

Reasoning Confidence Is Not Policy-Invariant: Calibrating Prefix Success under Model and Budget Shift

Xin Zhang¹, Qiyu Wei¹, Chenhan Yuan², Sophia Ananiadou¹

¹The University of Manchester, Manchester, United Kingdom

²Alibaba Group

xin.zhang-41@postgrad.manchester.ac.uk qiyu.wei@postgrad.manchester.ac.uk
yuanchenhan.ych@alibaba-inc.com sophia.ananiadou@manchester.ac.uk

Abstract

Inference-time scaling of LLM reasoning relies on a confidence score over each partial trace, yet existing estimators treat prefix quality as intrinsic to the prefix. A given prefix can succeed at very different rates and rankings under different continuation policies. Reasoning confidence is therefore not policy-invariant, and a squared-loss lower bound shows that policy-blind predictors retain residual error proportional to cross-policy disagreement. Calibration must condition on the continuation model and inference budget. To bridge this gap, we propose the Comparable Policy-Conditioned Calibrator (CPCC¹), built from a shared frozen-LLM prefix encoder, comparable scalar policy features under per-policy z -score normalization, and a Beta-Binomial head trained on bootstrap success counts. Across six continuation policies and three reasoning benchmarks (GSM8K, SVAMP, MATH-500), all evaluated under majority-vote aggregation, the CPCC family attains the lowest Brier on every evaluation cell, and CPCC is strongest on the held-out model families. It also avoids the held-out-family collapse of raw-hidden conditioning, lowers ECE by 13 to 26 percent on out-of-distribution math, and reduces downstream selective risk by 10 to 30 percent.

1 Introduction

Inference-time scaling has become a central mechanism for improving large language model (LLM) reasoning (Snell et al., 2024; Brown et al., 2024). Chain-of-thought prompting (Wei et al., 2022) and self-consistency aggregation (Wang et al., 2023) produce many candidate traces, while verifier-guided selection (Cobbe et al., 2021; Lightman et al., 2024), tree-based search (Yao et al., 2023), and confidence-weighted self-consistency (Taubenfeld et al., 2025; Kang et al., 2025; Fu et al., 2025)

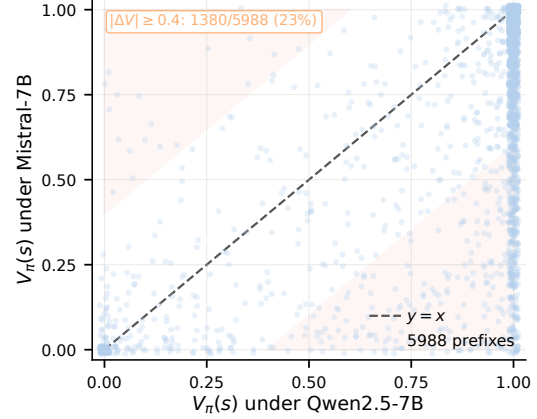


Figure 1: Same prefix, different continuation policy: per-prefix success $V_\pi(s)$ on GSM8K-train at $B=1$ under Qwen2.5-7B vs. Mistral-7B. 1380/5988 prefixes (23%) fall in the red disagreement corners ($|\Delta V| \geq 0.4$); cross-family ranking-flip rates reach 30–51% (Appendix D).

rank or prune those traces. All of these methods rely on a common primitive: a confidence or value score on a partial reasoning trace. Downstream decisions of budget allocation, beam pruning, and abstention are only as good as that score.

Existing approaches to producing such a score fall into three broad lines. Token-level and model-internal uncertainty methods estimate the predictive uncertainty of the current generator (Kadavath et al., 2022; Kuhn et al., 2023; Farquhar et al., 2024; Zhang et al., 2026). Response-level confidence methods such as Self-Certainty, CISC, and DeepConf score completed continuations and use those scores to weight or filter rollouts at decision time (Kang et al., 2025; Taubenfeld et al., 2025; Fu et al., 2025). Process reward models and prefix value models estimate the probability that a partial trajectory will eventually yield a correct answer (Lightman et al., 2024; Park et al., 2025; Ni et al., 2025). These three lines differ in what they score, yet they share an implicit assumption that prefix quality is a property of the prefix or of the

¹Code: <https://github.com/XR-OpenSource/EMNLP2026-CPCC-Official>

current generator’s distribution.

Reasoning confidence is not policy-invariant. Cross-family deployment breaks this assumption. Across six continuation policies and four data splits, the same prefix exhibits a cross-family success-rate gap of 0.18 to 0.26 between Qwen-7B and Mistral-7B (Figure 1). Prefix ranking-flip rates on the non-trivial subset reach 30 to 51 percent across cross-family pairs. A step that a strong continuation model would recover from can be fatal for a weaker one. Different continuation models therefore disagree on the relative ordering of prefixes, which is the signal that selective inference relies on. A direct conditioning fix that reads the continuation model’s hidden state collapses outside its training distribution: Spearman against empirical success drops to -0.09 on held-out Llama and to 0.03 on held-out Mistral, because hidden-state geometries of different LLM families are not registered to a shared coordinate system.

To address this gap, we propose the **Comparable Policy-Conditioned Calibrator (CPCC)**, a small head that estimates the policy-conditioned prefix-success probability from three ingredients. First, a shared frozen LLM prefix encoder produces a prefix representation that is the same object across all candidate continuation policies. Second, a low-dimensional vector of comparable scalar policy descriptors carries the policy-conditioning signal in coordinates whose operational semantics are stable across model families. These descriptors consist of moments and boundary uncertainties of the policy’s own prefix likelihood and answer-space and length-space summaries from a few unlabeled probe roll-outs. Third, a per-policy z-score normalization removes residual scale differences, and a Beta-Binomial predictive head models bootstrap success counts heteroscedastically and exposes both predictive mean and predictive variance. Our contributions are: (i) we formalize policy-conditioned prefix success $V_{\pi,B,A}$ and prove a squared-loss lower bound on any policy-blind predictor; (ii) we identify raw-hidden conditioning as a structural held-out-family failure mode, with Spearman dropping to -0.09 on held-out Llama and 0.03 on held-out Mistral; (iii) we empirically show, under majority-vote aggregation, that the CPCC family attains the lowest Brier on every evaluation cell, with CPCC strongest on the held-out families, ECE reductions of 13 to 26 percent on out-of-distribution math, and a 10 to 30 percent reduction in downstream selective risk.

2 Related Work

2.1 Confidence and Uncertainty for Fixed Generators

A large literature studies whether language models can assess their own correctness. [Kadavath et al. \(2022\)](#) show that prompted LLMs can estimate $P(\text{True})$ and $P(\text{IK})$, although calibration degrades under distribution shift. Semantic uncertainty methods estimate uncertainty over meaning classes rather than surface forms ([Kuhn et al., 2023](#)) and have been used to detect confabulations in open-ended generation ([Farquhar et al., 2024](#)), with lighter-weight variants approximating semantic entropy from internal states ([Kossen et al., 2024](#)). Reasoning-specific token-level uncertainty appears in TokUR through low-rank weight perturbations ([Zhang et al., 2026](#)). Closer to inference-time scaling, recent confidence-guided aggregation methods use a model-internal signal to weight, filter, or select among completed traces under a fixed generator. Self-Certainty drives scalable Best-of- N selection from distributional certainty ([Kang et al., 2025](#)), DeepConf filters low-quality traces during test-time parallel thinking ([Fu et al., 2025](#)), and CISC weights self-consistency votes with model-provided confidence ([Taubenfeld et al., 2025](#)). All of these methods estimate uncertainty of a fixed generator or response distribution and do not separate the prefix from the continuation policy that will complete it. We compare against all four on a matched subset in Section 4.5.

2.2 Process Reward Models, Prefix Values, and Search

Process reward models and verifiers estimate whether a generated step or trajectory will eventually yield a correct answer ([Cobbe et al., 2021](#); [Lightman et al., 2024](#)). Recent work on PRM calibration shows that such models can be miscalibrated and proposes calibrated success estimates for adaptive inference-time scaling ([Park et al., 2025](#)). UHeads train lightweight uncertainty heads on frozen LLM internal states for step-level verification ([Ni et al., 2025](#)), Tree-of-Thoughts generalizes decoding into search over reasoning units ([Yao et al., 2023](#); [Hao et al., 2023](#)), particle-based Monte Carlo views inference-time scaling probabilistically ([Puri et al., 2025](#)), and uncertainty-aware tree search allocates computation by uncertainty ([Huang et al., 2026](#); [Song et al., 2026](#)). PRM-style prefix values are usually trained under an

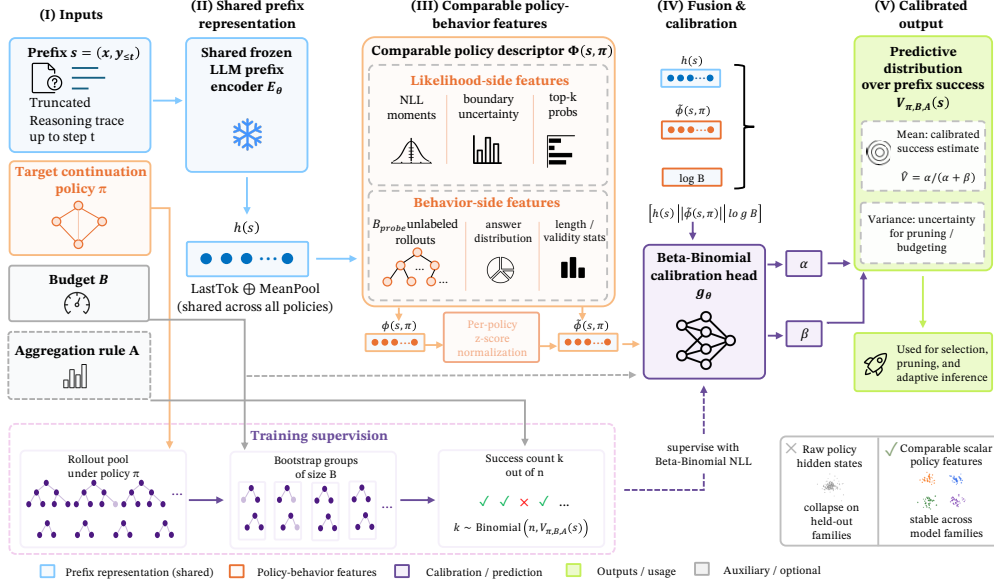


Figure 2: **Overview of CPCC.** A shared frozen-LLM encoder produces a prefix representation $h(s)$. For each continuation policy π , a low-dimensional vector of comparable scalar features $\phi(s, \pi)$ is computed from policy-side prefix-likelihood statistics and from a small number of unlabeled probe rollouts. Combined features are per-policy normalized and fed, together with $\log B$, to a Beta-Binomial head that predicts (α, β) . Training targets are bootstrapped success counts k out of n resamples at each budget.

implicit continuation distribution that is fixed at training time. Our claim is that this continuation distribution must be an explicit argument of the value function. A robust calibrator should treat the continuation policy, budget, and aggregation rule as conditioning variables rather than as hidden assumptions of the training data, and this is what CPCC does.

3 Policy-Conditioned Prefix Success and CPCC

Let x denote a problem, $y = (y_1, \dots, y_T)$ a generated reasoning trace, and $s_t = (x, y_{\leq t})$ a step-level prefix. A continuation condition is a triple $c = (\pi, B, A)$ specifying a continuation policy, an inference budget, and an aggregation rule. We define the policy-conditioned prefix success probability as

$$\begin{aligned} V_{\pi, B, A}(s_t) &= \Pr(R=1 \mid s_t, \pi, B, A) \\ &= \mathbb{E}_{\hat{y} \sim A(\pi, B \mid s_t)}[r(x, y_{\leq t}, \hat{y})], \end{aligned} \quad (1)$$

where r is an automatic verifier. For each (s_i, c_j) we observe a bootstrap success count $k_{ij, B} \sim \text{Binomial}(n, V_{\pi_j, B, A}(s_i))$ from n resampling groups of size B . For two conditions c_1, c_2 , any policy-blind predictor $f(s) \in [0, 1]$ satisfies

the squared-loss lower bound

$$\begin{aligned} \frac{1}{2} \mathbb{E}_s[(f - V_1)^2 + (f - V_2)^2] \\ \geq \frac{1}{4} \mathbb{E}_s[(V_1 - V_2)^2], \end{aligned} \quad (2)$$

because the conditional optimum is $f^* = (V_1 + V_2)/2$ with residual $(V_1 - V_2)^2/4$. Our bound becomes non-trivial whenever two continuation conditions disagree on a non-vanishing prefix mass.

CPCC estimates a predictive distribution over $V_{\pi, B, A}(s)$ through three independent modules (Figure 2). A **shared frozen-LLM prefix encoder** E_θ produces a policy-independent prefix representation $h(s)$. A **comparable policy-behavior feature module** Φ produces a low-dimensional vector $\phi(s, \pi)$ whose dimensions stay stable across model families. A **Beta-Binomial calibration head** g_θ maps $(h(s), \phi(s, \pi), \log B)$ to the parameters of a Beta predictive distribution over V , supervised by bootstrap success counts at multiple budgets.

3.1 Shared Prefix Representation

A single frozen LLM encoder E_θ , shared across all continuation policies, produces the prefix representation $h(s)$. We apply the same instruction template used at rollout time, run one forward pass through E_θ , and pool the last-layer hidden states over the prefix tokens, concatenating two pooled

views:

$$h(s) = \begin{bmatrix} \text{LastTok}(E_\theta(s)) \parallel \\ \text{MeanPool}(E_\theta(s)) \end{bmatrix} \in \mathbb{R}^{2d}, \quad (3)$$

Last-token pooling captures the local committed state at the generation boundary (the position a continuation would be sampled from), while mean pooling captures globally diffuse information that survives averaging. Together the two views are complementary and inexpensive to combine.

We share the encoder rather than read each policy’s own hidden state, because per-policy hidden states do not register across families, and keep it frozen so that $\phi(s, \pi)$ remains the head’s only policy-dependent input.

3.2 Comparable Policy-Behavior Features

Our policy descriptor $\phi(s, \pi) \in \mathbb{R}^{19}$ carries the policy-conditioning signal and concatenates two complementary families. *Likelihood-side features* (9 dims) come from a single forward pass of the policy on the prefix, with no sampling, and describe what the policy thinks about its own next token. *Behavior-side features* (10 dims) come from B_{probe} unlabeled continuations sampled from the policy, and describe what the policy actually does over a continuation, which is the quantity downstream aggregation operates on. Every dimension in either family keeps a consistent operational definition across the tested model families, even when the raw scale differs across tokenizers; we do not claim these nineteen descriptors are complete or universally optimal.

Likelihood-side features. Let $s = (y_1, \dots, y_L)$ be the prefix and $p_\pi(\cdot \mid y_{<t})$ the policy’s next-token distribution at position t . We forward s through π once and extract four moments of the per-token negative log-likelihood $\ell_t(\pi) = -\log p_\pi(y_t \mid y_{<t})$,

$$\begin{aligned} \mu_\ell &= \frac{1}{L} \sum_{t=1}^L \ell_t, & \sigma_\ell^2 &= \frac{1}{L} \sum_{t=1}^L (\ell_t - \mu_\ell)^2, \\ \ell_L, & \text{PPL} = \exp(\mu_\ell), \end{aligned} \quad (4)$$

together with five next-token uncertainty features at the generation boundary $t = L + 1$,

$$\begin{aligned} H_\pi(s) &= -\sum_{v \in \mathcal{V}} p_\pi(v) \log p_\pi(v), \\ p^{(1)}, & p^{(2)}, \quad p^{(1)} - p^{(2)}, \quad \tilde{H}_\pi^{(5)}(s), \end{aligned} \quad (5)$$

where $p^{(k)}$ is the k -th largest entry of $p_\pi(\cdot \mid s)$ and $\tilde{H}_\pi^{(5)}$ is the entropy of the renormalized top-5 distribution. Although numerical scales depend on

tokenizer and vocabulary, each feature still measures how peaked or uncertain the policy is on the prefix.

Behavior-side features. Let $\{\hat{y}^{(b)}\}_{b=1}^{B_{\text{probe}}}$ be B_{probe} unlabeled continuations of s sampled from π , and let $a^{(b)} = \text{Ans}(\hat{y}^{(b)})$ be the extracted answer of the b -th continuation. Define the empirical answer distribution $\hat{q}(a) = \frac{1}{B_{\text{probe}}} \sum_b \mathbf{1}[a^{(b)} = a]$. We compute five answer-distribution-shape features,

$$\begin{aligned} u &= \frac{|\{a: \hat{q}(a) > 0\}|}{B_{\text{probe}}}, & m &= \max_a \hat{q}(a), \\ H_A &= -\sum_a \hat{q}(a) \log \hat{q}(a), & H_A^{\text{SC}}, \\ \text{Agr} &= \left(\frac{B_{\text{probe}}}{2}\right)^{-1} \sum_{b < b'} \mathbf{1}[a^{(b)} = a^{(b')}], \end{aligned} \quad (6)$$

three generation-regularity features (mean and standard deviation of completion length μ_L, σ_L and duplicate-completion rate d), and two answer-validity features (valid-answer rate v , refusal rate r). These features live in answer- and length-space rather than token-space, so they generalize across tokenizers and target what majority-vote aggregation acts on.

Probe rollouts add a real deployment cost of B_{probe} unlabeled rollouts per scored prefix, which we count as part of CPCC’s budget in any cost-aware comparison. Probes are not discarded after feature extraction: whenever the prefix is continued, they serve as the first B_{probe} votes of the final aggregation (Appendix K).

3.3 Per-Policy Normalization

Likelihood-side features differ in scale across model families because tokenizer granularities and vocabulary sizes differ; behavior-side features differ because completion-length conventions and refusal behaviors differ. To make ϕ comparable across policies, we normalize per policy:

$$\begin{aligned} \tilde{\phi}(s, \pi) &= \frac{\phi(s, \pi) - \mu_\pi}{\sigma_\pi}, \\ (\mu_\pi, \sigma_\pi) &\text{ from train prefixes.} \end{aligned} \quad (7)$$

For a new continuation policy at deployment, (μ_π, σ_π) are estimated from a small set of unlabeled probe prefixes on the new policy. Normalization lets the head treat “the policy is at the 90th-percentile-uncertain end of its own range” as a comparable feature across policies whose underlying NLL scales can differ by a factor of several. Without normalization, out-of-distribution descriptors land far outside the train support of the head and the head extrapolates poorly.

3.4 Beta–Binomial Calibration Head

We implement the head g_θ as a small multilayer perceptron with GELU activations and layer normalization, followed by a two-dimensional output passed through softplus to obtain non-negative Beta parameters:

$$(\alpha, \beta) = \text{softplus}(g_\theta([h(s) \parallel \tilde{\phi}(s, \pi) \parallel \log B])) + \epsilon, \quad (8)$$

with $\epsilon = 10^{-4}$ for numerical stability. We interpret (α, β) as the parameters of a Beta predictive distribution over V . We compute the predictive mean and variance as

$$\hat{V} = \frac{\alpha}{\alpha + \beta}, \quad \text{Var}(V) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}. \quad (9)$$

We expose both quantities to downstream applications: the mean as the calibrated success estimate, and the variance for uncertainty-aware decisions such as lower-confidence-bound pruning or adaptive budgeting.

Each training target is a bootstrap observation k out of n aggregations of size B from the rollout pool. With a conjugate Beta–Binomial form, the per-row negative log-likelihood admits a closed expression:

$$\begin{aligned} -\log p(k \mid n, \alpha, \beta) &= -\log \binom{n}{k} \\ &\quad -\log \mathcal{B}(k + \alpha, n - k + \beta) \\ &\quad + \log \mathcal{B}(\alpha, \beta), \end{aligned} \quad (10)$$

where $\mathcal{B}(\cdot, \cdot)$ denotes the Beta function, distinct from the inference budget B . Using the Beta–Binomial likelihood absorbs the heteroscedastic dependence of k/n on the budget n rather than treating k/n as a fixed regression target, and jointly encodes location and width through (α, β) so the head cannot trade calibration for sharper predictions.

3.5 Bootstrap Supervision and Training Protocol

For each (s_i, π_j) we run N rollouts, form n bootstrap groups of size B at every training budget, and record the success count k . Our head sees the budget through the $\log B$ input; for the evaluated budgets $\log B \in [0, \log 64] \approx [0, 4.2]$, a scale comparable to the layer-normalized activations, and every calibrator variant receives the same encoding. We restrict training to non-degenerate budgets,

since $B=N$ collapses to a single aggregate sample with $n=1$. Training and validation prefixes are partitioned by question rather than by row, and optimization uses AdamW with early stopping on validation Beta–Binomial NLL. Deployment-time scoring requires one encoder pass, B_{probe} probe rollouts under the target policy, and one head pass.

4 Experiments

4.1 Datasets and Baselines

Datasets. We train on GSM8K-train prefixes and evaluate on three held-out reasoning benchmarks: GSM8K-test, MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), and SVAMP (Patel et al., 2021). GSM8K (Cobbe et al., 2021) provides grade-school math word problems. MATH-500 is a harder out-of-distribution test set of competition-style problems, and SVAMP is an arithmetic-style word-problem benchmark used as an additional out-of-distribution set. We additionally use the GSM8K-train split for the policy-shift analysis in Section D. All datasets are evaluated with the same automatic answer verifier.

Baselines. Baselines fall into three groups according to their relationship to our target. **Standard rollout heuristics** use classical sampling or probability signals, namely majority / self-consistency confidence (Wang et al., 2023) and an answer-agreement heuristic computed from the rollout pool (Appendix I). **Direct prefix-success calibration baselines** share CPCC’s training data and target and isolate the calibrator design axis. A **Policy-Blind Prefix Calibrator** uses $g_\theta([h(s), \log B])$ and ignores the continuation policy. An **Explicit-Descriptor Calibrator** uses $g_\theta([h(s), z_\pi, \log B])$ with z_π concatenating a one-hot model family, $\log(\text{parameter count})$, temperature, top- p , and a one-hot aggregation id, so that categorical and numeric policy attributes enter the head directly. For the corresponding feature-source ablation, **CPCC+Descriptor** adds the same z_π on top of CPCC’s comparable scalar features. A **Raw Policy-Hidden Calibrator** replaces $h(s)$ with the policy’s own hidden state on the prefix, as a diagnostic negative control. A **Single-Policy PRM-style Calibrator** (Cobbe et al., 2021; Lightman et al., 2024; Park et al., 2025) trains a Policy-Blind head on labels from one fixed continuation policy and transfers to the others. We report this transfer diagnostic separately in Figure 5, not in Table 1. **External adjacent trace-confidence meth-**

Table 1: Main calibration comparison on Brier ($\downarrow$), ECE ($\downarrow$), and Spearman ($\uparrow$), mean $\pm$ std over three seeds. Raw Policy-Hidden is reported as a diagnostic and is excluded from the per-column best/second highlight; blue marks the best, red marks held-out-family collapse. Beta-Binomial NLL is in Appendix E.

Method	gsm8k_test	svamp	math500	heldout P2	heldout P3
Brier ($\downarrow$)					
Policy-Blind	0.112 $\pm$.004	0.106 $\pm$.002	0.158 $\pm$.002	0.115 $\pm$.004	0.143 $\pm$.003
Explicit-Descriptor	0.115 $\pm$.007	0.108 $\pm$.002	0.160 $\pm$.012	0.118 $\pm$.012	0.145 $\pm$.005
Raw Policy-Hidden	0.107 $\pm$.002	0.098 $\pm$.003	0.221 $\pm$.016	0.232 $\pm$.033	0.389 $\pm$.048
CPCC + Descriptor	0.092 $\pm$.003	0.081 $\pm$.001	0.118 $\pm$.003	0.098 $\pm$.001	0.131 $\pm$.004
CPCC (Ours)	0.097 $\pm$.006	0.082 $\pm$.002	0.125 $\pm$.003	0.093 $\pm$.010	0.127 $\pm$.002
ECE ($\downarrow$)					
Policy-Blind	0.055 $\pm$.010	0.091 $\pm$.006	0.103 $\pm$.010	0.081 $\pm$.007	0.120 $\pm$.011
Explicit-Descriptor	0.073 $\pm$.021	0.102 $\pm$.005	0.112 $\pm$.029	0.093 $\pm$.028	0.129 $\pm$.015
Raw Policy-Hidden	0.043 $\pm$.014	0.070 $\pm$.013	0.251 $\pm$.033	0.270 $\pm$.058	0.461 $\pm$.053
CPCC + Descriptor	0.066 $\pm$.010	0.070 $\pm$.008	0.076 $\pm$.009	0.097 $\pm$.002	0.137 $\pm$.012
CPCC (Ours)	0.071 $\pm$.015	0.067 $\pm$.010	0.089 $\pm$.016	0.086 $\pm$.022	0.130 $\pm$.006
Spearman ($\uparrow$)					
Policy-Blind	0.396 $\pm$.003	0.244 $\pm$.001	0.443 $\pm$.012	0.350 $\pm$.003	0.387 $\pm$.007
Explicit-Descriptor	0.422 $\pm$.018	0.287 $\pm$.041	0.447 $\pm$.018	0.351 $\pm$.010	0.398 $\pm$.008
Raw Policy-Hidden	0.433 $\pm$.005	0.361 $\pm$.004	0.457 $\pm$.012	-0.090 $\pm$.032	0.028 $\pm$.029
CPCC + Descriptor	0.506 $\pm$.009	0.420 $\pm$.010	0.519 $\pm$.016	0.434 $\pm$.010	0.516 $\pm$.005
CPCC (Ours)	0.466 $\pm$.006	0.374 $\pm$.006	0.502 $\pm$.011	0.429 $\pm$.003	0.518 $\pm$.004

ods, namely Self-Certainty (Kang et al., 2025), DeepConf (Fu et al., 2025), CISC (Taubenfeld et al., 2025), and TokUR (Zhang et al., 2026), target a different quantity than $V_{\pi,B,A}$ and are reported separately in Section 4.5.

4.2 Implementation Details

Six continuation policies span three model families: four vary the model at fixed decoding $T=0.7$, top- $p=0.95$ (P0 Qwen2.5-3B-Instruct (Yang et al., 2024), P1 Qwen2.5-7B-Instruct, P2 Llama-3.1-8B-Instruct (Team, 2024), P3 Mistral-7B-Instruct-v0.3 (Jiang et al., 2023)), and two vary temperature on Qwen2.5 backbones (P4 at $T=1.0$, P5 at $T=0.3$). Held-out-family analyses exclude P2 and P3 from training. For each prefix-policy pair we run $N=64$ vLLM rollouts with majority-vote aggregation (A = majority throughout; every empirical result in this paper is specific to this aggregation rule), and form $n=32$ bootstrap groups at each budget $B \in \{1, 4, 16\}$ for training plus the deterministic $B=64$ for evaluation, producing 9,855 prefixes and 3.79M continuations across four data splits. Our shared prefix encoder is a frozen Qwen2.5-3B-Instruct (bf16), and the head is a small MLP trained with AdamW. Probe rollouts reuse the first $B_{\text{probe}}=8$ of the N labelled rollouts. Each cell reports mean $\pm$ std over

three seeds. Full hyperparameters and the question-disjoint split protocol are in Appendix B.

4.3 Main Calibration Results

Table 1 reports Brier, ECE, and Spearman for CPCC against the four direct prefix-success calibration baselines defined in Section 4.1. Beta-Binomial NLL is in Appendix E.

Calibration error. The CPCC family gives the lowest Brier on every evaluation cell, improving on Policy-Blind by 11% to 25%, and the two variants divide the grid: CPCC+Descriptor leads the three seen-distribution cells (0.092, 0.081, 0.118), while CPCC leads both held-out-family cells (Llama 0.093, Mistral 0.127) and, between the two variants, has the better held-out ECE and NLL. On ECE, CPCC cuts the two out-of-distribution math cells by 13% to 26% over Policy-Blind (SVAMP 0.091 $\rightarrow$ 0.067, MATH-500 0.103 $\rightarrow$ 0.089). The pattern is consistent with Eq. 2: the benefit of policy conditioning is largest under policy or domain shift, where a policy-blind predictor must average over differing conditional success targets. By the same logic the in-distribution picture is mixed: on *gsm8k_test* Policy-Blind attains lower binned ECE (0.055 vs 0.071) while CPCC still improves Brier (0.112 $\rightarrow$ 0.097) and Spearman

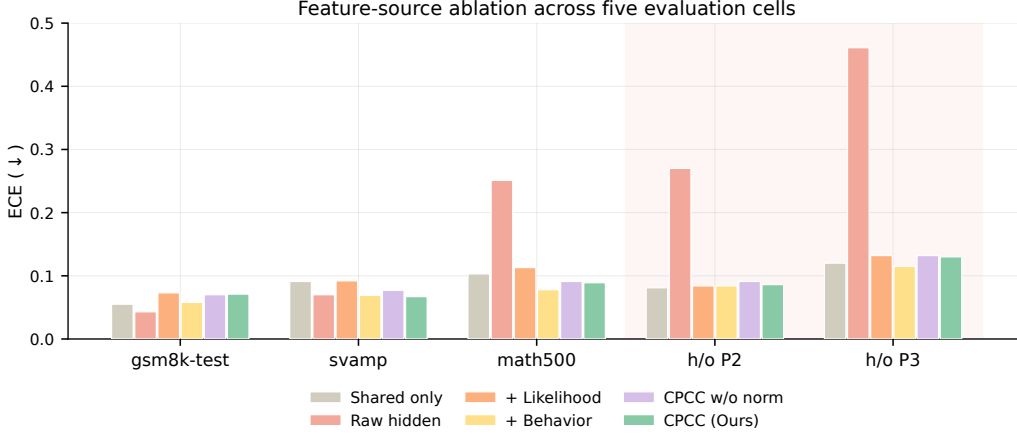


Figure 3: Feature-source ablation for CPCC, ECE ($\downarrow$) across five evaluation cells. Each bar is the mean over three seeds. Shaded red region marks the two held-out-family cells.

(0.396 $\rightarrow$ 0.466), so the claim this paper defends is robustness under shift, not uniform dominance of every in-distribution metric. In deployment terms the split yields a simple recommendation: CPCC for open-policy transfer, CPCC+Descriptor when the deployed policies are known and stay close to training.

Held-out ranking and collapse. Both CPCC variants lift held-out Spearman far above Policy-Blind (0.429 and 0.434 on Llama, 0.518 and 0.516 on Mistral, against 0.350 and 0.387), and lie within 0.005 of each other. Raw Policy-Hidden Calibrator remains competitive in distribution but collapses out of family (Spearman -0.090 on Llama, 0.028 on Mistral): a calibrator that reads each model’s own hidden state cannot register Llama representations onto Qwen or Mistral coordinates, so out-of-family inputs look like noise. Raw Policy-Hidden’s in-distribution Brier stays close to CPCC’s (e.g., 0.107 vs 0.097 on *gsm8k_test*) but roughly doubles to 0.232 and 0.389 on the two held-out cells, so in-distribution calibration does not predict held-out robustness. Across inference budgets, CPCC stays below Policy-Blind while Raw Policy-Hidden diverges on the hardest cell (Figure 4; GSM8K and SVAMP in Appendix G).

Single-policy transfer. A complementary diagnostic trains a Policy-Blind head on one fixed continuation policy in $\{P0, P1, P3\}$ and transfers to the other five on *gsm8k_test* (Figure 5): the proxy transfers reasonably from Qwen policies (ECE 0.06–0.09) but degrades when fit on Mistral (ECE 0.17), whereas CPCC, aggregating all policies in one comparable representation, gives competitive ECE

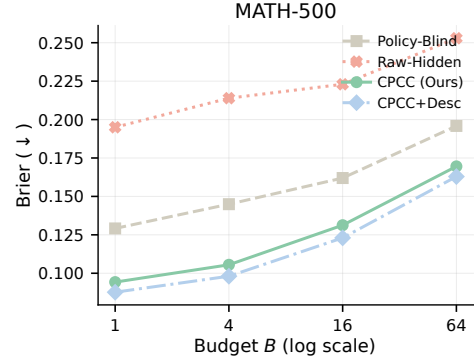


Figure 4: Brier ($\downarrow$) vs inference budget B on MATH-500. CPCC and CPCC+Descriptor stay below Policy-Blind at every budget; Raw Policy-Hidden diverges. GSM8K and SVAMP are in Appendix G.

(0.071) and the best Brier and Spearman.

4.4 Ablations and Design Analysis

We ablate the contribution of each source in $\phi(s, \pi)$ (Figure 3). Behavior-side features carry nearly all of the calibration signal: alone they recover most of the OOD ECE gain on SVAMP and MATH-500, a pattern consistent with majority-vote success depending directly on the continuation answer distribution. Likelihood-side features alone do not improve binned calibration over the shared encoder; their value shows up in risk ordering, where combining both sources gives the best selective risk on GSM8K and SVAMP (Table 2, Appendix K), a sampling-free complement to the behavior pool. **Per-policy normalization and probe coupling.** Replacing the normalized $\tilde{\phi}$ with raw ϕ (CPCC w/o norm) is roughly tied in distribution and slightly worse on OOD ECE (SVAMP 0.077 vs 0.067). Its clearest cost is stability on the held-out fam-

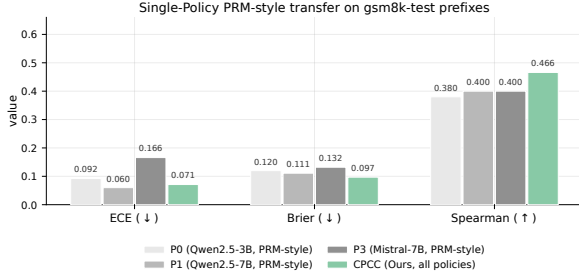


Figure 5: Single-Policy PRM-style transfer on *gsm8k_test* prefixes (proxy baseline inspired by PRM calibration, not an implementation of any specific published PRM). Mean over three seeds.

ily: across seeds, held-out Mistral Spearman drops from 0.518 ± 0.004 to 0.477 ± 0.043 , an order of magnitude larger variance. Per-policy normalization therefore functions less as a uniform accuracy knob and more as a mechanism that improves the stability of policy-feature coordinates under held-out-policy shift. Appendix N tests the transductive alternative of re-estimating (μ_π, σ_π) on unlabeled test prefixes and finds it unreliable under strong task shift. Behavior features reuse the first $B_{\text{probe}}=8$ labelled rollouts (Section 3.2). Probe features do not read correctness labels but are drawn from the same generation pool as the bootstrap targets.

Selective-inference utility. Downstream mechanisms such as continuation, pruning, and abstention consume the *ordering* of prefixes by predicted success. Table 2 evaluates this quantity directly as the area under the risk-coverage curve at $B=16$: prefixes are ranked by predicted success, and risk is the majority-vote error rate among the covered prefixes, averaged over continuation policies and prefix depths. CPCC lowers selective risk on all three datasets, by roughly 10% to 30% relative to Policy-Blind. This gain is complementary to the adaptive-budget null result of Appendix J: reallocating additional rollouts does not raise final majority-vote accuracy in these near-saturated pools, while the calibrated score still substantially improves the risk ordering that selective inference acts on. Appendix K reports the cost-accounted protocol, a zero-probe likelihood-only variant, and fixed-budget selection curves.

Three further robustness studies are reported in the appendix: prefixes generated by an unseen source family (Appendix L), sensitivity to the shared-encoder choice (Appendix M), and target-batch feature re-normalization (Appendix N). Un-calibrated rollout heuristics remain in Appendix I.

Table 2: Selective risk (area under the risk-coverage curve, $\downarrow$) for ranking prefixes by predicted success at $B=16$, averaged over continuation policies and prefix depths.

Method	GSM8K	SVAMP	MATH-500	Macro
Policy-Blind	0.081	0.074	0.420	0.191
CPCC	0.061	0.052	0.380	0.164

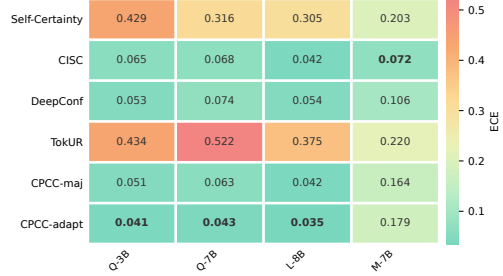


Figure 6: ECE ($\downarrow$) on SVAMP across four continuation policies (Qwen-3B/7B, Llama-3.1-8B, Mistral-7B); deeper teal = lower ECE (better), warmer red = higher (worse), bold = best per column. GSM8K / MATH-500 ECE and the full AUROC grid are in Appendix F.

4.5 Comparison with External Trace-Confidence Methods

Four external methods defined in Section 4.1 (Self-Certainty, CISC, DeepConf, TokUR) score complete continuations under a fixed generator rather than the policy-conditioned prefix success $V_{\pi,B,A}$. We treat them as adjacent contextual comparisons and evaluate them on a 300-prefix subset of each test set, with 16 rescored rollouts per prefix under each of four continuation policies (Qwen2.5-3B, Qwen2.5-7B, Llama-3.1-8B-Instruct, Mistral-7B-Instruct). CPCC enters the comparison as *CPCC-majority* @ $B=16$ (plain majority vote at the fixed budget) and *CPCC-adaptive* ($\tau=0.7$) (the smallest budget whose predicted Beta predictive mean exceeds τ , upper cap $B=64$). Figure 6 shows ECE on SVAMP, CPCC’s strongest dataset: CPCC-adaptive attains the lowest ECE on all three Qwen/Llama policies (0.041, 0.043, 0.035), lower than any external baseline on those columns, and only on the held-out Mistral-7B policy does CISC lead. GSM8K and MATH-500 ECE and the full AUROC grid (Appendix F) show the same pattern: across both metrics and all four policies a CPCC variant is best on 12 of 24 cells, and on seven of those the two variants occupy the top two ranks. CISC and DeepConf lead on Mistral-7B and on MATH-500 AUROC, where Mistral’s unusually strong token-level signal favors logprob-reading

Table 3: Three real held-out Mistral-7B prefixes ($B=16$). Predictions $\hat{V}$ closest to empirical V in bold; the CPCC column is shaded blue and the collapsed Raw-Hidden column red.

Question and partial reasoning (truncated)	V	Policy-Blind	Raw-Hidden	CPCC
<i>High-V (Lillian, 6 feeders, 20 birds each).</i> "... Lillian builds 3 feeders, buys 3 more, total 6 feeders. Each attracts 20 birds. Store-bought attract $3 \times 20 = 60$ birds..." (next step: add $3 \times 10 = 30$ self-made extras, gold 150).	1.00	0.23	0.08	0.99
<i>Mid-V (Jason, 240 min walking).</i> "To solve this problem, let's break it down step by step. Total time needed to break in the shoes: 240 minutes. Number of weeks Jason has: 3 weeks..." (next step: $3 \times 4 = 12$ days, then $240 / 12 = 20$, gold 20).	0.69	0.98	0.19	0.93
<i>Low-V (Bill, streaming-service pricing).</i> "... First 6 months at \$8/month, total \$48. Next 2 months (months 7 and 8) at the normal \$12/month..." (gold 284; prefix mis-counts the second tier as 2 months instead of 8, fatal).	0.00	0.67	0.96	0.17

baselines, and CPCC’s model-agnostic prefix encoder does not exploit this signal, an architectural trade-off for cross-policy comparability rather than a data-distribution gap. Self-Certainty and TokUR are poorly calibrated throughout.

4.6 Case Study

Table 3 traces three real prefixes from *gsm8k_test* through the three learned calibrators on the held-out Mistral-7B (P3) condition at $B=16$ (seed 0). CPCC is the closest of the three calibrators to the empirical V on all three prefixes, although the mid-range prefix remains the hardest to place. Policy-Blind, having averaged over the five seen policies during training, predicts 0.23 for the prefix that Mistral solves with probability 1.00 and predicts 0.67 for the prefix that Mistral never recovers. Raw Policy-Hidden produces almost the reverse of the empirical ordering: the high- V prefix is scored at 0.08 and the low- V prefix at 0.96. Per-prefix reversal mirrors the aggregate Spearman of 0.03 on held-out Mistral and illustrates the structural failure that Table 1 reports in aggregate.

5 Conclusion

Prefix confidence in LLM reasoning is not policy-invariant. Cross-family prefix ranking-flip rates of 35% to 49% for the strongest pair per dataset (30.1% to 50.5% over the full cross-family grid), together with the sharp held-out collapse of a hidden-state-based calibrator, indicate that prefix success depends on the continuation policy and inference setting rather than on the prefix alone. Under majority-vote aggregation, the CPCC family reduces Brier over Policy-Blind on every evaluation cell by 11% to 25%, lowers ECE by 13% to 26% on the OOD math cells, and cuts downstream

selective risk by 10% to 30%, while remaining competitive on the seen distribution. Raw-hidden conditioning is the direct calibrator that collapses out of family, while the CPCC variants attain the strongest held-out ranking, improving Spearman over Policy-Blind by 13% to 54% across cells.

Limitations

CPCC’s behavior-side features require B_{probe} unlabeled continuations of the target prefix at deployment, a compute cost we count as part of CPCC’s budget in any cost-aware comparison with response-level methods. Our experiments cover three model families (Qwen2.5, Llama-3.1, Mistral-7B) and six continuation policies, and all empirical evidence is obtained under majority-vote aggregation. The notation $V_{\pi, B, A}$ makes the aggregation rule an explicit argument because changing A changes the estimated target, but our results do not establish transfer to verifier-weighted or search-based aggregation; this is a limitation of the present evidence. Extending CPCC to such aggregation rules, search-time selection, or tool-using agents is left to future work.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and suggestions. This work was supported by the British Heart Foundation Manchester Research Excellence Award (RE/24/130017). The work leading to these results has been funded by the European Union under Grant Agreement No. 101289094 (project AIRIS). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HaDEA). Neither the

European Union nor the granting authority can be held responsible for them. We also acknowledge the CSF3 at the University of Manchester for providing GPU resources. Xin Zhang is supported by the UoM-CSC Joint Scholarship.

References

- Bradley C. A. Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. [Large language monkeys: Scaling inference compute with repeated sampling](#). *CoRR*, abs/2407.21787.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *CoRR*, abs/2110.14168.
- Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. [Detecting hallucinations in large language models using semantic entropy](#). *Nat.*, 630(8017):625–630.
- Yichao Fu, Xuwei Wang, Yuandong Tian, and Jiawei Zhao. 2025. [Deep think with confidence](#). *CoRR*, abs/2508.15260.
- Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. [Reasoning with language model is planning with world model](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 8154–8173. Association for Computational Linguistics.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the MATH dataset](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.
- Jianguo Huang, Zhiyi Lyu, Fuxiang Zhang, Yanchen Deng, Deheng Ye, and Bo An. 2026. Uncertainty-aware tree search for efficient llm reasoning.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *CoRR*, abs/2310.06825.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. [Language models \(mostly\) know what they know](#). *CoRR*, abs/2207.05221.
- Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. [Scalable best-of-n selection for large language models via self-certainty](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*.
- Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth A. Malik, and Yarin Gal. 2024. [Semantic entropy probes: Robust and cheap hallucination detection in llms](#). *CoRR*, abs/2406.15927.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. [Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. [Let’s verify step by step](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Andrey Malinin and Mark J. F. Gales. 2021. [Uncertainty estimation in autoregressive structured prediction](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Jingwei Ni, Ekaterina Fadeeva, Tianyi Wu, Mubashara Akhtar, Jiaheng Zhang, Elliott Ash, Markus Leippold, Timothy Baldwin, See-Kiong Ng, Artem Shelmanov, and Mrinmaya Sachan. 2025. [Reasoning with confidence: Efficient verification of LLM reasoning steps via uncertainty heads](#). *CoRR*, abs/2511.06209.
- Young-Jin Park, Kristjan Greenewald, Kaveh Alimohammadi, Hao Wang, and Navid Azizan. 2025. [Know what you don’t know: Uncertainty calibration of process reward models](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*.
- Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. [Are NLP models really able to solve simple math word problems?](#) In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2080–2094. Association for Computational Linguistics.

Isha Puri, Shivchander Sudalairaj, Guangxuan Xu, Abhishek Bhandwal, Kai Xu, and Akash Srivastava. 2025. [Rollout roulette: A probabilistic inference approach to inference-time scaling of llms using particle-based monte carlo methods](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. [Scaling LLM test-time compute optimally can be more effective than scaling model parameters](#). *CoRR*, abs/2408.03314.

Zeen Song, Zihao Ma, Wenwen Qiang, Changwen Zheng, and Gang Hua. 2026. [Adaptive uncertainty-aware tree search for robust reasoning](#). *CoRR*, abs/2602.06493.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. [Confidence improves self-consistency in llms](#). In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pages 20090–20111. Association for Computational Linguistics.

Llama Team. 2024. [The llama 3 herd of models](#). *CoRR*, abs/2407.21783.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. [Qwen2.5 technical report](#). *CoRR*, abs/2412.15115.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. [Tree of thoughts: Deliberate problem solving with large language models](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tunyu Zhang, Haizhou Shi, Yibin Wang, Hengyi Wang, Xiaoxiao He, Zhuowei Li, Haoxian Chen, Ligong Han, Kai Xu, Huan Zhang, and 1 others. 2026. [Tokur: Token-level uncertainty estimation for large language model reasoning](#). In *International Conference on Learning Representations*, volume 2026, pages 52910–52942.

A Proof of the Squared-Loss Lower Bound

We restate and prove the squared-loss lower bound for any policy-blind predictor used in Section 3 (Eq. 2).

Proposition 1 (Squared-loss lower bound for policy-blind prediction). *Let $V_1, V_2 : \mathcal{S} \rightarrow [0, 1]$ be two policy-conditioned prefix-success functions under continuation conditions c_1 and c_2 , and let $\mathcal{P}$ denote any probability distribution over prefixes $s \in \mathcal{S}$. For any policy-blind predictor $f : \mathcal{S} \rightarrow [0, 1]$,*

$$\begin{aligned} \frac{1}{2} \mathbb{E}_{s \sim \mathcal{P}} [(f(s) - V_1(s))^2 + (f(s) - V_2(s))^2] \\ \geq \frac{1}{4} \mathbb{E}_{s \sim \mathcal{P}} [(V_1(s) - V_2(s))^2]. \end{aligned} \quad (11)$$

Equality is attained, uniquely, at $f^*(s) = \frac{1}{2}(V_1(s) + V_2(s))$.

Proof. Fix $s \in \mathcal{S}$ and write $a := V_1(s)$, $b := V_2(s)$, $x := f(s)$. We write the pointwise squared error as

$$\begin{aligned} \frac{1}{2} [(x - a)^2 + (x - b)^2] \\ = x^2 - (a + b)x + \frac{1}{2}(a^2 + b^2). \end{aligned}$$

Completing the square in x gives

$$\begin{aligned} \frac{1}{2} [(x - a)^2 + (x - b)^2] \\ = \left(x - \frac{a+b}{2}\right)^2 + \frac{1}{4}(a - b)^2. \end{aligned}$$

Taking the expectation over $s \sim \mathcal{P}$,

$$\begin{aligned} \frac{1}{2} \mathbb{E}_s [(f - V_1)^2 + (f - V_2)^2] \\ = \mathbb{E}_s \left[\left(f(s) - \frac{V_1(s) + V_2(s)}{2}\right)^2 \right] \\ + \frac{1}{4} \mathbb{E}_s [(V_1 - V_2)^2]. \end{aligned}$$

Non-negativity of the first term on the right, with equality iff $f(s) = \frac{1}{2}(V_1(s) + V_2(s))$, proves both the lower bound and the equality condition. $\square$

Reading. Residual term $\frac{1}{4} \mathbb{E}_s [(V_1 - V_2)^2]$ on the right of Eq. 11 depends only on how much the two continuation conditions disagree pointwise, and is

Algorithm 1 CPCC training

Require: Problems $\mathcal{D}$, source policy π_0 , continuation policies Π , budgets $\mathcal{B}$, bootstrap repeats n , probe budget B_{probe}

- 1: **for** $x \in \mathcal{D}$ **do**
 - 2: Generate $y \sim \pi_0(\cdot \mid x)$; split y into prefixes $\{s_t\}$
 - 3: **for** $\pi \in \Pi$ **do**
 - 4: Roll out N continuations from s_t under π ; compute ϕ from likelihoods and first B_{probe} rollouts
 - 5: **for** $B \in \mathcal{B}$ **do**
 - 6: n bootstrap groups of size B ; count correct aggregations $\rightarrow k$
 - 7: **end for**
 - 8: **end for**
 - 9: **end for**
 - 10: Compute per-policy (μ_π, σ_π) on train prefixes
 - 11: Train g_θ to minimize $\sum -\log p(k \mid n, \alpha, \beta)$ with $(\alpha, \beta) = g_\theta([h(s), \phi(s, \pi), \log B])$
-

in particular independent of f . A policy-blind predictor is therefore bounded below by a quantity non-vanishing on cross-family pairs (Appendix D). Our bound therefore motivates treating the continuation policy as an explicit input to the calibrator. Our argument is stated only for squared loss and does not constitute a universal impossibility result for arbitrary scoring rules.

B Algorithm and Implementation Notes

Our shared encoder is bf16 and runs on a single A100 or H200 GPU. Probe-rollout features use the same generation backend as labelled rollouts but do not require correctness labels. In our experiments we reuse the first $B_{\text{probe}}=8$ of the $N=64$ rollouts so probe statistics and main rollouts share the same generation. Training data restricts to budgets $\{1, 4, 16\}$ because the largest budget $B=64$ yields $n=1$ bootstrap group (deterministic), making it an unsuitable noisy target.

C Dataset and Rollout Statistics

Table 4 summarizes the prefix counts and total rollout budget per split used throughout our experiments.

We extract prefixes by greedy decoding from a base policy (Qwen2.5-3B-Instruct, $T=0$) and slicing at sentence and step boundaries, capped at three prefixes per problem, with prefixes that already con-

Dataset	Problems	Prefixes	Rollouts
gsm8k_train	2000	5,988	5,988·6·64
gsm8k_test	500	1,500	1,500·6·64
math500	500	1,471	1,471·6·64
svamp	300	896	896·6·64
Total	3300	9,855	3,784,320

Table 4: Prefix counts and rollout budget per dataset. Prefixes are step-level slices of a greedy source trace (max three per problem) with answer-bearing prefixes filtered out. All rollouts use vLLM with majority-vote aggregation.

tain a parseable final answer filtered out. We apply the same protocol to all four datasets. Verifier behaviour is sanity-checked under both a permissive and a strict math extractor, and the conclusions in Sections D–4.3 are stable under both.

D Policy Shift and Ranking Flip

For each dataset we extract step-level prefixes from a greedy reasoning trace, label every (prefix, policy) pair with 64 rollouts at $B=1$, and compute four statistics per cross-family policy pair: the mean absolute success-rate gap, Spearman and Kendall rank correlations, the pairwise ranking-flip rate on the non-trivial subset $\{s : 0.2 \leq V_{\pi_i}(s), V_{\pi_j}(s) \leq 0.8\}$, and the fraction of prefixes with $|V_{\pi_i} - V_{\pi_j}| \geq 0.4$.

Table 5 reports the strongest cross-family pair per data split. Mistral-7B (P3) is consistently the most distinctive continuation policy. Mean absolute gap with Qwen-7B is 0.18 to 0.26, and pairwise ranking-flip rate on the non-trivial subset reaches 35 to 49% for the strongest pair per dataset. Fraction of prefixes with $|\Delta V| \geq 0.4$ ranges from 13% to 28%. Table 6 gives the full pairwise grid, where flip rates range from 30% to 51% (maximum 50.5%, SVAMP P2×P5). Combined with the squared-loss bound of Eq. 2, these magnitudes leave non-trivial irreducible risk for any policy-blind predictor.

Dataset	Pair	mean $ \Delta V $	S_ρ	Flip(%)	$\geq .4(\%)$
gsm8k_train	P3×P5	0.224	0.49	41.1	23.8
gsm8k_test	P3×P5	0.211	0.45	42.2	22.4
math500	P3×P5	0.184	0.42	35.1	18.4
svamp	P1×P3	0.258	0.55	48.9	27.5

Table 5: Strongest cross-family policy pair per dataset, $B=1$, non-trivial subset.

Table 6: Cross-family policy-pair prefix ranking-flip rate (%) on the non-trivial subset $\{0.2 \leq V \leq 0.8\}$ at $B=1$, for every cross-family pair and every data split. Strongest pair per row in bold. Range: 30.1% to 50.5%.

Dataset	P0×P2	P0×P3	P1×P2	P1×P3	P2×P3	P2×P4	P2×P5	P3×P4	P3×P5
GSM8K-train	37.6	41.5	37.1	42.4	39.6	39.0	37.6	39.7	41.1
GSM8K-test	33.6	39.2	39.9	42.5	41.9	40.2	39.7	39.4	42.2
MATH-500	38.7	38.3	34.6	41.2	36.8	37.1	34.4	42.0	35.1
SVAMP	30.1	32.6	46.0	48.9	33.4	30.2	50.5	38.0	41.2

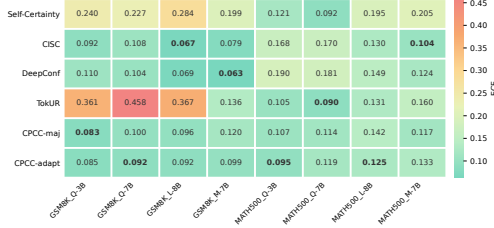


Figure 7: ECE (↓) on GSM8K and MATH-500. Deeper teal cells = lower ECE (better); warmer red cells = higher ECE (worse). Bold marks the best method per column.

E Main Calibration Results: NLL

Body Table 1 reports Brier, ECE, and Spearman. We report the same five direct prefix-success calibration baselines on the same evaluation cells in terms of Beta-Binomial NLL (↓, Table 7).

Reading. The division of labor mirrors the body table: CPCC+Descriptor obtains the best NLL on the three seen-distribution cells, CPCC on held-out Llama, and Policy-Blind on held-out Mistral, where predictive densities of all conditioned methods widen. Raw Policy-Hidden Calibrator’s NLL roughly doubles on the held-out-family cells, mirroring the Brier, ECE, and Spearman collapse reported in Body Table 1.

F Full External-Baseline Heatmaps

Figure 6 shows SVAMP ECE. Figure 7 gives ECE on GSM8K and MATH-500, and Figure 8 gives AUROC over all three datasets. CPCC leads ECE on the Qwen cells of GSM8K and on the Qwen2.5-3B and Llama cells of MATH-500, and AUROC on the Qwen cells of GSM8K and the Qwen/Llama cells of SVAMP, while CISC and DeepConf lead on the held-out Mistral-7B policy and on MATH-500 AUROC, consistent with the SVAMP pattern in the body.

G Brier Across Inference Budgets

Figure 9 traces Brier as a function of the inference budget $B \in \{1, 4, 16, 64\}$ for the four direct cali-

brators on GSM8K-test and SVAMP, complementing the MATH-500 panel in the body (Figure 4). CPCC and CPCC+Descriptor sit below Policy-Blind across every budget on both sets.

H Deployment Cost

At inference, scoring a new prefix under a target continuation policy requires three things: (i) one forward pass through the shared frozen encoder, (ii) B_{probe} continuation rollouts under the target policy to populate the behavior-side features, and (iii) a forward pass through the small head. Our encoder pass is independent of the policy and is shared across all candidate policies. Our behavior probe is the dominant runtime cost. It is the price of cross-family comparability, and it is amortized over downstream selection or budget-allocation decisions that the calibrator informs.

I Uncalibrated Rollout Heuristics

We compare against two uncalibrated rollout heuristics that are computable directly from the rollout pool, without a learned head. *Majority / Self-Consistency Heuristic* uses the within-budget majority-answer fraction as a confidence score (Wang et al., 2023). *Answer-Agreement Heuristic* uses the mean pairwise answer-agreement rate across the B rollouts, a behavior-side feature that we found on held-out Spearman to be the most informative single scalar in our ϕ . We list it as a heuristic separately from the standard log-likelihood and predictive-entropy line (Malinin and Gales, 2021; Kadavath et al., 2022), noting that mean pairwise answer agreement is not a log-likelihood or predictive-entropy quantity. On *gsm8k_test* ($B=4$), the Majority / SC heuristic obtains ECE 0.074 and Spearman 0.621, and the Answer-Agreement heuristic obtains ECE 0.266 and Spearman 0.776. Strong Spearman from the Answer-Agreement heuristic confirms that probe-rollout statistics carry a strong ranking signal, consistent with their role in ϕ inside CPCC. Both heuristics are sharply miscalibrated

Table 7: Main calibration comparison: Beta-Binomial NLL ($\downarrow$). Methods and cells match Table 1. Mean $\pm$ std over seeds $\{0, 1, 2\}$. Raw Policy-Hidden is a diagnostic and excluded from the highlight, as in Table 1; light-blue cells mark the best remaining value per column, light-red cells mark held-out-family collapse.

Method	gsm8k_test	svamp	math500	heldout P2	heldout P3
Policy-Blind	1.247 $\pm$.021	1.232 $\pm$.024	1.701 $\pm$.019	1.193 $\pm$.012	2.156 $\pm$.061
Explicit-Descriptor	1.298 $\pm$.090	1.295 $\pm$.047	1.789 $\pm$.164	1.213 $\pm$.068	2.197 $\pm$.081
Raw Policy-Hidden	1.202 $\pm$.012	1.138 $\pm$.034	1.941 $\pm$.036	2.253 $\pm$.241	3.791 $\pm$.444
CPCC + Descriptor	1.218 $\pm$.011	1.136 $\pm$.019	1.655 $\pm$.010	1.219 $\pm$.054	2.429 $\pm$.118
CPCC (Ours)	1.248 $\pm$.001	1.150 $\pm$.032	1.689 $\pm$.018	1.138 $\pm$.048	2.266 $\pm$.045

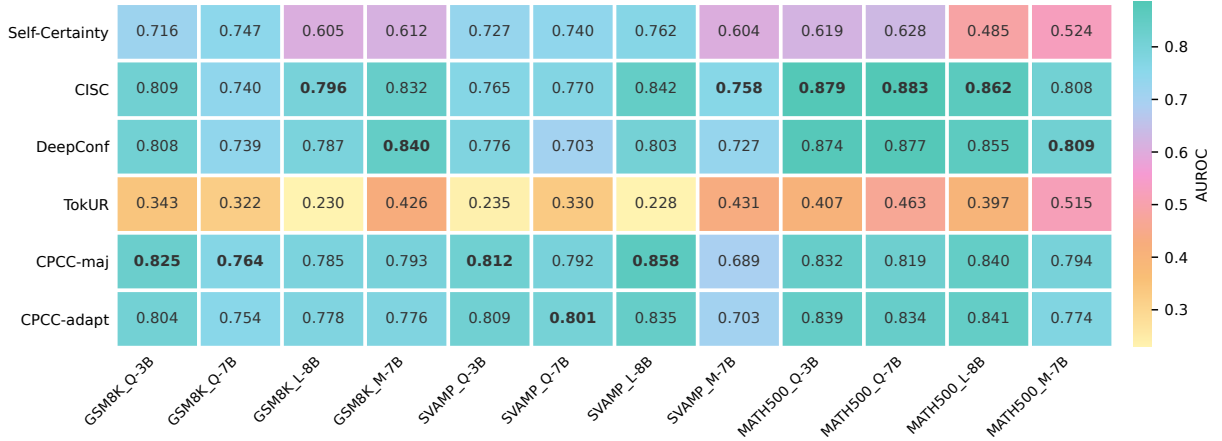


Figure 8: AUROC ($\uparrow$) over all three datasets $\times$ four policies. Deeper teal cells = higher AUROC (better); yellow / pink cells = lower (worse). Bold marks the best method per column.

as probabilistic predictors. ECE for the Answer-Agreement heuristic is $4\times$ CPCC’s on the same cell. CPCC preserves the ranking signal of the scalar features and adds the calibration of a Beta-Binomial head trained on bootstrap targets.

J Adaptive Budget Allocation (Null Result)

For each test set we sweep the threshold τ of the adaptive rule that selects the smallest budget B with predicted Beta predictive mean $\geq \tau$ (denoted *CPCC-adaptive* (τ sweep) in Figure 10), and we additionally evaluate a lower-confidence-bound variant (*CPCC-LCB*) that subtracts λ times the predictive standard deviation from the mean before applying the threshold; we report $\lambda=0.5$.

Across thresholds the adaptive policy traces an interpolant between the uniform- B Pareto points (Figure 10). On in-distribution *gsm8k_test* the adaptive rule slightly underperforms the corresponding uniform budget at matched average B , while on the harder *math500* the curves overlap. Our adaptive setting therefore acts as a downstream sanity check that the predicted Beta predictive distribution orders prefixes meaningfully, rather than

an end-to-end accuracy gain.

K Cost-Aware Selective Continuation

Selective continuation asks, within each (dataset, continuation policy, prefix depth) cell, which of the N prefixes are worth completing under a fixed roll-out budget. Prefixes are ranked by the calibrator’s predicted success at $B=16$; a continued prefix is answered by majority vote over its first 16 rollouts with numerically canonicalized answers and ties counted as failures. Cost accounting separates the two feature regimes. Methods without probe rollouts (Policy-Blind and a likelihood-only variant of CPCC) spend $16K$ rollouts to continue K prefixes, and unselected prefixes remain unanswered. Probe-based methods spend $8N$ probe rollouts up front, reuse those probes as the first eight votes wherever the prefix is continued, and answer unselected prefixes by majority vote over the eight probes alone. A uniform reference answers every prefix at $B=m$ for the same total budget $T=mN$.

Three readings follow. First, the likelihood-only variant needs no sampled probes and improves AURC only mildly over Policy-Blind, while its fixed-budget solved fraction is statistically indis-

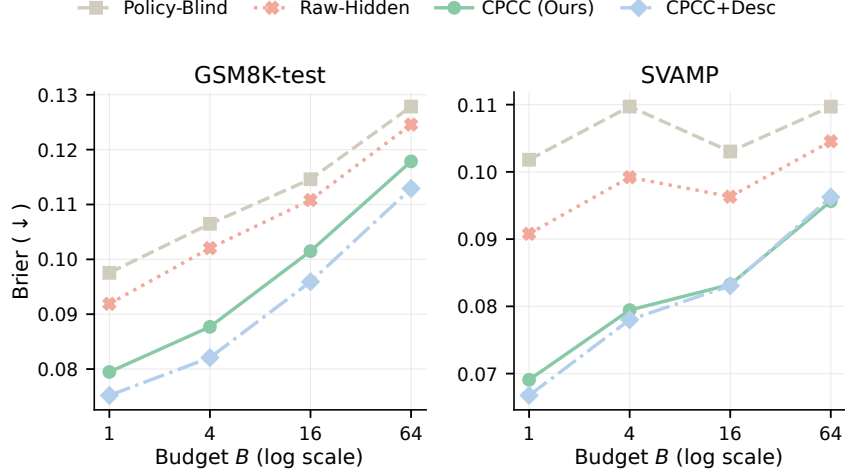


Figure 9: Brier ($\downarrow$) versus inference budget B for four direct calibrators on GSM8K-test and SVAMP (the MATH-500 panel is in the body, Figure 4). CPCC and CPCC+Descriptor sit below Policy-Blind at every budget.

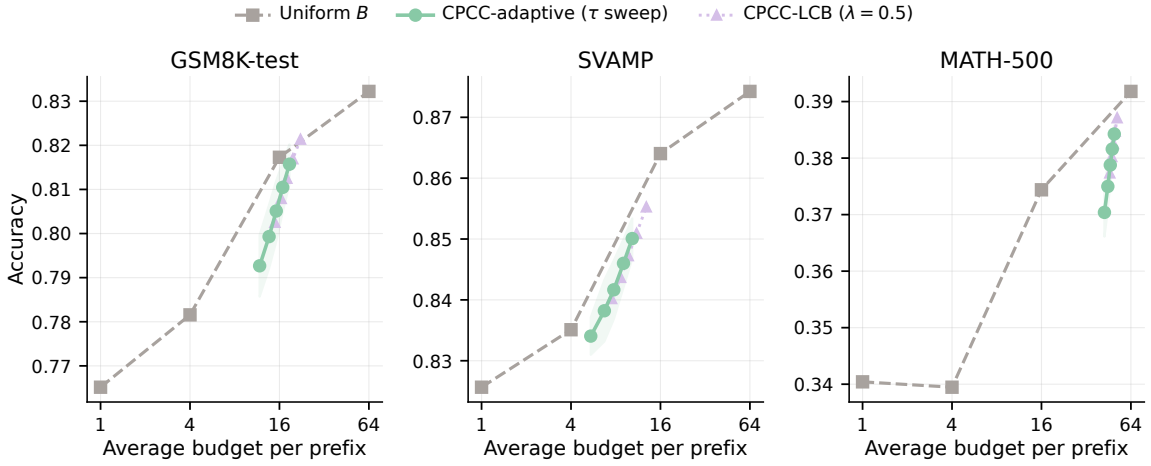


Figure 10: Adaptive-budget Pareto. Accuracy versus average per-prefix budget for uniform $B \in \{1, 4, 16, 64\}$ (grey dashed) and two CPCC-driven adaptive rules (CPCC-adaptive with τ sweep, CPCC-LCB at $\lambda=0.5$). Adaptive curves trace an interpolant of the uniform- B Pareto points and do not strictly dominate them.

Table 8: Selective risk (AURC $\downarrow$) by feature source, averaged over continuation policies and prefix depths. Probe count is the number of sampled continuations charged per scored prefix.

Method	Probes	GSM8K	SVAMP	MATH-500	Macro
Policy-Blind	0	0.081	0.074	0.420	0.191
Likelihood-only	0	0.078	0.069	0.413	0.187
Answer-agreement	8	0.073	0.058	0.392	0.175
Behavior-only	8	0.062	0.056	0.369	0.163
CPCC	8	0.061	0.052	0.380	0.164

tinguishable from Policy-Blind (the difference is about 0.1 percentage points and the problem-level bootstrap interval crosses zero): the cross-policy downstream signal is carried by the behavior-side features, so the probe cost buys the ranking gain rather than being overhead on top of it. Second, at $T=8N$ the probe-based configuration answers ev-

Table 9: Solved fraction of all prefixes under a fixed total budget $T = mN$, macro-averaged over datasets, policies, and depths.

Method	$m=8$	$m=10$	$m=12$	$m=14$	$m=16$
Policy-Blind	0.407	0.491	0.567	0.633	0.689
Likelihood-only	0.408	0.491	0.569	0.634	0.689
CPCC	0.677	0.677	0.680	0.683	0.689
Uniform $B=m$	0.677	0.680	0.685	0.686	0.689

ery prefix from its probes and solves 27.0 percentage points more of the pool than Policy-Blind selection (cluster-bootstrap 95% interval $[26.0, 28.2]$), an advantage that shrinks monotonically to zero at $T=16N$; at $T=8N$ no top-up selection occurs, so this row reflects the value of reusing probes as votes rather than of the calibrated ranking itself. Third, allocation rules that answer all prefixes differ by at

most 0.7 percentage points from the uniform reference because majority voting is near-saturated beyond eight rollouts on these pools, which is the same saturation that limits the adaptive allocator of Appendix J; the robust benefit of calibration is the risk ordering of Table 2, not raw accuracy on saturated pools.

L Prefixes from an Unseen Source Family

The main experiments hold the prefix generator fixed (Qwen2.5-3B, greedy) to isolate continuation-policy shift. This appendix varies the generator instead. For 335 GSM8K-test questions we generate prefixes with greedy Mistral-7B-Instruct-v0.3, keep one mid-trace prefix per question (the split point with fraction closest to 0.5; prefixes containing an explicitly committed answer are filtered), and label them with 64 rollouts under each of the six continuation policies. The same mid-trace selection applied to the Qwen-source pool on the same questions yields a paired comparison. Calibrators are the TC1 models trained purely on Qwen-source GSM8K-train prefixes; no Mistral-source prefix is seen in training. The Mistral-source pool is intrinsically harder (source accuracy 35.8%), which lowers absolute calibration for both methods.

Table 10: Paired source-shift comparison on GSM8K-test (335 questions, one mid-trace prefix each, all budgets). Excl. P3 removes the continuation policy that shares the source family.

Method	Prefix source	Brier ↓	Spearman ↑
Policy-Blind CPCC	Qwen	0.104	0.332
	Qwen	0.085	0.452
Policy-Blind CPCC	Mistral	0.243	0.238
	Mistral	0.219	0.389
Policy-Blind CPCC	Mistral, excl. P3	0.218	0.245
	Mistral, excl. P3	0.190	0.449

CPCC keeps lower Brier and substantially higher Spearman on the unseen source family, and the ranking advantage grows from 0.120 on Qwen-source prefixes to 0.151 on Mistral-source prefixes and 0.203 once P3 is excluded. Absolute calibration degrades for both methods, indicating that source-generator shift is an additional distribution axis beyond continuation-policy shift; the paired design controls for pool difficulty when comparing the two calibrators.

M Shared-Encoder Sensitivity

The shared prefix encoder is not the continuation policy; its role is to place every prefix in one coordinate system. To test sensitivity to this choice we repeat the full comparison of CPCC against Policy-Blind under four frozen encoders from three model families, retraining each head over three seeds on identical labels and evaluation cells.

Table 11: Encoder sensitivity: mean improvement of CPCC over Policy-Blind across the five evaluation cells (three seeds per head). Negative Δ Brier and positive Δ Spearman favor CPCC.

Shared encoder	Mean Δ Brier ↓	Mean Δ Spearman ↑
Qwen2.5-3B	−0.022	+0.094
Qwen2.5-7B	−0.019	+0.066
Llama-3.1-8B	−0.031	+0.081
Mistral-7B	−0.021	+0.055

Across the four encoders the direction favors CPCC in all 40 three-seed mean comparisons (five Brier and five Spearman cells per encoder); we read these as a stable direction rather than 40 independent significance tests. Absolute levels vary moderately, and the default Qwen2.5-3B encoder is the cheapest while remaining among the strongest, showing that the CPCC advantage is not specific to one shared-encoder backbone.

N Target-Batch Feature Re-Normalization

Per-policy normalization statistics (μ_π, σ_π) are estimated on training prefixes. A transductive alternative re-estimates them on a small batch of unlabeled test-domain questions. We evaluate the same trained calibrator under both statistics: for each evaluation set, K adaptation questions are drawn problem-disjoint from the evaluation remainder ($K \in \{32, 128\}$, three disjoint draws, three model seeds), and the paired change from switching to target-batch statistics is recorded on the identical remainder.

Re-estimation is inert in distribution and mildly improves SVAMP ECE, but on MATH-500 the calibration metrics (NLL, Brier, ECE) degrade in every seed and adaptation draw at both K , while rank correlation rises slightly. This pattern is consistent with re-centering removing cross-task scale information used by the trained head, so that shifted prefixes are scored as if they were average training-domain prefixes. Training-prefix statistics are there-

Table 12: Paired effect of target-batch re-normalization (target minus train statistics; negative ECE and Brier are improvements).

Evaluation set	K	Δ Brier	Δ ECE	Δ Spearman
GSM8K-test	32	-0.001	-0.001	-0.003
GSM8K-test	128	-0.001	-0.002	+0.003
SVAMP	32	+0.000	-0.003	-0.010
SVAMP	128	+0.000	-0.005	-0.010
MATH-500	32	+0.028	+0.078	+0.029
MATH-500	128	+0.022	+0.069	+0.032

fore the safer default under task-domain shift; re-estimating statistics remains necessary only when onboarding a new continuation policy, for which no training statistics exist.