M | **27.42%** | 55.9 | 90.2 | 56.4 | 696M | 25.53% | 26.3 | **93.9** | 32.0 | 431M | 43.52% |
| AdaPO w/ RF++ | 10.4 | **46.7** | 20.0 | 170M | | 38.3 | 81.9 | 44.6 | 176M | | 56.2 | 89.8 | 57.6 | 518M | | 26.5 | 90.9 | 28.9 | 243M | |
| **Llama-3.2-1B-Instruct** | | | | | | | | | | | | | | | | | | | | |
| TTRL w/ GRPO | 3.5 | 23.3 | **6.7** | 134M | **47.62%** | 14.9 | 51.8 | 18.1 | 128M | **78.36%** | 30.3 | 66.0 | 30.8 | 181M | 14.77% | 26.4 | 82.2 | 28.4 | 134M | 44.74% |
| AdaPO w/ GRPO | 1.7 | 20.0 | **6.7** | 70M | | 15.2 | **55.4** | 16.9 | 28M | | 35.4 | 64.4 | 36.0 | 155M | | 25.2 | 81.2 | 26.9 | **74M** | |
| TTRL w/ PPO | **3.8** | **30.0** | **6.7** | 47M | 13.70% | 13.9 | **55.4** | 19.3 | 75M | 41.76% | 31.2 | 64.6 | 31.6 | 246M | **44.49%** | 25.7 | 83.2 | 28.9 | 159M | **46.19%** |
| AdaPO w/ PPO | **3.8** | **30.0** | **6.7** | 41M | | **17.1** | 54.2 | **22.9** | 44M | | 34.1 | 64.4 | 34.8 | **137M** | | 26.7 | 81.7 | **31.0** | 86M | |
| TTRL w/ RF++ | 1.5 | 16.7 | **6.7** | 44M | 30.02% | 11.4 | 54.2 | 16.9 | 32M | 18.96% | 33.4 | 65.4 | 32.2 | 557M | 29.40% | **27.1** | **85.8** | 30.5 | 478M | 41.96% |
| AdaPO w/ RF++ | 1.7 | 23.3 | **6.7** | **31M** | | 11.0 | 51.8 | 13.3 | **26M** | | **36.1** | **66.6** | **38.0** | 393M | | 26.7 | **85.8** | 28.9 | 278M | |
| **Qwen2.5-7B** | | | | | | | | | | | | | | | | | | | | |
| TTRL w/ GRPO | 23.5 | 40.0 | 23.3 | 87M | 38.56% | 52.1 | 78.3 | 54.2 | 111M | **49.06%** | 79.7 | 91.0 | 80.0 | 172M | 45.02% | 35.1 | 85.3 | 39.1 | 98M | 40.31% |
| AdaPO w/ GRPO | 23.5 | 43.3 | 23.3 | **53M** | | **56.8** | 77.1 | **57.8** | **56M** | | 79.5 | **91.6** | 79.8 | 94M | | 35.7 | 84.8 | 38.6 | **59M** | |
| TTRL w/ PPO | **24.4** | **46.7** | 23.3 | 104M | **42.85%** | 54.3 | 79.5 | 55.4 | 101M | 41.66% | **80.3** | 91.0 | 80.4 | 171M | **48.96%** | **37.9** | 83.8 | **40.1** | 109M | 39.83% |
| AdaPO w/ PPO | 24.0 | 43.3 | **26.7** | 60M | | 53.2 | 79.5 | 55.4 | 59M | | 78.1 | 91.0 | 78.4 | **87M** | | 36.8 | **88.8** | 39.1 | 66M | |
| TTRL w/ RF++ | 20.8 | 40.0 | 23.3 | 345M | 41.45% | 52.9 | **80.7** | 55.4 | 430M | 43.50% | 78.9 | 90.8 | 80.0 | 886M | 44.52% | 33.0 | 85.3 | 32.5 | 139M | **45.47%** |
| AdaPO w/ RF++ | 20.6 | 40.0 | 23.3 | 202M | | 56.0 | 79.5 | **57.8** | 243M | | 79.4 | 91.0 | **80.6** | 491M | | 33.0 | 84.3 | 34.0 | 76M | |

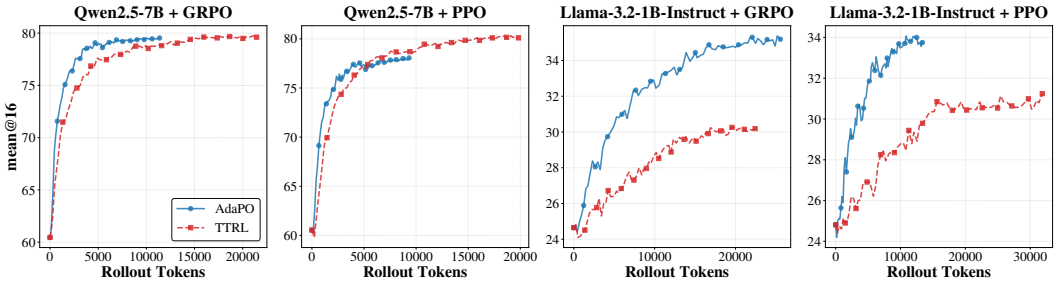


Figure 2: Performance vs. rollout tokens on the MATH-500 benchmark, comparing AdaPO and TTRL across different RL algorithms (GRPO and PPO).

Table 2: Performance of AdaPO-SFT vs. the test-time supervised fine-tuning (TTSFT) baseline.

| | AIME 2024 | | | | | AMC 2023 | | | | | MATH-500 | | | | | GPQA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mean @16 | Pass @16 | Pass @1 | Toks. (↓) | Toks. saving | Mean @16 | Pass @16 | Pass @1 | Toks. (↓) | Toks. Saving | Mean @16 | Pass @16 | Pass @1 | Toks. (↓) | Toks. Saving | Mean @16 | Pass @16 | Pass @1 | Toks. (↓) | Toks. Saving |
| **Qwen2.5-Math-1.5B** | | | | | | | | | | | | | | | | | | | | |
| TTSFT | 13.1 | 46.7 | **20.0** | 166M | 0.61% | 37.9 | 80.7 | 42.2 | 138M | 12.08% | **59.7** | 90.8 | **63.8** | 200M | 25.29% | **27.9** | **81.8** | 30.3 | 113M | 51.17% |
| AdaPO-SFT | **13.5** | **50.0** | **20.0** | **165M** | | **39.7** | **81.9** | **43.4** | **121M** | | 59.0 | **91.2** | 60.2 | **150M** | | 27.6 | 78.8 | **31.3** | **55M** | |
| **Llama-3.1-8B-Instruct** | | | | | | | | | | | | | | | | | | | | |
| TTSFT | 3.8 | 26.7 | **6.7** | 314M | 15.33% | **20.1** | **63.9** | **28.9** | **326M** | 17.16% | 43.5 | 81.2 | 44.2 | 653M | 19.84% | 28.8 | **85.9** | 30.3 | 382M | 61.89% |
| AdaPO-SFT | **6.2** | **36.7** | 3.3 | **266M** | | 18.7 | 60.2 | 19.3 | 270M | | **43.6** | **82.8** | **47.0** | **523M** | | **28.9** | 82.8 | **32.3** | **146M** | |

Table 3: Performance and wall-clock time of the batch-sequential variant of AdaPO ($B_{\text{par}} = 8$).

| | AIME 2024 | | | | | AMC 2023 | | | | | MATH-500 | | | | | GPQA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mean @16 | Pass @16 | Pass @1 | Time (↓) | Toks. saving | Mean @16 | Pass @16 | Pass @1 | Time (↓) | Toks. Saving | Mean @16 | Pass @16 | Pass @1 | Time (↓) | Toks. Saving | Mean @16 | Pass @16 | Pass @1 | Time (↓) | Toks. Saving |
| **Qwen2.5-Math-1.5B** | | | | | | | | | | | | | | | | | | | | |
| TTRL w/ GRPO | 15.6 | 43.3 | 20.0 | 15h39m | 19.63% | 47.7 | **83.1** | 54.2 | 23h52m | 33.16% | 71.8 | 88.4 | 71.4 | 41h26m | 40.09% | **26.5** | 87.8 | 30.5 | 34h41m | 52.28% |
| AdaPO w/ GRPO | 15.6 | **46.7** | **26.7** | 20h26m | | **48.9** | 81.9 | 54.2 | 27h41m | | **72.3** | **90.8** | **73.4** | 40h05m | | 26.3 | **89.8** | **31.0** | 27h37m | |
| TTRL w/ PPO | **18.5** | 46.7 | 23.3 | 20h51m | 18.54% | 48.1 | 83.1 | 55.4 | 28h03m | 33.61% | 71.9 | **91.2** | 73.8 | 51h58m | 44.52% | 25.9 | **87.8** | 29.9 | 43h04m | 50.69% |
| AdaPO w/ PPO | 18.3 | 46.7 | **26.7** | 38h14m | | 48.1 | 83.1 | 55.4 | 33h33m | | **72.6** | 90.8 | 73.8 | 52h16m | | **26.6** | 86.8 | **32.5** | 36h16m | |
| **Llama-3.2-1B-Instruct** | | | | | | | | | | | | | | | | | | | | |
| TTRL w/ GRPO | 3.5 | 23.3 | **6.7** | 14h29m | 36.86% | **14.9** | 51.8 | **18.1** | 20h57m | 53.30% | **30.3** | **66.0** | **30.8** | 36h42m | 26.27% | **26.4** | **82.2** | **28.4** | 17h13m | 53.84% |
| AdaPO w/ GRPO | 3.5 | 23.3 | 3.3 | 15h19m | | 13.7 | 51.8 | 15.7 | 17h46m | | 28.3 | 64.4 | 29.2 | 42h52m | | 25.3 | 81.7 | 27.4 | 13h33m | |

Llama-3.2-1B-Instruct [29]. For test-time SFT experiments, we further employ Llama-3.1-8B-Instruct to ensure robustness across larger parameter scales.

**Baseline Methods.** AdaPO is a model-agnostic, plug-and-play framework designed to enhance test-time training efficiency. We primarily benchmark against **TTRL** [9], the current state-of-the-art test-time reinforcement learning framework. For both AdaPO and TTRL, we integrate with several representative policy optimization algorithms, including **PPO** [13], **GRPO** [14], and Reinforce++ (**RF++**) [22]. Furthermore, we extend our evaluation to a test-time supervised fine-tuning (**TTSFT**) setting to assess the generalizability of our rollout allocation mechanism.

**Implementation Details.** Rollouts are generated with nucleus sampling ($p = 0.95$) at temperature $0.6$. We set the maximum rollout budget to $M = 64$ per instance and the minimum rollout budget to $N \in \{8, 16, 32\}$ ($N = 32$ unless stated otherwise); the batch-sequential variant uses $B_{\text{par}} = 8$ unless stated otherwise. The SPRT parameters are $(\alpha, \beta) = (0.05, 0.05)$, corresponding to a $95\%$ confidence level, and $\gamma = 0.95$. **Evaluation protocol.** Since a practitioner can stop test-time training at any step, every method's Mean@16, Pass@16 and Pass@1 are the maximum over the validation points during training; Pass@1 is the accuracy of one sampled answer per question. We estimate $p_0$ simply as the majority ratio among the first $N$ rollouts, scaled by a degradation factor $\eta = 0.6$ that offsets the overestimation bias of majority-based pseudo-accuracy. Likewise, $m$ is the number of distinct answers among the first $N$ rollouts (at least 5), and both estimates are then frozen. If all rollouts collected so far (at least $N$) agree on a single answer, sampling stops immediately, since no competing candidate exists. Section 4.3 shows that the results are largely insensitive to the bias of $p_0$. For RL, we perform a single policy-gradient step per instance with a KL penalty coefficient of $0.001$; the actor learning rate is $5 \times 10^{-7}$ and, for PPO, the critic learning rate is $9 \times 10^{-6}$. For test-time SFT (TTSFT and AdaPO-SFT), we use AdamW [30] with a learning rate of $2 \times 10^{-5}$. All experiments run on NVIDIA H20 GPUs (96GB): Qwen2.5-Math-1.5B and Llama-3.2-1B-Instruct use one GPU per run, and Qwen2.5-7B and Llama-3.1-8B-Instruct use four GPUs with tensor parallelism. Our code builds on verl [31] and TTRL [9] and is released under Apache-2.0, like both. Licenses: Qwen2.5 models Apache-2.0, Llama models Llama 3.1/3.2 Community Licenses, MATH-500 MIT, GPQA CC BY 4.0; AIME 2024 and AMC 2023 are public competition problems packaged by TTRL.

### 4.2 Main Results

Table 1 summarizes the performance of our proposed AdaPO across various benchmarks and backbone models. Figure 2 shows the training curves, which make a comparison between AdaPO and TTRL across different policy optimization algorithms. Our key findings are detailed below:

**Substantial efficiency gains with comparable or better accuracy.** AdaPO consistently matches or exceeds the accuracy of TTRL baselines across all benchmarks while significantly reducing computational overhead. Across various backbone models (1B to 7B) and algorithms (GRPO, PPO, RF++), AdaPO typically achieves compute savings between 14% and 78%. For instance, on GPQA with Qwen2.5-Math-1.5B (PPO), AdaPO improves Mean@16 from 25.9 to 26.6 while reducing token consumption significantly from 262M to 144M (45.06% reduction). The savings come from stopping early on "easy" instances.

**Generalization across backbone models and algorithms.** We observe that both the lightweight Llama-3.2-1B and the more capable Qwen2.5-7B benefit from AdaPO, with the 1B model showing particularly high token savings (e.g., 78.36% on AMC 2023 with GRPO). More importantly, AdaPO integrates seamlessly with diverse RL algorithms (e.g., GRPO, PPO, and Reinforce++). In all configurations, AdaPO maintains higher accuracy per token, making it a versatile optimization for test-time reinforcement learning.

**Effective adaptability in test-time fine-tuning.** Table 2 compares our AdaPO-SFT against the standard TTSFT baseline. The efficiency gains are most pronounced on challenging reasoning tasks like GPQA, where AdaPO-SFT reduces token consumption by 51.17% for Qwen2.5-Math-1.5B and a staggering 61.89% for Llama-3.1-8B-Instruct. Notably, these savings do not compromise quality, where AdaPO-SFT largely improves Pass@1 on GPQA. This suggests that adaptive stopping can reduce computational waste without degrading the pseudo-labels used for fine-tuning.

**Batch-sequential variant parallelizes while preserving efficiency.** The sequential variant is the most token-efficient, but generating one rollout at a time underuses the GPU. The batch-sequential variant instead generates $B_{\text{par}}$ rollouts per prompt in each round for all unfinished prompts in parallel, and updates each prompt's stopping state only after the round (Algorithm 1); rollouts generated after the stopping point within a round are the price paid for parallelism. Table 3 ($B_{\text{par}} = 8$) shows that it

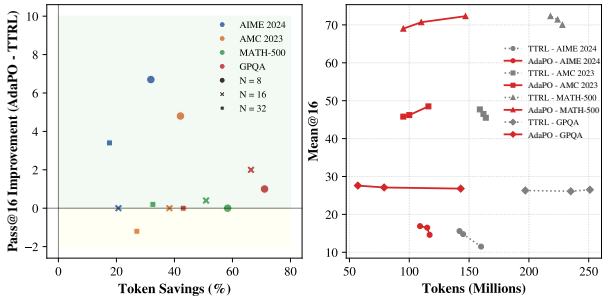


Figure 3: Performance of AdaPO and TTRL under minimum number of rollouts $N \in \{8, 16, 32\}$ (with Qwen2.5-Math-1.5B and GRPO for optimization).

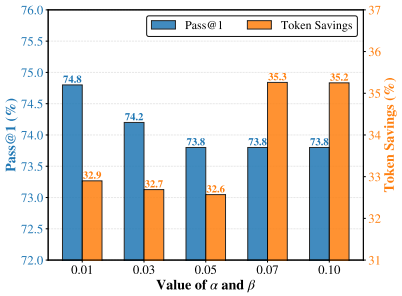


Figure 4: Sensitivity of AdaPO to error guarantees $\alpha$ and $\beta$ on the MATH-500 benchmark.

retains substantial token savings while remaining competitive in accuracy. Wall-clock time matters most on large datasets, and there AdaPO is faster than TTRL: on GPQA it saves 20.37% (GRPO) and 15.79% (PPO) of the time for Qwen2.5-Math-1.5B and 21.30% for Llama-3.2-1B, with 50–54% token savings, and on MATH-500 with GRPO it saves 3.26% of the time and 40.09% of the tokens. $B_{\text{par}}$ trades tokens for time: a larger round keeps the GPU busier at the cost of more rollouts beyond the stopping point. In the two MATH-500 settings where $B_{\text{par}} = 8$ is slower than TTRL, a larger round makes AdaPO faster while keeping most of the token savings: with PPO, $B_{\text{par}} = 32$ takes 51h26m vs. 51h58m for TTRL (40.34% fewer tokens, Pass@1 75.0 vs. 73.8), and with Llama-3.2-1B and GRPO, $B_{\text{par}} = 16$ takes 32h04m vs. 36h42m (47.8% fewer tokens, Pass@1 39.8 vs. 30.8). On small datasets such as AIME and AMC, a round contains few prompts, so the time is bounded by the longest rollouts and the batch-sequential variant does not reduce wall-clock time.

**Comparison with adaptive stopping baselines.** Simply stopping early is not enough. Table 4 plugs three adaptive stopping rules into the same RL loop as AdaPO and TTRL: **LRU** stops once the leader has remained unchanged for 5 rollouts, **VME** stops once the vote margin reaches 5, and **ASC** [20] stops when its posterior confidence reaches 0.95, all with the same budget $M = 64$. These rules cut tokens aggressively but lose accuracy, whereas AdaPO saves tokens while matching or improving on TTRL. The same pattern appears for the voting rules in isolation, without RL (Appendix C).

Table 4: AdaPO vs. adaptive stopping baselines in the RL loop (Qwen2.5-Math-1.5B, GRPO). P@1: Pass@1; Tok.: rollout tokens.

| | AMC 2023 | | MATH-500 | | GPQA | |
|---|---|---|---|---|---|---|
| Method | P@1 | Tok. | P@1 | Tok. | P@1 | Tok. |
| TTRL | **54.2** | 159M | 73.6 | 218M | 30.5 | 251M |
| LRU+RL | 50.6 | 24M | 71.0 | 31M | 29.4 | 26M |
| VME+RL | 53.0 | 97M | 71.0 | 85M | 30.5 | 39M |
| ASC+RL | 53.0 | 102M | 70.4 | 98M | 31.5 | 42M |
| AdaPO | **54.2** | 116M | **73.8** | 147M | **34.5** | 143M |

**Takeaways.** (1) Adaptive early stopping through BSPRT improves efficiency without sacrificing accuracy; (2) AdaPO improves the accuracy–compute trade-off of test-time training; and (3) AdaPO-SFT is a lightweight alternative when full RL-style updates are too expensive.

### 4.3 Analyses and Discussions

To further evaluate the robustness of AdaPO, we conduct an in-depth analysis using Qwen2.5-Math-1.5B as the backbone model. Our analytical setup and key findings are as follows:

**Impact of the minimum rollout budget $N$.** We examine the performance of TTRL and AdaPO across different minimum rollout budgets $N \in \{8, 16, 32\}$. The results are reported in Figure 3. As $N$ decreases, the efficiency gains of AdaPO become more remarkable. For instance, at $N = 16$, AdaPO achieves a substantial token saving of 17.25% on AIME2024 and 26.93% on AMC 2023 compared to TTRL. When reducing $N$ to 8, AdaPO achieves token savings of 32.07% and 42.24%, respectively. These results demonstrate that AdaPO is robust to preserve high-quality reasoning performance even under extreme computational constraints, where the TTRL baseline typically struggles.

**Sensitivity analysis on the error guarantees $\alpha$ and $\beta$.** Figure 4 shows the performance of AdaPO under different error guarantees on the MATH-500 benchmark. The results demonstrate that AdaPO

Table 5: Sensitivity of AdaPO to the degradation factor $\eta \in [0.4, 1.0]$ on AMC 2023.

| $\eta$ | Mean@16 | Pass@16 | Pass@1 | Toks. | Toks. Saving |
|---|---|---|---|---|---|
| 0.4 | 48.3 | 81.9 | 54.2 | 121M | 23.99% |
| 0.5 | 48.6 | 83.1 | 53.0 | 118M | 26.11% |
| 0.6 | 48.5 | 81.9 | 54.2 | 116M | 26.93% |
| 0.7 | 47.7 | 83.1 | 54.2 | 114M | 28.72% |
| 0.8 | 47.1 | 80.7 | 53.0 | 111M | 30.09% |
| 0.9 | 49.2 | 83.1 | 55.4 | 111M | 30.49% |
| 1.0 | 47.9 | 81.9 | 56.6 | 109M | 31.63% |

Table 6: Sensitivity of AdaPO to the confidence $\gamma$ of the *Confirming* stage on AMC 2023.

| $\gamma$ | Pass@1 | Toks. | Toks. Saving | Confirm Round |
|---|---|---|---|---|
| 0.900 | 49.4 | 85.8M | 47.79% | 1.005 |
| 0.950 | 51.8 | 85.6M | 47.87% | 1.005 |
| 0.960 | 51.8 | 88.3M | 46.21% | 2.013 |
| 0.965 | 53.0 | 88.0M | 46.44% | 2.042 |
| 0.970 | 51.8 | 88.8M | 45.96% | 2.083 |
| 0.975 | 51.8 | 88.2M | 46.32% | 2.175 |
| 0.980 | 54.2 | 90.3M | 45.02% | 2.499 |
| 0.985 | 51.8 | 90.4M | 44.94% | 2.940 |
| 0.990 | 53.0 | 90.4M | 44.93% | 3.323 |
| 0.995 | 54.2 | 91.3M | 44.40% | 4.507 |

maintains a remarkably stable and high Pass@1 accuracy across a wide range of error tolerances. As the error budgets are relaxed from 0.01 to 0.05, we observe a slight trade-off in accuracy (from 74.8% to 73.8%) while maintaining consistent efficiency. Notably, further increasing the error tolerance to 0.07 and 0.10 yields a significant boost in token savings (reaching 35.3%) without any further degradation in Pass@1 performance, so AdaPO still prunes redundant rollouts with looser guarantees.

**Sensitivity analysis on the degradation factor $\eta$.** We vary the degradation factor $\eta \in [0.4, 1.0]$ to evaluate the robustness of AdaPO across different initial $p_0$ estimation configurations. Table 5 shows that AdaPO is notably insensitive to variations in $\eta$. While token savings scale monotonically from 23.99% to 31.63% as $\eta$ increases, accuracy metrics (Mean@16, Pass@16, and Pass@1) remain remarkably stable with only marginal fluctuations. The slight increase in token consumption at lower $\eta$ values is expected, as a more conservative initial belief necessitates more evidence during hypothesis testing. AdaPO therefore does not depend on a precise estimate of $p_0$.

**Sensitivity analysis on confidence factor $\gamma$ and confirmation round $Z$.** Table 6 evaluates AdaPO's sensitivity to the posterior confidence level $\gamma$ during the *Confirming* state on the AMC 2023 benchmark. The results highlight the trade-off between rigorous verification and computational efficiency, where higher values of $\gamma$ (e.g., 0.990 and 0.995) ensure greater reliability but necessitate more extensive rollout evidence. At $\gamma \leq 0.95$ we have $Z^\star(\gamma) = 1$, so the *Confirming* stage adds essentially no rollouts (1.005 rounds on average); raising $\gamma$ to 0.995 increases the average to 4.5 rounds and the tokens by 6.7% (85.6M to 91.3M), with Pass@1 between 51.8 and 54.2 for all $\gamma \geq 0.95$. For the variant utilizing a fixed number of confirmation rounds $Z$, please refer to Appendix D for detailed analysis.

**Robustness under model misspecification.** Appendix B shows that AdaPO keeps a negligible accuracy gap and strong agreement with full-budget voting when the symmetric noise model is violated, remains stable across rollout-correlation levels induced by the sampling temperature, and is insensitive to moderate misspecification of $p_0$, as the stopping threshold is piecewise constant in $p_0$.

**Limitations and Future Work.** We do not model the exact rollout distribution, which would make the decision rule intractable; AdaPO instead relies on the empirical observation that rollout distributions are dominated by a few top candidates. Relaxing our modeling assumptions while keeping the rule tractable is a challenging theoretical direction. Our estimates of $p_0$ and $m$ from the first $N$ rollouts are simple heuristics, and better estimators may further reduce cost. This work optimizes majority-voting-based TTT; Appendix G outlines how AdaPO extends to continuous signals from an external reward model.

## 5 Conclusion

In this paper, we present AdaPO, a principled framework that addresses the inherent inefficiency of rollout allocation in test-time training. By casting reward estimation as a deterministic two-stage voting procedure with an SPRT-induced trigger rule and a posterior-threshold confirmation rule, AdaPO improves heuristic majority voting with an adaptive, evidence-driven mechanism. Our theory identifies the conditions under which the electing-stage trigger reduces to the classical SPRT, and experiments across diverse reasoning benchmarks and backbones show that AdaPO achieves significant compute savings (e.g., over 40% fewer rollout tokens on GPQA) without sacrificing accuracy. We hope this work encourages further research on scalable LLM adaptation at test time.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62372314, the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. T41-517/25-N), and HK PolyU Research Grant (No. A0064486). The experimental part of this work was supported by The Centre for Large AI Models (CLAIM) of The Hong Kong Polytechnic University. We also thank Ren Da (PolyU COMP), Marco Bettoni (PolyU Industrial Center), and the PolyU Industrial Center for their support.

## References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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

[4] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[5] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022.

[6] 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.

[7] Jonas Hübotter, Sascha Bongni, Ido Hakimi, and Andreas Krause. Efficiently learning at test-time: Active fine-tuning of llms. In *The Thirteenth International Conference on Learning Representations*, 2025.

[8] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025.

[9] Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. TTRL: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

[10] Abraham Wald. Sequential tests of statistical hypotheses. In *Breakthroughs in statistics: Foundations and basic theory*, pages 256–298. Springer, 1992.

[11] Xiao-Yong Wei and Zhen-Qun Yang. Mining in-class social networks for large-scale pedagogical analysis. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 639–648, 2012.

[12] Xiao-Yong Wei and Zhen-Qun Yang. Coaching the exploration and exploitation in active learning for interactive video retrieval. *IEEE transactions on image processing*, 22(3):955–968, 2012.

[13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[14] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[15] Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. TTT++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34:21808–21820, 2021.

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

[17] Yanyang Li, Michael R Lyu, and Liwei Wang. Learning to reason from feedback at test-time. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5241–5253, 2025.

[18] Dulhan Jayalath, Shashwat Goel, Thomas Foster, Parag Jain, Suchin Gururangan, Cheng Zhang, Anirudh Goyal, and Alan Schelten. Compute as teacher: Turning inference compute into reference-free supervision. *arXiv preprint arXiv:2509.14234*, 2025.

[19] Youkang Wang, Jian Wang, Rubing Chen, Xiao-Yong Wei, and Qing Li. Probabilistic optimality for inference-time scaling. *arXiv e-prints*, pages arXiv–2506, 2025.

[20] Pranjal Aggarwal, Aman Madaan, Yiming Yang, et al. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. *arXiv preprint arXiv:2305.11860*, 2023.

[21] Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. *arXiv preprint arXiv:2401.10480*, 2024.

[22] Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

[23] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[24] MAA. American invitational mathematics examination - aime. `https://huggingface.co/datasets/AI-MO/aimo-validation-aime`, February 2024.

[25] MAA. American Mathematics Competitions (AMC) 2023. `https://huggingface.co/datasets/math-ai/amc23`, 2023.

[26] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[27] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[28] Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL `https://qwenlm.github.io/blog/qwen2.5/`.

[29] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[31] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, 2025.

[32] Xinglin Wang, Shaoxiong Feng, Yiwei Li, Peiwen Yuan, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. Make every penny count: Difficulty-adaptive self-consistency for cost-efficient reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6904–6917, 2025.

# A Detailed Algorithmic Pseudocode

**Algorithm 2:** AdaPO: Pseudocode for Sequential and Batch-Sequential Variants

**Input:** prompts $\{x_p\}$ (single $x$ for sequential), policy $\pi_\theta$, extractor $\mathcal{E}$, $M$, $N$, $\alpha$, $\beta$, $\gamma$, $\eta$; *batch-seq. only:* parallel batch size $B_{\text{par}}$; *update type:* RL or SFT

**Output:** pseudo-label(s) $a_p^\star$; retained rollouts; updated policy if training is applied

*// Subscript p indexes prompts; for the single-prompt sequential case, it can be omitted.*

**Fixed constants:**

$A \leftarrow \frac{1-\beta}{\alpha}$; $B \leftarrow \frac{\beta}{1-\alpha}$

**Init** per-prompt state $\mathcal{S}_p \leftarrow (\Phi_p = \textsc{Elc},\ V_p = \mathbf{0},\ Z_p = 0,\ c_p = \varnothing,\ n_p = 0)$

**Subroutine** $\textsc{Update}(a_t, \mathcal{S}_p)$:

- $V_p(a_t)$ += 1;
- $n_p$ += 1;
- $a_{L,p} \leftarrow \arg\max_a V_p(a)$, $a_{R,p} \leftarrow \arg\max_{a \neq a_{L,p}} V_p(a)$;
- $\Delta \leftarrow V_p(a_{L,p}) - V_p(a_{R,p})$ ▷ *vote gap between leader and runner-up*
- **if** $n_p = N$ **then** ▷ *estimated once from the initial $N$ rollouts, then frozen*
  - $\tilde{p}_{0,p} \leftarrow \max_a V_p(a)/n_p$;
  - $\hat{p}_{0,p} \leftarrow \eta\, \tilde{p}_{0,p}$;
  - $m_p \leftarrow \max(|\{a : V_p(a) > 0\}|, 5)$
- **end**
- **if** $n_p \geq N$ **then**
  - $\kappa_p \leftarrow \frac{\hat{p}_{0,p}(m_p - 1)}{1 - \hat{p}_{0,p}}$;
  - $\Delta_{A,p} \leftarrow \lceil \frac{\log A}{\log \kappa_p} \rceil$; $\Delta_{B,p} \leftarrow \lfloor \frac{\log B}{\log \kappa_p} \rfloor$;
  - $Z_p^\star \leftarrow \lceil 1 + \frac{\log \frac{\gamma}{1-\gamma} - \log A}{\log \kappa_p} \rceil$
- **end**
- **if** $\Phi_p = \textsc{Elc}$ **then** ▷ *Electing*
  - **if** $n_p \geq N$ **and** $\Delta \geq \Delta_{A,p}$ **then** $\Phi_p \leftarrow \textsc{Cfm},\ c_p \leftarrow a_{L,p},\ Z_p \leftarrow 1$
- **else if** $\Phi_p = \textsc{Cfm}$ **then** ▷ *Confirming*
  - $\Delta_c \leftarrow V_p(c_p) - \max_{a \neq c_p} V_p(a)$
  - **if** $\Delta_c \leq \Delta_{B,p}$ **then**
    - $\Phi_p \leftarrow \textsc{Elc},\ Z_p \leftarrow 0$
  - **else if** $\Delta_c \geq \Delta_{A,p}$ **then**
    - $Z_p$ += 1
- **if** $\Phi_p = \textsc{Cfm}$ **and** $Z_p \geq Z_p^\star$ **then** $\Phi_p \leftarrow \textsc{End}$
- **return** $\mathcal{S}_p, a_{L,p}, \mathsf{done}_p \leftarrow [\Phi_p = \textsc{End}]$

**[Sequential Variant]** ▷ *single prompt; stops exactly when BSPRT fires*

- **while** $n_p < M$ **and not** *done* **do**
  - sample $y \sim \pi_\theta(\cdot|x)$;
  - $\mathcal{S}, a_L, \mathsf{done} \leftarrow \textsc{Update}(\mathcal{E}(y), \mathcal{S})$
- **end**

**[Batch-sequential Variant]** ▷ *prompt batch; stops per-prompt after each vLLM round*

- $\mathcal{P} \leftarrow \{\text{all prompts}\}$
- **while** $\mathcal{P} \neq \emptyset$ **do**
  - **for** $p \in \mathcal{P}$: allocate $s_p = \min(B_{\text{par}}, M - n_p)$ slots for $x_p$
  - $\mathbf{Y} \leftarrow \pi_\theta$`.generate`$([\, x_p \times s_p : p \in \mathcal{P}\,])$ ▷ *one parallel call*
  - **for** $p \in \mathcal{P}$:
    - **for** $y \in \mathbf{Y}_p$:
    - $\mathcal{S}_p, a_{L,p}, \mathsf{done}_p \leftarrow \textsc{Update}(\mathcal{E}(y), \mathcal{S}_p)$;
    - **if** $\mathsf{done}_p$ **then break**
    - **if** $\mathsf{done}_p$ **or** $n_p \geq M$ **then** remove $p$ from $\mathcal{P}$
  - **end**

**Finalise per prompt:**

- $a_p^\star \leftarrow a_{L,p}$ if $\mathsf{done}_p$, else $\arg\max_a V_p(a)$ ▷ *budget-exhausted fallback*
- retain rollouts collected up to stopping (or budget $M$)
- **if** *update type = RL* **then**
  - assign $R(y_i; a_p^\star) = \mathbf{1}[\mathcal{E}(y_i) = a_p^\star]$ and update $\theta$ with PPO/GRPO/RF++
- **else if** *update type = SFT* **then**
  - update $\theta$ with $\mathcal{L}(\theta) = -\frac{1}{P} \sum_p \log \pi_\theta(a_p^\star \mid x_p)$

For completeness, we provide the full pseudocode for AdaPO in both the **Sequential** and **Batch-sequential** settings. The listing makes explicit the per-prompt state variables, the *Electing* / *Confirming* transitions used by the underlying stopping rule, and the distinction between rollout collection and downstream policy updates. This presentation aims to complement the main-text description by giving a precise procedural reference for implementation.

# B Robustness Analysis to Model Misspecification

## B.1 Does AdaPO break when the symmetric categorical noise assumption fails?

The electing-stage trigger rule (Proposition 3.1) is derived under a symmetric categorical noise model, in which each non-majority answer receives equal probability $q_0 = \frac{1-p_0}{m-1}$. In practice, rollout distributions are highly asymmetric: a single dominant incorrect answer may attract far more probability mass than the remaining alternatives. A natural concern is whether this model misspecification degrades AdaPO's performance. We conduct a systematic empirical study demonstrating that AdaPO is robust to severe violations of the symmetry assumption.

**Setup.** We analyze rollout distributions from GPQA [26] with $M = 32$, Qwen2.5-Math-1.5B backbone, across 15 training epochs (2,880 data instances total, i.e., 192 questions' rollout distribution $\times$ 15 epochs; 1,861 non-unanimous).

**Skew metric.** For each question-instance with observed vote counts, let $V_1 = V(a_L)$, $V_2 = V(a_R)$, and $V_{\text{rest}} = \sum_{a \notin \{a_L, a_R\}} V(a)$ denote the leader, runner-up, and remaining vote counts, respectively, and let $k = m - 2$ be the number of non-top-2 candidates. We define the *skew metric*

$$s \;=\; \frac{V_2}{V_{\text{rest}}/k},$$

which compares the runner-up's vote share to the average share of the remaining candidates. Under the symmetric noise model ($q_0 = \frac{1-p_0}{m-1}$), we have $s = 1$ by symmetry. Values $s \gg 1$ indicate a dominant incorrect attractor, i.e., the misspecification regime the theory does not cover. Instances with exactly two observed answers ($k = 0$) correspond to $s = \infty$, where the top-2 reduction is exact by construction.

**Real distributions are substantially asymmetric.** Figure 5 shows the empirical distribution of the skew metric across all non-trivial instances with $m \geq 3$. The mean skew is $\bar{s} = 3.76$ with standard deviation 3.93; the median is 2.25. Notably, 66.5% of instances exhibit $s \geq 2$ and 23.1% exhibit $s \geq 5$, confirming that the symmetric noise assumption is substantially violated in practice.

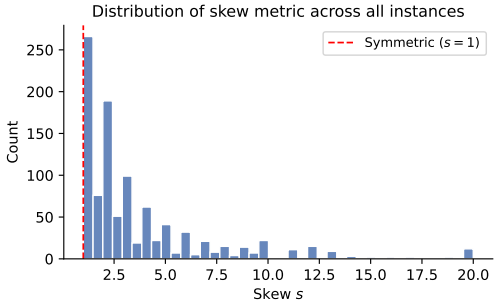


Figure 5: Distribution of the skew metric $s$ across all non-unanimous instances with $m \geq 3$ (1,022 instances). The dashed line marks $s = 1$, corresponding to the symmetric noise assumption. The majority of instances fall well above $s = 1$, confirming asymmetry in real rollout distributions.

**AdaPO is robust across all skew levels.** To evaluate whether this misspecification harms AdaPO, we simulate its sequential BSPRT stopping rule on each question-instance by drawing 200 random vote orderings consistent with the observed distribution and applying the electing and confirming

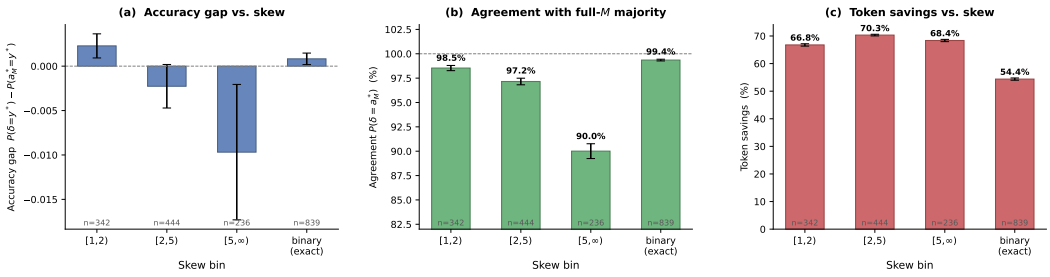


Figure 6: Robustness of AdaPO across skew levels. **(a)** Accuracy gap between AdaPO and full-$M$ majority voting remains near zero across all bins. **(b)** Agreement with the full-budget majority answer stays above 90%, even under severe misspecification ($s \geq 5$). **(c)** Token savings are maintained at 54–70% regardless of skew level. Error bars show standard error of the mean.

stages with default parameters ($\alpha = \beta = 0.05$, $\gamma = 0.95$). We measure three quantities across skew strata (Table 7, Figure 6):

- **Accuracy gap** $\big(P(\delta = y^*) - P(a_M^* = y^*)\big)$: the difference in ground-truth accuracy between AdaPO's early selection $\delta$ and the full-$M$ majority vote $a_M^*$. A gap near zero means AdaPO does not sacrifice accuracy relative to exhaustive voting.
- **Agreement** $P(\delta = a_M^*)$: how often AdaPO selects the same answer as the full-$M$ majority vote.
- **Token savings** $1 - \mathbb{E}[\tau]/M$: the fraction of rollout budget saved by early stopping.

Table 7: Robustness of AdaPO across skew levels. Skew $s = V_2/(V_{\text{rest}}/k)$ measures deviation from the symmetric categorical noise assumption ($s = 1$). Agreement and accuracy gap are computed via Monte Carlo simulation over 200 random vote orderings per instance.

| Skew bin | $n$ | Accuracy gap | Agreement (%) | Token savings (%) | Top-2 mass |
|---|---|---|---|---|---|
| $[1,\ 2)$ | 342 | $+0.002$ | 98.5 | 66.8 | 0.855 |
| $[2,\ 5)$ | 444 | $-0.002$ | 97.2 | 70.3 | 0.858 |
| $[5,\ \infty)$ | 236 | $-0.010$ | 90.0 | 68.4 | 0.905 |
| Binary (exact) | 839 | $+0.001$ | 99.4 | 54.4 | 1.000 |

The results are clear-cut. **Accuracy gap** is negligible across all bins: $|\text{gap}| < 0.01$ even under extreme skew ($s \geq 5$). **Agreement** with the full-$M$ majority exceeds 97% for low-to-moderate skew and remains above 90% even in the most misspecified regime. **Token savings** are consistently large (54–70%) and do *not* collapse under asymmetry, refuting the concern that efficiency gains are limited to symmetric (easy) cases.

**Why the top-2 reduction still holds.** The theoretical derivation reduces multi-class voting to a binary test between the leader $a_L$ and runner-up $a_R$. Two empirical observations justify this reduction even under asymmetry. First, for instances with $m \geq 3$, the top-2 mass $(V_1 + V_2)/M$ exceeds 85% on average across all skew bins (Table 7), indicating that the race for the majority vote is indeed dominated by two candidates. Second, and perhaps counter-intuitively, higher skew actually *strengthens* the top-2 reduction: when one incorrect answer dominates the remaining alternatives, the multi-class problem collapses to an effective binary competition, which is exactly the scenario the BSPRT is designed for.

**Interpretation: why misspecification does not hurt.** The key insight is that the BSPRT trigger condition $\Delta_t \geq \lceil \log A / \log \kappa \rceil$ depends only on the *vote gap* between the leader and runner-up—it does not require the remaining candidates to share equal probability. The symmetric noise assumption is used to derive the threshold in closed form, but the resulting decision rule is robust because:

1. **Under high skew** ($s \gg 1$): a single dominant incorrect answer concentrates all competition into the $(a_L, a_R)$ pair. The top-2 reduction is *exactly correct* in this regime, and the vote gap grows rapidly, leading to fast and reliable stopping.

2. **Under low skew** ($s \approx 1$): the distribution matches the symmetric assumption, and the theoretical guarantees apply directly.
3. **Overall**: the worst-case misspecification scenario (moderate skew with multiple competitors) accounts for a minority of instances, and even there, the agreement rate exceeds 97%.

Altogether, the empirical evidence supports that the symmetric categorical noise model, while not a faithful description of real rollout distributions, yields a stopping rule that is robust to the forms of misspecification encountered in practice. The overall agreement rate across all non-trivial instances is 97.5%, and the accuracy gap is $-0.001$, confirming that AdaPO's early stopping does not compromise the quality of the selected pseudo-label.

### B.2 Does the correlation between rollouts affect AdaPO's performance?

Table 8: AdaPO's performance under various rollout correlation levels (controlled by temperature $t$).

| | **Llama-3.2-1B-Instruct + MATH500** | | | | | **Qwen-2.5-Math-1.5B + AIME 2024** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $t$ | Mean@16 | Pass@16 | Pass@1 | Toks. | Toks. Saving | Mean@16 | Pass@16 | Pass@1 | Toks. | Toks. Saving |
| 0.2 | 28.8 | 60.0 | 29.2 | 126M | 30.37% | 12.7 | **46.7** | 23.3 | 117M | 17.61% |
| 0.4 | **30.5** | 62.2 | 31.0 | 113M | 37.57% | **16.5** | **46.7** | 23.3 | 115M | 19.01% |
| 0.6 | 28.3 | 64.4 | 29.2 | 133M | 26.52% | 15.6 | **46.7** | **26.7** | 113M | **20.42%** |
| 0.8 | 29.9 | 63.6 | **31.2** | 111M | **38.67%** | 15.8 | **46.7** | 20.0 | 114M | 19.72% |
| TTRL (0.6) | 30.3 | **66.0** | 30.8 | 181M | 0% | 15.6 | 43.3 | 20.0 | 142M | 0% |

LLM rollouts are typically not independent: low-temperature decoding tends to produce highly correlated samples, while higher temperatures induce greater diversity and thus weaker correlation. To empirically probe whether AdaPO is sensitive to violation of i.i.d. assumption, we vary the sampling temperature $t$ to simulate different correlation levels, with lower $t$ corresponding to more correlated rollouts. Table 8 shows that AdaPO remains robust across this range. On Llama-3.2-1B-Instruct + MATH500, performance varies only moderately across temperatures, while token savings remain substantial (26.52%–38.67%) and even peak at higher temperature. On Qwen-2.5-Math-1.5B + AIME 2024, AdaPO is similarly stable: Mean@16 stays in a narrow range, Pass@16 is consistently strong, and token savings remain around 18%–20% across all temperatures. Compared with the TTRL baseline at $t = 0.6$, AdaPO reduces token usage at every tested temperature on both settings while maintaining comparable, and sometimes better, accuracy. Overall, these results suggest that although LLM rollouts are indeed correlated, AdaPO does not rely critically on independence and continues to deliver favorable quality–efficiency trade-offs under different correlation levels.

### B.3 Does misspecified $p_0$ harm AdaPO's results?

AdaPO is known to be insensitive to misspecification of $p_0$, as also reflected by the ablation results regarding degradation facter $\eta$ (Table 5) in § 4.3. Here we provide an intuitive explanation for why the method remains robust when the estimated majority probability is biased.

Figure 7 plots the triggering threshold $\Delta_A$ as a function of $p_0$. As $p_0$ increases, $\kappa = \frac{p_0(m-1)}{1-p_0}$ also increases, so the required threshold $\Delta_A$ decreases only in a piecewise-constant manner after the ceiling operation. This means that a moderate perturbation in the estimate of $p_0$ often leaves the induced BSPRT threshold unchanged.

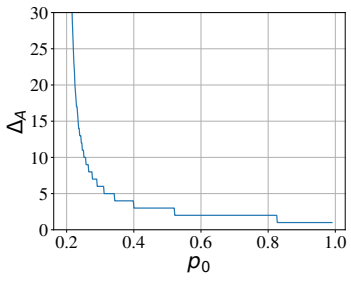


Figure 7: Function of $\Delta_A$ versus $p_0$, where $\Delta_A := \left\lceil \frac{\log A}{\log \kappa} \right\rceil$.

Equivalently, the stopping rule is locally stable with respect to small estimation errors: even if the estimated success probability shifts, the resulting integer boundary may remain the same over a substantial interval of $p_0$ values. Consequently, moderate bias in estimating $p_0$ often does not change the induced stopping threshold at all, which helps explain the empirical robustness of AdaPO to misspecified $p_0$.

Table 9: RL results under the same settings as TTRL and AdaPO on Qwen2.5-Math-1.5B + GRPO.

| | AMC 2023 | | | | MATH500 | | | | GPQA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Mean@16 | Pass@16 | Pass@1 | Toks. | Mean@16 | Pass@16 | Pass@1 | Toks. | Mean@16 | Pass@16 | Pass@1 | Toks. |
| **Qwen-2.5-Math-1.5B** | | | | | | | | | | | | |
| TTRL | 47.7 | **83.1** | **54.2** | 159M | **72.3** | 90.6 | 73.6 | 218M | 26.5 | **87.8** | 30.5 | 251M |
| AdaPO (Ours) | **48.5** | 81.9 | **54.2** | 116M | **72.3** | 90.8 | **73.8** | 147M | 26.8 | **87.8** | **34.5** | 143M |
| LRU ($S = 5$) + RL | 44.4 | 81.9 | 50.6 | 24M | 69.0 | 90.4 | 71.0 | 31M | 26.7 | 87.3 | 29.4 | 26M |
| VME ($T = 5$) + RL | 46.1 | 81.9 | 53.0 | 97M | 69.0 | **91.4** | 71.0 | 85M | **27.0** | 86.8 | 30.5 | 39M |

# C Comparison with Manually-curated Baselines

## C.1 Simpler Adaptive Rollout Allocation Baselines for TTT

While there is currently no efficient TTT baseline that directly addresses the same rollout-allocation problem as AdaPO, we construct two simple adaptive baselines beyond the original TTRL to provide a clearer study of the effectiveness of the proposed voting mechanism for AdaPO:

(1) **LRU** (**L**eader **R**emain **U**nchanged): stop when the leader remains unchanged for a fixed number of steps $S$; and (2) **VME** (**V**ote **M**argin **E**xceedance): stop when the vote margin reaches a predefined threshold $T$.

We run test-time policy update (RL) with LRU and VME voting mechanisms under the same settings as TTRL and AdaPO. The results are shown in Table 9; Table 4 in the main text additionally includes ASC [20] inside the RL loop (confidence threshold $0.95$, $M = 64$).

The RL results suggest that simple adaptive heuristics can sometimes be competitive on individual metrics, but their behavior is much less balanced than AdaPO. In particular, LRU and VME often stops prematurely, achieving very low token consumptions while severely degrading accuracies. When they occasionally attain strong scores on a subset of tasks, they often do so with inconsistent trade-offs across datasets and sometimes with substantially larger token usage. By contrast, AdaPO delivers a more stable quality–efficiency profile across both backbones: it remains competitive in accuracy while consistently reducing token consumption relative to TTRL. This supports the view that AdaPO's advantage is not merely due to using an adaptive stopping rule, but to using a better-calibrated voting mechanism for controlling rollout allocation.

## C.2 Experiment Results: Voting Mechanism in Isolation

In addition to the full RL experiments above, we also compare the underlying pure voting strategies in isolation: majority voting (used in TTRL), the voting mechanism in AdaPO, **LRU** for $T = 5$ steps **VME** with a fixed gap threshold $S = 5$, together with three recent voting mechanism proposed in test-time scaling literature: DSC (Difficulty-Adaptive Self-Consistency) [32], ASC (Adaptive Self-Consistency) [20], and ESC (Early-stopping Self-Consistency) [21]. We isolate the voting mechanism from the RL process by directly prompting the model for multi-sample rollout voting, based on the stopping rule of each method. We run experiments on 3 models with different maximum sample sizes $\max_N$. Here we report results for $\max_N = 16$ and $64$. The results are shown in Table 10.

The voting-only comparison makes the pattern even clearer. Across the stronger backbones, AdaPO voting is typically the best or tied-best method in accuracy, while also using far fewer tokens than full majority voting and remaining competitive with more aggressive early-stopping baselines. LRU and VME can reduce compute substantially, but they are less reliable: their gains are often accompanied by noticeable drops on some benchmarks, indicating that fixed heuristic criteria do not adapt well to changing uncertainty across questions. The comparison with DSC, ASC, and ESC further shows that AdaPO's advantage is not simply early stopping per se; rather, its sequential decision rule yields a stronger accuracy–efficiency trade-off across both smaller and larger sample budgets.

Table 10: Voting mechanism in isolation results without RL across three backbone models. For each model, rows are grouped by maximum sample size (for $\max_N = 16$ and $\max_N = 64$).

| | AIME24 | | AIME25 | | AMC23 | | MATH500 | | GSM8K | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | Acc. | Toks. | Acc. | Toks. | Acc. | Toks. | Acc. | Toks. | Acc. | Toks. |
| **Deepseek-R1-Distill-Llama-8B** | | | | | | | | | | |
| Majority Voting ($\max_N = 16$) | 50.00% | 161,285 | 23.33% | 120,587 | 80.00% | 72,769 | 90.20% | 52,784 | 84.61% | 17,799 |
| DSC ($\max_N = 16$) | 50.00% | 134,750 | 13.33% | **19,566** | 80.00% | 37,476 | 89.40% | 30,491 | 84.53% | 11,183 |
| ESC ($\max_N = 16$) | 50.00% | 136,123 | 23.33% | 66,133 | 80.00% | 51,119 | 90.20% | 38,102 | 84.61% | 13,600 |
| ASC ($\max_N = 16$) | 50.00% | 127,121 | 23.33% | 53,738 | 80.00% | 34,899 | 90.00% | 30,707 | 84.53% | 11,155 |
| LRU ($\max_N = 16$) | 53.33% | **85,784** | 23.33% | 53,098 | 82.50% | 34,268 | 88.60% | 25,530 | 82.87% | 8,583 |
| VME ($\max_N = 16$) | 50.00% | 115,472 | 23.33% | 37,230 | 85.00% | **17,599** | 89.80% | 21,582 | 83.78% | 7,364 |
| AdaPO voting ($\max_N = 16$) | **66.67%** | 98,756 | **30.00%** | 79,063 | **90.00%** | 25,878 | **92.80%** | **17,522** | **92.04%** | **5,216** |
| Majority Voting ($\max_N = 64$) | 56.67% | 652,661 | 23.33% | 476,857 | 80.00% | 287,408 | 91.80% | 210,630 | 87.49% | 71,237 |
| DSC ($\max_N = 64$) | 56.67% | 350,599 | 23.33% | 93,539 | 80.00% | 64,658 | 91.00% | 57,752 | 86.88% | 20,846 |
| ESC ($\max_N = 64$) | 56.67% | 450,976 | 23.33% | 124,966 | 80.00% | 116,284 | 91.40% | 91,497 | 87.19% | 32,855 |
| ASC ($\max_N = 64$) | 56.67% | 338,790 | 23.33% | 93,775 | 80.00% | 60,750 | 91.40% | 57,899 | 86.88% | 20,805 |
| LRU ($\max_N = 64$) | 53.33% | **86,387** | 23.33% | **53,098** | 82.50% | 34,470 | 88.60% | 25,774 | 82.94% | 8,596 |
| VME ($\max_N = 64$) | 56.67% | 383,487 | 23.33% | 77,777 | 85.00% | **26,566** | 90.80% | 54,104 | 85.06% | 17,783 |
| AdaPO voting ($\max_N = 64$) | **66.67%** | 176,340 | **36.67%** | 187,170 | **92.50%** | 34,227 | **93.40%** | **25,769** | **91.81%** | **7,157** |
| **Llama-3.1-8B-Instruct** | | | | | | | | | | |
| Majority Voting ($\max_N = 16$) | **6.67%** | 108,067 | **0.00%** | 105,305 | 30.00% | 31,319 | 59.60% | 19,067 | 88.55% | 4,182 |
| DSC ($\max_N = 16$) | **6.67%** | 95,051 | **0.00%** | 91,174 | 30.00% | 28,698 | 59.20% | 16,009 | 84.38% | 1,772 |
| ESC ($\max_N = 16$) | **6.67%** | 108,067 | **0.00%** | 105,305 | 30.00% | 31,104 | 59.60% | 17,949 | 88.55% | 2,928 |
| ASC ($\max_N = 16$) | **6.67%** | 95,051 | **0.00%** | 91,174 | 30.00% | 28,698 | 59.60% | 15,987 | 88.55% | 2,390 |
| LRU ($\max_N = 16$) | 3.33% | **51,933** | **0.00%** | **51,243** | 25.00% | **16,297** | 57.40% | **9,520** | 87.87% | 2,022 |
| VME ($\max_N = 16$) | **6.67%** | 90,671 | **0.00%** | 85,040 | 30.00% | 26,729 | 59.20% | 14,265 | 88.02% | 1,776 |
| AdaPO voting ($\max_N = 16$) | **6.67%** | 79,246 | **0.00%** | 67,298 | **35.00%** | 22,681 | **64.00%** | 12,401 | **89.92%** | **1,559** |
| Majority Voting ($\max_N = 64$) | 6.67% | 386,783 | **0.00%** | 348,361 | 32.50% | 113,401 | 60.80% | 70,643 | 89.16% | 14,697 |
| DSC ($\max_N = 64$) | 6.67% | 177,707 | **0.00%** | 168,646 | 32.50% | 61,045 | 60.20% | 35,505 | 86.58% | 3,453 |
| ESC ($\max_N = 64$) | 6.67% | 386,783 | **0.00%** | 348,361 | 32.50% | 106,797 | 60.80% | 56,787 | 89.39% | 6,665 |
| ASC ($\max_N = 64$) | 6.67% | 177,707 | **0.00%** | 168,646 | 32.50% | 61,045 | 60.80% | 35,464 | 89.31% | 4,028 |
| LRU ($\max_N = 64$) | 3.33% | **51,933** | **0.00%** | **51,243** | 25.00% | **16,352** | 57.40% | **9,531** | 87.87% | **2,030** |
| VME ($\max_N = 64$) | 6.67% | 285,351 | **0.00%** | 264,136 | 30.00% | 86,964 | 60.20% | 45,584 | 88.17% | 3,877 |
| AdaPO voting ($\max_N = 64$) | **10.00%** | 247,196 | **0.00%** | 189,324 | **47.50%** | 56,345 | **65.60%** | 31,034 | **89.92%** | 2,112 |
| **Qwen3-8B** | | | | | | | | | | |
| Majority Voting ($\max_N = 16$) | 83.33% | 215,354 | 70.00% | 251,192 | 85.00% | 109,692 | 89.60% | 65,062 | 95.45% | 27,268 |
| DSC ($\max_N = 16$) | 83.33% | 104,034 | 70.00% | 108,489 | 87.50% | 40,619 | 89.60% | 20,325 | 95.38% | **4,806** |
| ESC ($\max_N = 16$) | 83.33% | 132,703 | 70.00% | 151,447 | 85.00% | 54,344 | 89.60% | 31,656 | 95.38% | 10,941 |
| ASC ($\max_N = 16$) | 83.33% | 98,010 | 70.00% | 105,036 | 87.50% | 38,642 | 89.60% | 24,860 | 95.45% | 8,733 |
| LRU ($\max_N = 16$) | 83.33% | 99,652 | 66.67% | 123,448 | 87.50% | 48,046 | 90.00% | 29,448 | 95.45% | 12,193 |
| VME ($\max_N = 16$) | 83.33% | 62,186 | 66.67% | **75,515** | 87.50% | **26,207** | 89.60% | **16,932** | 95.45% | 6,379 |
| AdaPO voting ($\max_N = 16$) | **86.67%** | **52,506** | **80.00%** | 83,056 | **92.50%** | 30,036 | **94.40%** | 18,470 | **95.75%** | 5,740 |
| Majority Voting ($\max_N = 64$) | 80.00% | 862,151 | 73.33% | 996,858 | 85.00% | 438,448 | 89.20% | 260,304 | 95.68% | 109,154 |
| DSC ($\max_N = 64$) | 80.00% | 193,705 | 73.33% | 174,043 | 87.50% | 47,935 | 89.40% | 33,927 | 95.60% | 7,001 |
| ESC ($\max_N = 64$) | 80.00% | 252,786 | 73.33% | 364,172 | 85.00% | 82,789 | 89.20% | 52,974 | 95.60% | 15,686 |
| ASC ($\max_N = 64$) | 80.00% | 182,606 | 73.33% | 170,936 | 87.50% | 53,329 | 89.40% | 38,463 | **95.75%** | 11,402 |
| LRU ($\max_N = 64$) | 83.33% | 99,652 | 66.67% | 126,535 | 87.50% | 48,046 | 90.00% | 29,483 | 95.45% | 12,240 |
| VME ($\max_N = 64$) | 83.33% | 128,838 | 70.00% | 146,768 | 87.50% | **40,894** | 89.60% | **26,601** | **95.75%** | 9,712 |
| AdaPO voting ($\max_N = 64$) | **86.67%** | **61,250** | **76.67%** | **119,148** | **95.00%** | 52,885 | **95.60%** | 32,697 | **95.75%** | **6,194** |

# D Additional Ablation Experiment Results

**Analysis of Minimum $N$.** Table 11 studies the impact of the minimum rollout budget $N$ on both quality and efficiency. Across all $N$, AdaPO consistently reduces token usage relative to TTRL, with the largest savings occurring at smaller $N$ (e.g., at $N$=8: 58.19% on MATH-500, and 70.86% on GPQA), and with savings shrinking as $N$ increases (reflecting less room for early stopping / allocation gains). Importantly, these compute reductions do not come at the expense of overall performance: on AIME, AdaPO substantially improves mean@16 at $N$=8 (16.9 vs. 11.5) while using far fewer tokens (109M vs. 160M), and remains competitive at larger $N$. On AMC and MATH-500, AdaPO matches or slightly improves pass@16/pass@1 in most settings while still cutting tokens by 26.93–58.19%. On GPQA, AdaPO maintains comparable pass@16 and can improve pass@1 (notably at $N$=32, 34.5 vs. 30.5) while preserving meaningful savings (43.20% at $N$=32). Overall, the ablation indicates that AdaPO provides a favorable quality–efficiency trade-off across rollout budgets, with particularly strong benefits in the low-$N$ regime where compute is most constrained.

**Analysis of Confirmation Rounds $Z$** In this study, we fix the number of confirmation rounds per question while conducting experiments all based on minimum $N$=8. Table 12 shows that increasing the number of confirmation rounds $Z$ will generally yield increased performance, although AdaPO is comparative less *sensitive* to the choice of a fixed confirmation horizon $Z$ on AIME. In terms of

Table 11: Impact of the minimum number of rollouts ($N$) on AdaPO and TTRL. All experiments are conducted using Qwen2.5-Math-1.5B backbone with GRPO as the optimization algorithm.

| | AIME 2024 | | | | | AMC 2023 | | | | | MATH-500 | | | | | GPQA | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mean @16 | Pass @16 | Pass @1 | Toks. (↓) | Toks. Saving | Mean @16 | Pass @16 | Pass @1 | Toks. (↓) | Toks. Saving | Mean @16 | Pass @16 | Pass @1 | Toks. (↓) | Toks. Saving | Mean @16 | Pass @16 | Pass @1 | Toks. (↓) | Toks. Saving |
| $N=\mathbf{8}$ | | | | | | | | | | | | | | | | | | | | |
| TTRL | 11.5 | 43.3 | 20.0 | 160M | 32.07% | 45.5 | 80.7 | 53.0 | 164M | 42.24% | 70.0 | 90.8 | 72.2 | 228M | 58.19% | 26.3 | 87.3 | 31.5 | 197M | 70.86% |
| AdaPO | 16.9 | 50.0 | 23.3 | 109M | | 45.8 | 85.5 | 51.8 | 95M | | 69.0 | 90.8 | 71.8 | 95M | | 27.6 | 88.3 | 29.9 | 57M | |
| $N=\mathbf{16}$ | | | | | | | | | | | | | | | | | | | | |
| TTRL | 14.8 | 46.7 | 23.3 | 145M | 20.38% | 46.5 | 83.1 | 54.2 | 162M | 38.40% | 71.4 | 90.8 | 73.6 | 224M | 50.66% | 26.1 | 86.3 | 31.5 | 235M | 66.13% |
| AdaPO | 16.5 | 46.7 | 23.3 | 115M | | 46.2 | 83.1 | 55.4 | 100M | | 70.7 | 91.2 | 72.2 | 110M | | 27.1 | 88.3 | 31.5 | 79M | |
| $N=\mathbf{32}$ | | | | | | | | | | | | | | | | | | | | |
| TTRL | 15.6 | 43.3 | 20.0 | 142M | 17.25% | 47.7 | 83.1 | 54.2 | 159M | 26.93% | 72.3 | 90.6 | 73.6 | 218M | 32.57% | 26.5 | 87.8 | 30.5 | 251M | 43.20% |
| AdaPO | 14.6 | 46.7 | 20.0 | 117M | | 48.5 | 81.9 | 54.2 | 116M | | 72.3 | 90.8 | 73.8 | 147M | | 26.8 | 87.8 | 34.5 | 143M | |

compute, increasing $Z$ also generally only raises token usage slightly, yielding a clear efficiency–accuracy trade-off: smaller $Z$ provides the largest token saving (29.58% at $Z$=1) with slightly reduced pass@1 (20.0), whereas moderate $Z$ (e.g., $Z$=2, 3, 5) improves pass@1 to 23.3 with only a modest increase in tokens and still substantial savings (19.72–23.94%). Overall, the results suggest that AdaPO does not require careful tuning of $Z$ to obtain strong performance, and moderate confirmation horizons strike a favorable balance between accuracy and compute.

Table 12: Sensitivity of AdaPO to different settings of the confirmation $Z$ on the AIME benchmark.

| $Z$ | Mean@16 | Pass@16 | Pass@1 | Toks. | Toks. Saving |
|---|---|---|---|---|---|
| 1 | 14.0 | 46.7 | 20.0 | 100M | 29.58% |
| 2 | 14.4 | 46.7 | 23.3 | 114M | 19.72% |
| 3 | 14.2 | 46.7 | 23.3 | 108M | 23.94% |
| 4 | 13.8 | 43.3 | 23.3 | 107M | 24.65% |
| 5 | 14.6 | 46.7 | 23.3 | 112M | 21.13% |
| 6 | 14.6 | 46.7 | 20.0 | 112M | 21.13% |

# E Full Derivations & Proofs at *Electing* Stage

This appendix collects the proofs supporting Section 3.3: the triggering rule (Proposition 3.1) in Appendix E.1, the expected triggering time (Theorem 3.2) in Appendix E.2, and the decision regret bound (Proposition 3.3) in Appendix E.3.

We adopt throughout the notation of Section 3.3. In particular, $\mathcal{A} = \{1, \dots, m\}$ is the answer set; $D_t \in \mathcal{A}$ is the answer extracted from the $t$-th rollout; $V(a \mid D_{1:t}) := \sum_{k=1}^{t} \mathbb{I}\{D_k = a\}$ is the cumulative vote count of answer $a$; $a_L = \arg\max_{a \in \mathcal{A}} V(a \mid D_{1:t})$ and $a_R = \arg\max_{a \neq a_L} V(a \mid D_{1:t})$ are the current leader and runner-up; and $\Delta_t = V(a_L \mid D_{1:t}) - V(a_R \mid D_{1:t})$ is the vote gap. The latent quantity $a_M^\star$ denotes the full-budget majority winner. Under the symmetric categorical noise model, votes are conditionally independent given $a_M^\star$ with

$$\mathbb{P}(D_t = a_M^\star) = p_0, \qquad \mathbb{P}(D_t = a) = q_0 := \frac{1 - p_0}{m - 1} \ \text{ for } a \neq a_M^\star,$$

where $p_0 > 1/m$. We further write

$$\kappa := \frac{p_0}{q_0} = \frac{p_0(m-1)}{1 - p_0} > 1, \qquad A := \frac{1-\beta}{\alpha}, \qquad B := \frac{\beta}{1-\alpha}.$$

## E.1 Proof of Proposition 3.1: Triggering Rule

We derive the triggering rule by reducing the multi-class voting problem to a binary sequential test and applying Wald's SPRT.

**Reduction to a binary test.** At time $t$, consider the two hypotheses introduced in the main text:

$$H_1 : a_M^\star = a_L, \qquad H_0 : a_M^\star = a_R.$$

Under the symmetric noise model, every non-leading candidate $a \notin \{a_L, a_R\}$ shares the same per-rollout likelihood parameter $q_0$ as $a_R$ but has a strictly smaller vote count. Its posterior probability of being $a_M^\star$ is therefore uniformly dominated by that of $a_R$, so the dominant error event $\{a_L \neq a_M^\star\}$ is governed by the most competitive alternative $a_R$. It thus suffices to discriminate $H_1$ from $H_0$.

**Likelihood ratio.** Adopting a uniform prior on $(H_0, H_1)$, the posterior ratio coincides with the likelihood ratio. Under $H_i$ ($i \in \{0, 1\}$) the symmetric model gives

$$\mathbb{P}(D_{1:t} \mid H_i) = p_0^{V(a_i^\star | D_{1:t})} \prod_{a \neq a_i^\star} q_0^{V(a|D_{1:t})},$$

where $a_1^\star := a_L$ and $a_0^\star := a_R$. The Bayes factor is

$$\Lambda_t = \frac{\mathbb{P}(D_{1:t} \mid H_1)}{\mathbb{P}(D_{1:t} \mid H_0)} = \frac{p_0^{V(a_L)}\, q_0^{V(a_R)}}{p_0^{V(a_R)}\, q_0^{V(a_L)}} = \left(\frac{p_0}{q_0}\right)^{V(a_L)-V(a_R)} = \kappa^{\Delta_t},$$

where all factors corresponding to candidates $a \notin \{a_L, a_R\}$ cancel pairwise. Since $\kappa > 1$, $\Lambda_t$ is monotone increasing in $\Delta_t$.

**Wald's SPRT.** For target type-I and type-II error rates $(\alpha, \beta)$, Wald's Sequential Probability Ratio Test thresholds the likelihood ratio at $A$ and $B$ defined above and is optimal in the sense of minimizing the expected stopping time among all tests satisfying the same error constraints. The SPRT acceptance rule for $H_1$ is to declare $a_L$ correct at the first time $\Lambda_t \geq A$.

Substituting $\Lambda_t = \kappa^{\Delta_t}$ and taking logarithms gives the equivalent condition $\Delta_t \geq \log A / \log \kappa$. Since $\Delta_t \in \mathbb{Z}$, this is equivalent to $\Delta_t \geq \Delta_A$ with the integer boundary

$$\Delta_A := \left\lceil \frac{\log A}{\log \kappa} \right\rceil.$$

Allowing an optional minimum rollout count $t_0 \geq 1$, the trigger time is

$$\tau = \inf\{t \geq t_0 : \Delta_t \geq \Delta_A\},$$

which is the SPRT acceptance time for $H_1$ and establishes Proposition 3.1. □

**Scope: data-defined hypotheses and the one-sided trigger.** Wald's optimality theorem concerns a fixed pair of simple hypotheses tested with both boundaries active from the start. In AdaPO, $a_L$ and $a_R$ are the running arg-max and second arg-max of the votes, and the electing stage only checks the upper boundary $A$. The argument above is therefore exact on the identity-stable path $\mathcal{G}$, on which the pair $(a_L, a_R)$ does not change between $t_0$ and $\tau$: there the hypotheses are fixed and $\tau$ is the Wald acceptance time for $H_1$. Leaving $\mathcal{G}$ requires a competing answer to overtake a leader whose gap has positive drift $\mu$. The probability of such an overtake is controlled by the same large-deviation estimates used for the leader-stability step in Appendix E.2 and for the truncation term in Appendix E.3, and decays exponentially in the vote gap; the data-dependence of the hypotheses thus contributes an exponentially small correction. Appendix B verifies the top-2 reduction empirically. The lower boundary $B$ is not discarded: it is the condition that sends *Confirming* back to *Electing* (Theorem 3.4). The classical optimality guarantee therefore applies to the electing sub-procedure conditional on $\mathcal{G}$, while the two-stage automaton as a whole is a new construction built from these SPRT components.

### E.2 Proof of Theorem 3.2: Expected Triggering Time

We bound $\mathbb{E}[\tau]$ by reducing the gap process $\{\Delta_t\}$ to a biased random walk and applying Wald's identity.

**Drift of the gap process.** Condition on the high-probability event that the current leader is the true majority winner ($a_L = a_M^\star$). At each step a rollout produces $a_L$ with probability $p_0$, the runner-up $a_R$ with probability $q_0$, and any other particular candidate with probability $q_0$. The increment $\Delta_{t+1} - \Delta_t$ thus equals $+1$, $-1$, or $0$ with probabilities $p_0$, $q_0$, and $1 - p_0 - q_0$, respectively. Therefore,

$$\mathbb{E}[\Delta_{t+1} - \Delta_t \mid \mathcal{F}_t] = p_0 - q_0 = \frac{mp_0 - 1}{m - 1} =: \mu > 0,$$

where positivity follows from $p_0 > 1/m$, and the increments satisfy $|\Delta_{t+1} - \Delta_t| \leq 1$.

Strictly speaking, the leader $a_L$ may change with $t$. However, once $a_M^\star$ becomes the leader, every incorrect candidate has strictly negative drift in its gap relative to $a_M^\star$, so the probability that any incorrect candidate overtakes $a_M^\star$ at any future time decays exponentially. Conditioning on the high-probability event that $a_L = a_M^\star$ for all sufficiently large $t$, the dynamics of $\Delta_t$ near stopping are governed by a one-dimensional random walk with constant drift $\mu$; this approximation becomes exact in the boundary-crossing regime relevant to $\tau$.

**Expected hitting time via Wald's identity.** Let $\{X_t\}$ be i.i.d. with $\mathbb{E}[X_t] = \mu > 0$ and bounded support, and let $S_n = \sum_{k=1}^{n} X_k$. For the hitting time $\tau_A = \inf\{n \geq 0 : S_n \geq \Delta_A\}$, Wald's identity gives

$$\mathbb{E}[S_{\tau_A}] = \mu \cdot \mathbb{E}[\tau_A].$$

Since $S_{\tau_A} \geq \Delta_A$ and the overshoot $S_{\tau_A} - \Delta_A$ has bounded expectation (the increments are bounded), $\mathbb{E}[S_{\tau_A}] = \Delta_A + O(1)$, and hence

$$\mathbb{E}[\tau_A] = \frac{\Delta_A}{\mu} + O(1).$$

**Expected triggering time.** Combining the above with the leader-stability argument, on the high-probability event that $H_1$ is eventually accepted (which occurs with probability $1 - O(\alpha)$),

$$\mathbb{E}[\tau] = t_0 + \mathbb{E}[\tau_A \mid \Delta_{t_0}] + O(1) = O\left(\frac{\Delta_A}{\mu}\right).$$

Substituting $\Delta_A = \lceil \log A / \log \kappa \rceil$ and $A = (1-\beta)/\alpha$ yields

$$\mathbb{E}[\tau] = O\left(\frac{1}{\mu} \cdot \frac{\log(1/\alpha)}{\log \kappa}\right),$$

which establishes the first part of Theorem 3.2.

**Small-gap regime.** For $p_0 = 1/m + \xi$ with $0 < \xi \ll 1$, direct expansion gives $q_0 = 1/m - \xi/(m-1)$, so

$$\mu = \frac{m}{m-1}\,\xi = \Theta(\xi), \qquad \log \kappa = \log\big(1 + \Theta(\xi)\big) = \Theta(\xi).$$

Substituting yields

$$\mathbb{E}[\tau] = O\left(\frac{\log(1/\alpha)}{\xi^2}\right),$$

matching the classical Chernoff complexity for hypothesis testing with vanishing signal and establishing the second part of Theorem 3.2.

**Effect of truncation.** Under truncation at horizon $M$, write $\rho_M := \mathbb{P}(\tau > M)$. By Hoeffding's inequality applied to $\Delta_M - \mu(M - t_0)$ with bounded increments,

$$\rho_M \leq \exp\big(-c\,(M - t_0)\,\mu^2\big)$$

for an absolute constant $c > 0$. Provided $M \gtrsim \log(1/\alpha)/\mu^2$, the truncated stopping time satisfies

$$\mathbb{E}[\tau \wedge M] \leq \mathbb{E}[\tau] + M\rho_M = \mathbb{E}[\tau]\,(1 + o(1)),$$

so truncation does not affect the asymptotic rate. □

### E.3 Proof of Proposition 3.3: Decision Regret

We bound the decision regret $\mathcal{R}_M = \mathbb{P}(a_\tau \neq a_M^\star)$ of the truncated procedure by decomposing it into the SPRT error and an additional truncation term, and we identify the sharp exponential rate of the latter.

**Error decomposition.** Consider the two-sided SPRT

$$\tau = \inf\{t \geq t_0 : \Delta_t \geq \Delta_A \text{ or } \Delta_t \leq \Delta_B\}, \qquad \Delta_B := \lfloor \log B / \log \kappa \rfloor,$$

truncated at horizon $M$. If $\tau > M$, the procedure outputs the current leader $a_L(M)$.

Under $H_0$ (i.e., $a_M^\star = a_R$), an error occurs either by the SPRT accepting $H_1$ before $M$ or by truncating with the wrong leader still ahead:

$$\mathbb{P}_{H_0}(\text{error}) = \mathbb{P}_{H_0}(\tau \leq M,\ \Delta_\tau \geq \Delta_A) + \mathbb{P}_{H_0}(\tau > M,\ \Delta_M > 0).$$

By Wald's classical SPRT bound (up to the standard overshoot correction, which is negligible at the scale considered here), the first term is at most $\alpha$. For the second,

$$\{\tau > M,\ \Delta_M > 0\} \subseteq \{\Delta_M > 0\}, \qquad \mathbb{P}_{H_0}(\tau > M,\ \Delta_M > 0) \leq \mathbb{P}_{H_0}(\Delta_M > 0).$$

A symmetric decomposition holds under $H_1$ with $\beta$ and the event $\{\Delta_M < 0\}$.

**Sharp exponent of the truncation term.** Under $H_0$, each rollout contributes to the gap as

$$X_t = \begin{cases} +1, & \text{w.p. } q_0 \quad (\text{rollout} = a_L), \\ -1, & \text{w.p. } p_0 \quad (\text{rollout} = a_R), \\ 0, & \text{w.p. } 1 - p_0 - q_0 \quad (\text{rollout} \notin \{a_L, a_R\}), \end{cases}$$

and $\Delta_M = \sum_{t=1}^{M} X_t$. The increments are i.i.d. with negative mean $\mu_0 := q_0 - p_0 < 0$ and bounded support $|X_t| \le 1$.

By Cramér's theorem,

$$\mathbb{P}_{H_0}(\Delta_M > 0) \asymp \exp\left( M \inf_{\lambda \in \mathbb{R}} \Lambda(\lambda) \right), \qquad \Lambda(\lambda) := \log\big(q_0 e^{\lambda} + p_0 e^{-\lambda} + (1 - p_0 - q_0)\big).$$

Setting $\Lambda'(\lambda^*) = 0$ yields $q_0 e^{\lambda^*} = p_0 e^{-\lambda^*}$, i.e., $\lambda^* = \frac{1}{2}\log(p_0/q_0)$. Substituting,

$$\Lambda(\lambda^*) = \log\big(2\sqrt{p_0 q_0} + (1 - p_0 - q_0)\big) = \log\big(1 - (\sqrt{p_0} - \sqrt{q_0})^2\big),$$

which is strictly negative whenever $p_0 \neq q_0$ (in particular whenever $p_0 > 1/m$). Defining

$$c := -\log\big(1 - (\sqrt{p_0} - \sqrt{q_0})^2\big) > 0,$$

we obtain $\mathbb{P}_{H_0}(\Delta_M > 0) \asymp e^{-cM}$. (The coarser bound $\mathbb{P}_{H_0}(\Delta_M > 0) \le \exp(-M\mu_0^2/2)$ also follows directly from Hoeffding's inequality applied to the bounded increments; we retain the sharp Cramér exponent above.) The case $m = 2$ recovers $1 - p_0 - q_0 = 0$ and $c = -\log(2\sqrt{p_0 q_0})$ as a special case.

By the symmetric role of $p_0$ and $q_0$ in the LMGF, the same exponent $c$ applies to $\mathbb{P}_{H_1}(\Delta_M < 0)$.

**Conclusion.** Combining the SPRT and truncation bounds under both hypotheses,

$$\mathcal{R}_M \le \max\{\alpha, \beta\} + e^{-cM},$$

with $c = -\log\big(1 - (\sqrt{p_0} - \sqrt{q_0})^2\big) > 0$ depending only on $(p_0, m)$. This establishes Proposition 3.3. □

# F Full Derivations & Proofs at *Confirming* Stage

This appendix collects the proofs supporting Section 3.4: the Confirming-stage stopping rule (Theorem 3.4) in Appendix F.1, the expected confirmation time (Theorem 3.5) in Appendix F.2, and the decision regret bound under accumulative confirmation (Corollary 3.6) in Appendix F.3.

We adopt throughout the notation of Section 3.4 and Appendix E. In particular, $a_L$ is the leader at the trigger time $\tau$ (from Proposition 3.1) and $a_R$ is the runner-up; $\Delta_t = V(a_L \mid D_{1:t}) - V(a_R \mid D_{1:t})$ is the vote gap; $a_M^\star$ is the latent full-budget majority winner. The symmetric categorical noise model gives $p_0 := \mathbb{P}(D_t = a_M^\star)$, $q_0 := (1 - p_0)/(m - 1)$, $\kappa := p_0/q_0 > 1$, and drift $\mu := p_0 - q_0 = (mp_0 - 1)/(m - 1) > 0$. The SPRT thresholds are

$$A := \frac{1 - \beta}{\alpha}, \qquad B := \frac{\beta}{1 - \alpha}, \qquad \Delta_A := \left\lceil \frac{\log A}{\log \kappa} \right\rceil, \qquad \Delta_B := \left\lfloor \frac{\log B}{\log \kappa} \right\rfloor.$$

We work with the binary hypothesis reduction from Appendix E.1:

$$H_1 : a_M^\star = a_L, \qquad H_0 : a_M^\star = a_R.$$

Under this reduction, the per-rollout Bayes factor between $H_1$ and $H_0$ equals $\kappa$ (vote $= a_L$), $1/\kappa$ (vote $= a_R$), or 1 (vote $\notin \{a_L, a_R\}$), and the cumulative Bayes factor is $\Lambda_t = \kappa^{\Delta_t}$.

For $t \ge \tau$, let $Z_t$ denote the number of *confirmation rounds* accumulated since entering *Confirming*, i.e., the number of distinct time steps $s \in [\tau, t]$ at which the dominance condition $\Delta_s \ge \Delta_A$ holds (with $Z_\tau := 1$ counting the trigger itself). This matches the counter incremented in Algorithm 2.

### F.1 Proof of Theorem 3.4: Confirming-Stage Stopping Rule

Theorem 3.4 comprises three transitions out of *Confirming*. We derive each in turn.

**Effective Bayes factor after $Z$ confirmations.** For $Z \geq 1$, define the event

$$E_Z := \{\Delta_t \geq \Delta_A \text{ at } Z \text{ distinct time steps } t \geq \tau\},$$

and let $\Lambda_{\text{eff}}(Z) := \mathbb{P}(E_Z \mid H_1)/\mathbb{P}(E_Z \mid H_0)$ be the corresponding Bayes factor. By the chain rule,

$$\Lambda_{\text{eff}}(Z) = \frac{\mathbb{P}(E_1 \mid H_1)}{\mathbb{P}(E_1 \mid H_0)} \cdot \prod_{k=2}^{Z} \frac{\mathbb{P}(E_k \mid E_{k-1}, H_1)}{\mathbb{P}(E_k \mid E_{k-1}, H_0)}.$$

*First factor.* The event $E_1$ is the SPRT acceptance event for $H_1$ from Appendix E.1, so by Wald's bound

$$\frac{\mathbb{P}(E_1 \mid H_1)}{\mathbb{P}(E_1 \mid H_0)} \geq A.$$

*Subsequent factors.* Conditioned on $E_{k-1}$, the gap satisfies $\Delta_t \geq \Delta_A$. The next informative rollout (i.e., one voting for $a_L$ or $a_R$) is $a_L$ with conditional probability $p_0/(p_0+q_0)$ under $H_1$ and $q_0/(p_0+q_0)$ under $H_0$, so its likelihood ratio is exactly $\kappa$. Tracking only the contribution of the rollouts that move the gap, each subsequent confirmation contributes a Bayes factor of $\kappa$ in the SPRT sense:

$$\frac{\mathbb{P}(E_k \mid E_{k-1}, H_1)}{\mathbb{P}(E_k \mid E_{k-1}, H_0)} \;\approx\; \kappa.$$

This relation is exact for $m = 2$, where every rollout is informative, and is the standard SPRT amplification approximation in the general $m$-class case.

Combining,

$$\Lambda_{\text{eff}}(Z) \;\geq\; A \cdot \kappa^{Z-1}.$$

**Posterior bound and the End transition.** Under a uniform prior on $(H_0, H_1)$,

$$\mathbb{P}(H_1 \mid E_Z) = \frac{\Lambda_{\text{eff}}(Z)}{1 + \Lambda_{\text{eff}}(Z)},$$

which is monotone increasing in $\Lambda_{\text{eff}}$. Substituting the lower bound on $\Lambda_{\text{eff}}(Z)$ gives

$$\mathbb{P}(a_M^\star = a_L \mid E_Z) \;\geq\; \frac{A\kappa^{Z-1}}{A\kappa^{Z-1}+1}.$$

To guarantee posterior confidence at least $\gamma \in (0,1)$, it suffices that

$$\frac{A\kappa^{Z-1}}{A\kappa^{Z-1}+1} \geq \gamma \iff A\kappa^{Z-1} \geq \frac{\gamma}{1-\gamma} \iff Z \geq 1 + \frac{\log\big(\gamma/(1-\gamma)\big) - \log A}{\log \kappa}.$$

Taking the smallest integer satisfying this inequality gives

$$Z^\star(\gamma) := \left\lceil 1 + \frac{\log\big(\gamma/(1-\gamma)\big) - \log A}{\log \kappa} \right\rceil,$$

and the End transition $Z_t \geq Z^\star(\gamma)$ stated in Theorem 3.4 follows.

**Confirming-to-Electing transition.** The lower-boundary transition $\Delta_t < \log B/\log \kappa$ is equivalent to $\Lambda_t < B$, i.e., the SPRT rejection condition for $H_1$. When this holds, the runner-up has acquired sufficient statistical dominance to invalidate the current leader; the procedure discards the accumulated confirmation count and resumes Electing with a fresh leader-runner-up pair. □

### F.2 Proof of Theorem 3.5: Expected Confirmation Time

We bound the expected number of additional rollouts $\tilde{t}$ required to accumulate $Z^\star(\gamma) - 1$ further confirmations after the trigger.

Throughout, we condition on the high-probability event that the trigger-time leader is correct, i.e., $H_1$ holds. By the Electing-stage analysis (Appendix E.3), this occurs with probability at least $1 - \alpha - e^{-cM}$. Under $H_1$, the gap process has positive drift $\mu$ and bounded increments $|\Delta_{t+1} - \Delta_t| \leq 1$.

**Expected inter-confirmation time.** Let $T_1$ denote the time between two successive confirmations, starting at a state with $\Delta_t = \Delta_A$. Decomposing on the next step under $H_1$:

- with probability $1 - q_0 = p_0 + (1 - p_0 - q_0)$, the next step yields $\Delta_{t+1} \geq \Delta_A$, registering an immediate confirmation ($T_1 = 1$);
- with probability $q_0$, the next step yields $\Delta_{t+1} = \Delta_A - 1$, after which the expected first-passage time back to $\Delta_A$ in a positive-drift random walk with bounded increments is $1/\mu$ by Wald's identity.

Therefore,

$$\mathbb{E}[T_1] \leq 1 \cdot (1 - q_0) + (1 + 1/\mu) \cdot q_0 = 1 + q_0/\mu = O(1/\mu).$$

**Total confirmation time.** Since $Z^\star(\gamma) - 1$ additional confirmations are required after the initial trigger and inter-confirmation times have expectation $O(1/\mu)$, by linearity of expectation

$$\mathbb{E}[\tilde{t}] = (Z^\star(\gamma) - 1) \cdot \mathbb{E}[T_1] = O\left(\frac{Z^\star(\gamma) - 1}{\mu}\right),$$

which establishes Theorem 3.5. □

**Combined expected stopping time.** Combining with the expected first-hitting time from Theorem 3.2,

$$\mathbb{E}[\tau + \tilde{t}] = O\left(\frac{1}{\mu} \cdot \frac{\log(1/\alpha)}{\log \kappa}\right) + O\left(\frac{Z^\star(\gamma) - 1}{\mu}\right) = O\left(\frac{\log(1/\alpha) + \log\big(\gamma/(1-\gamma)\big)}{\mu \log \kappa}\right).$$

**Small-gap regime.** For $p_0 = 1/m + \xi$ with $\xi \ll 1$, $\mu = \Theta(\xi)$ and $\log \kappa = \Theta(\xi)$, hence

$$\mathbb{E}[\tau + \tilde{t}] = O\left(\frac{\log(1/\alpha) + \log\big(\gamma/(1-\gamma)\big)}{\xi^2}\right).$$

Adaptive accumulative confirmation thus preserves the Electing-stage $\xi^{-2}$ sample complexity while adding only an additive $\log(\gamma/(1-\gamma))/\xi^2$ overhead for the user-controlled correctness target $\gamma$.

### F.3 Proof of Corollary 3.6: Decision Regret with Confirmation

We bound the decision regret $\mathcal{R}_M^{(\gamma)} := \mathbb{P}(a_{\tau_\gamma \wedge M} \neq a_M^\star)$ of the truncated procedure with adaptive confirmation count $Z^\star(\gamma)$, where $\tau_\gamma := \tau + \tilde{t}$ is the total stopping time.

**Error decomposition.** The error event decomposes as

$$\{a_{\tau_\gamma \wedge M} \neq a_M^\star\} \subseteq \underbrace{\{\tau_\gamma \leq M,\ a_{\tau_\gamma} \neq a_M^\star\}}_{\text{accepted, wrong leader}} \cup \underbrace{\{\tau_\gamma > M,\ a_L(M) \neq a_M^\star\}}_{\text{truncated, wrong leader}},$$

so by the union bound

$$\mathcal{R}_M^{(\gamma)} \leq \mathbb{P}(a_{\tau_\gamma} \neq a_M^\star \mid \tau_\gamma \leq M) + \mathbb{P}(\tau_\gamma > M,\ a_L(M) \neq a_M^\star).$$

**Posterior bound on the acceptance event.** By Theorem 3.4, the End transition fires only when the posterior satisfies $\mathbb{P}(a_M^\star = a_L \mid E_{Z_{\tau_\gamma}}) \geq \gamma$. Hence the conditional probability of accepting the wrong leader is

$$\mathbb{P}(a_{\tau_\gamma} \neq a_M^\star \mid \tau_\gamma \leq M) = 1 - \mathbb{P}(a_M^\star = a_L \mid E_{Z_{\tau_\gamma}}) \leq 1 - \gamma.$$

**Truncation bound.** The truncation event imposes the same vote-gap deviation as analyzed in Appendix E.3, yielding

$$\mathbb{P}(\tau_\gamma > M,\ a_L(M) \neq a_M^\star) \leq e^{-cM}, \qquad c = -\log\big(1 - (\sqrt{p_0} - \sqrt{q_0})^2\big) > 0.$$

The constant $c$ depends only on $(p_0, m)$ and is the sharp Cramér exponent derived in Appendix E.3.

**Conclusion.** Combining the two bounds,

$$\mathcal{R}_M^{(\gamma)} \;\le\; (1-\gamma) + e^{-cM},$$

which establishes Corollary 3.6. □

# G Future Work: Extension to External Reward Based TTT Methods

While the scope of this work targets the most widely-adopted settings in test-time training where reward signals are derived from majority voting over discrete answers, our principled sequential testing automaton philosophy can be extended to test-time training frameworks relying on external reward models that provide scalar-valued feedback (e.g., preference models, verifiers, or learned reward functions). This appendix presents the adapted sufficient statistic, trigger rule, and confirmation rule, and outlines the principal challenges that remain open. The structural correspondence with the discrete case is summarized in Table 13.

## G.1 Scalar Reward Setting and Objective

Let $r : \mathcal{Y} \to [0,1]$ be an external reward model and let $r_t := r(y_t)$. Each rollout now contributes a real-valued signal rather than a discrete vote. The natural analogue of the full-budget majority winner $a_M^\star$ is the answer with the highest expected reward,

$$a^\star_{\text{rew}} \;:=\; \arg\max_{a\in\mathcal{A}} \mu_a, \qquad \mu_a := \mathbb{E}\big[r(y) \mid \mathcal{E}(y) = a,\, x\big],$$

with empirical analogue $\hat{a}^\star_M := \arg\max_a \hat{\mu}_a(M)$, where $\hat{\mu}_a(t)$ and $n_a(t)$ are the empirical mean reward and rollout count for answer $a$ at time $t$. The adaptive allocation objective from Section 3.3 is unchanged,

$$\min_{\tau,\delta}\; \mathbb{E}[\tau] \quad \text{s.t.} \quad \mathbb{P}(\delta \neq \hat{a}^\star_M) \le \epsilon,$$

with $\hat{a}^\star_M$ playing the role of the test-time reference signal.

## G.2 Gaussian Reward Model and Electing-Stage Trigger

The automaton $\mathcal{Q} = \{\mathit{Start}, \mathit{Elc}, \mathit{Cfm}, \mathit{End}\}$ retains its topology; the integer vote gap $\Delta_t$ is replaced by a continuous summary of accumulated reward evidence. Let $a_L, a_R$ denote the empirical reward leader and runner-up, $n_{LR}(t) := n_{a_L}(t) + n_{a_R}(t)$, and $G_t := \hat{\mu}_{a_L}(t) - \hat{\mu}_{a_R}(t) \ge 0$ the empirical reward gap.

**Gaussian noise model.** For tractability we assume rewards are i.i.d. Gaussian conditional on the answer,

$$r_t \mid \mathcal{E}(y_t) = a \;\overset{\text{i.i.d.}}{\sim}\; \mathcal{N}(\mu_a, \sigma^2),$$

with $\sigma^2$ pre-estimated from the initial $N$ rollouts, and posit a minimum effect size $|\mu_{a_L} - \mu_{a_R}| = \delta_0 > 0$. This plays the same role as the symmetric categorical noise model in Section 3.3: it provides the minimal homogeneous structure that reduces the multi-class reward comparison to a binary test with an explicit, computable statistic. Without it, the joint distribution depends on the full empirical mean vector across all $m$ answers, yielding an intractable high-dimensional decision rule.

**Log-likelihood ratio.** Let $S_L(t), S_R(t)$ denote the accumulated reward sums for $a_L, a_R$. Under the two-point hypotheses $H_1 : (\mu_{a_L}, \mu_{a_R}) = (\mu + \frac{\delta_0}{2}, \mu - \frac{\delta_0}{2})$ vs. $H_0$ (symmetric) for an unknown common offset $\mu$, the log-likelihood ratio is

$$\log \Lambda_t \;=\; \frac{\delta_0}{\sigma^2}\big(S_L(t) - S_R(t)\big) - \frac{\delta_0^2}{2\sigma^2}\big(n_{a_L}(t) - n_{a_R}(t)\big).$$

Under approximately balanced allocation ($n_{a_L}(t) \approx n_{a_R}(t) \approx n_{LR}(t)/2$), the nuisance term vanishes and

$$\log \Lambda_t \;\approx\; \frac{\delta_0}{\sigma^2}\big(S_L(t) - S_R(t)\big) \;=\; \frac{\delta_0\, n_{LR}(t)}{2\sigma^2}\, G_t. \tag{1}$$

Each informative rollout contributes an expected log-likelihood increment of $\mu_{\text{rew}} := \delta_0^2/(2\sigma^2) > 0$, the continuous analogue of the discrete drift $\mu = (mp_0 - 1)/(m-1)$.

**Proposition G.1** (Electing-Stage Trigger for Scalar Rewards)**.** *Under the Gaussian reward model, the SPRT-optimal trigger fires at*

$$\tau_{\text{rew}} \;:=\; \inf\{t \geq t_0 : \log \Lambda_t \geq \log A\}, \qquad A := \frac{1-\beta}{\alpha}.$$

*Equivalently, by* (1)*, the trigger is a time-varying threshold on the empirical reward gap,*

$$G_t \;\geq\; \frac{2\sigma^2 \log A}{\delta_0\, n_{LR}(t)},$$

*and minimizes expected sample size subject to error constraints* $(\alpha, \beta)$ *by Wald's SPRT theorem.*

Unlike the fixed integer boundary $\Delta_A = \lceil \log A / \log \kappa \rceil$ in Proposition 3.1, the scalar trigger on $G_t$ shrinks at rate $1/n_{LR}(t)$, reflecting the reduced uncertainty about the true mean difference as evidence accumulates. By Wald's identity, the expected trigger time recovers the classical Gaussian SPRT complexity,

$$\mathbb{E}[\tau_{\text{rew}}] \;=\; O\!\left(\frac{\sigma^2}{\delta_0^2} \log \frac{1}{\alpha}\right),$$

the continuous analogue of Theorem 3.2: easy instances (large $\delta_0$) terminate quickly while harder instances consume more budget in a controlled manner.

### G.3 Confirming-Stage Stopping Rule

Let $\kappa_{\text{rew}} := e^{\delta_0^2/(2\sigma^2)}$ denote the per-step likelihood amplification factor. Upon entering *Confirming* at $\hat{t} = \tau_{\text{rew}}$, the electing-stage analysis guarantees a Bayes factor of at least $A$ in favor of $a_L$, and each subsequent informative rollout multiplies it by $\kappa_{\text{rew}}$ in expectation. The same posterior calculation as Theorem 3.4 yields:

**Proposition G.2** (Confirming-Stage Stopping for Scalar Rewards)**.** *Let* $A = (1-\beta)/\alpha$*,* $B = \beta/(1-\alpha)$*, and let* $Z_t$ *count rollouts since entering Confirming. The transition rule*

$$\pi_t = \begin{cases} \textit{End}, & \dfrac{A\,\kappa_{\text{rew}}^{Z_t - 1}}{A\,\kappa_{\text{rew}}^{Z_t - 1} + 1} \geq \gamma, \\ \textit{Electing}, & \log \Lambda_t < \log B, \\ \textit{Confirming}, & \textit{otherwise,} \end{cases}$$

*has stopping condition equivalent to*

$$Z_t \;\geq\; Z^{\star}_{\text{rew}}(\gamma) \;:=\; \left\lceil 1 + \frac{2\sigma^2}{\delta_0^2}\left(\log \frac{\gamma}{1-\gamma} - \log A\right)\right\rceil.$$

This formula is identical in form to $Z^{\star}(\gamma)$ in Theorem 3.4, with $\log \kappa$ replaced by $\delta_0^2/(2\sigma^2)$, confirming that the posterior-threshold confirmation mechanism is agnostic to the reward type and depends only on the per-step information rate. The decision regret bound $\mathcal{R}_M^{(\gamma)} \leq (1-\gamma) + e^{-cM}$ and the $O(1)$ asymptotic confirmation overhead from Theorem 3.5 carry over verbatim, with $c > 0$ now depending on $(\delta_0, \sigma^2)$.

### G.4 Summary of Analogies

Table 13 summarizes the one-to-one correspondences between the discrete and scalar formulations, confirming that the AdaPO framework is architecturally separable from the specific reward modality.

### G.5 Open Challenges of the Extension

For future researchers who want to implement this extension, several open challenges remain before the scalar extension becomes fully rigorous and deployable: the trigger and confirmation rules depend on estimating $(\delta_0, \sigma^2)$ robustly from initial rollouts; the Gaussian fixed-$\delta_0$ model should be generalized to composite or non-Gaussian reward settings, for example via GLR or mixture-SPRT style tests; reward-model bias, especially multiplicative or nonlinear distortion, can affect the

Table 13: Correspondence between the discrete (majority-voting) and scalar (external reward) formulations of the two-stage sequential testing automaton.

| **Component** | **Discrete (AdaPO)** | **Scalar (Extension)** |
|---|---|---|
| Noise model | Symmetric categorical, $p_0$ | Gaussian, $(\delta_0, \sigma^2)$ |
| Sufficient statistic | Vote gap $\Delta_t \in \mathbb{Z}_{\geq 0}$ | Log-LR $\log \Lambda_t \in \mathbb{R}$ |
| Per-step amplification | $\kappa = \frac{p_0(m-1)}{1-p_0}$ | $\kappa_{\text{rew}} = e^{\delta_0^2/(2\sigma^2)}$ |
| Drift under $H_1$ | $\mu = \frac{mp_0-1}{m-1}$ | $\mu_{\text{rew}} = \frac{\delta_0^2}{2\sigma^2}$ |
| Electing trigger | $\Delta_t \geq \lceil \log A / \log \kappa \rceil$ (fixed) | $G_t \geq \frac{2\sigma^2 \log A}{\delta_0 \, n_{LR}(t)}$ (time-varying) |
| Expected trigger time | $O\Big(\frac{1}{\mu} \cdot \frac{\log(1/\alpha)}{\log \kappa}\Big)$ | $O\Big(\frac{\sigma^2}{\delta_0^2} \log \frac{1}{\alpha}\Big)$ |
| Revert trigger | $\Delta_t < \lfloor \log B / \log \kappa \rfloor$ | $\log \Lambda_t < \log B$ |
| Required confirmations | $\lceil 1 + \frac{\log \frac{\gamma}{1-\gamma} - \log A}{\log \kappa} \rceil$ | $\lceil 1 + \frac{2\sigma^2}{\delta_0^2}(\log \frac{\gamma}{1-\gamma} - \log A) \rceil$ |
| Decision regret | $(1-\gamma) + e^{-cM}$ | $(1-\gamma) + e^{-cM}$ |

effective signal-to-noise ratio and boundary calibration; extending the leader-vs-runner-up reduction to asymmetric multi-answer settings and process reward models may require more complex best-arm style machinery; and, in implementation, the scalar trigger is time-varying and must be recomputed after each batch using the updated $n_{LR}$, though this adds little overhead in the existing batch-sequential algorithm. These are obviously interesting points to study for the extension, but beyond the scope of this work.

## 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: The abstract and Section 1 state the contributions, the modeling assumptions under which the theoretical results hold (symmetric categorical noise, stable leader/runner-up identities), and the empirical scope (Section 4.1).

   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: The scope of the optimality result and the modeling assumptions are discussed in Section 3.3; limitations and future work are discussed at the end of Section 4.3, and robustness to misspecification is studied in Appendix B.

   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: Each result states its assumptions (symmetric categorical noise model, identity-stable path, untruncated or truncated process); the main text gives proof sketches and Appendices E and F give full proofs, stating explicitly where standard approximations (Wald overshoot, top-2 reduction) are 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: Section 4.1, Algorithms 1 and 2 give all settings, and the released code contains the exact launch commands.

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: The code, data preprocessing scripts and launch commands are released at `https://github.com/GeometricMLMechanism/AdaPO`; all benchmarks are public.

   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: See the implementation details in Section 4.1 (sampling parameters, rollout budgets, SPRT parameters, learning rates, optimizer).

   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: [No]

   Justification: Each test-time training run takes up to tens of GPU hours, so we do not report error bars over repeated training runs; evaluation metrics are averaged over 16 sampled responses per question (Mean@16), and for every method each metric is the maximum over the validation points during training (Section 4.1), and the voting analyses in Appendix B are averaged over many random vote orderings.

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: Section 4.1 lists the GPU type and the number of GPUs per run, and Table 3 reports the wall-clock time of complete runs.

   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: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   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: [N/A]

Justification: AdaPO is a foundational method that reduces the compute (and hence energy) needed for test-time training of LLMs; it does not create new capabilities beyond those of the underlying models, and we see no direct path to negative applications specific to it.

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: [N/A]

Justification: The paper poses no such risks.

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 models, datasets and code bases are cited, and their licenses are listed in Section 4.1.

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: [Yes]

    Justification: The released code (Apache-2.0) is documented in its README, including installation, data preparation, launch commands and hyper-parameters; no new datasets or models are released.

    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: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    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: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

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 are used only for editing and formatting purposes. All the core methodology, scientific rigor, and originality of the research are from the authors.

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.