SAME BELIEFS, DIFFERENT POLICIES: LEARNING REUSABLE REPRESENTATIONS FOR EARLY STOPPING IN LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning LLMs often continue reasoning after reaching a correct answer, wasting computation and sometimes changing it to an incorrect one. Deciding when to stop is a metareasoning problem: the model must weigh what further reasoning could achieve against its cost, given what it can already answer. We introduce BELT (BELief-based Termination), which learns beliefs about current and future answers from a frozen LLM's hidden states, then fits a linear stopping policy. We show that predicting reasoning outcomes yields reusable features for stopping control. Changing the token price changes the value of these outcomes, but leaves their prediction targets unchanged, so only the policy needs refitting. Under common policy fitting, belief-trained representations yield higher stopping utility than raw or utility-trained features. Averaged over five out-of-distribution math benchmarks, BELT reduces generated tokens by 66% on Qwen3-1.7B (Qwen Team, 2025) (69% at a one-point accuracy tolerance) and 53% on OLMo-3-7B-Think (Team Olmo, 2025) at matched test-frontier accuracy relative to natural completion. The same math-trained probe transfers to coding without code-side training. In both domains, its accuracy-cost trade-offs are competitive with or better than those of end-to-end controllers. On Qwen3, it uses $6.3\times$ fewer controller parameters than the strongest baseline and $\sim 25\times$ less training compute across five token prices.

1 INTRODUCTION

Reasoning large language models (LLMs) often overthink: they keep reasoning after the answer has stopped improving (Chen et al., 2025; Yang et al., 2025). Yet deciding when to stop requires more than knowing how likely the current answer is to be correct. A model may be uncertain about its current answer yet close to finding a better one. The same uncertainty may also occur when further reasoning is unlikely to help (Muennighoff et al., 2025; Ni et al., 2026b). Even a correct answer can be overturned by continued reasoning (Zhang et al., 2025). These situations call for different stopping decisions. Stopping is therefore a metareasoning problem. The model must assess what further computation is likely to achieve and whether that improvement is worth its cost (Russell & Wefald, 1991; Sabbata et al., 2025).

We separate this problem into learning to predict the outcomes of answering now or continuing, and learning to act on those predictions. The decision depends on the trade-off between accuracy and computation (Wang et al., 2024). For a frozen reasoner with a fixed decoding rule, changing the token price changes how outcomes are valued, but not the outcome prediction targets at a given reasoning prefix. The predictive representation can therefore be reused across prices instead of being learned anew for each stopping objective.

We study whether learning to predict reasoning outcomes yields effective, reusable representations for stopping control. We introduce BELT (BELief-based Termination), which trains a compact probe (the BELT probe) (Shelmanov et al., 2025; Ni et al., 2026a) on a frozen LLM's hidden states. The probe learns a control representation by jointly predicting four outcomes. We start with two reference outcomes: correctness when answering now (*Now*, Zhang et al., 2025; Yang et al., 2026; Hosseini et al., 2026) and at natural completion (*End*, Snell et al., 2025). But continuing preserves

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

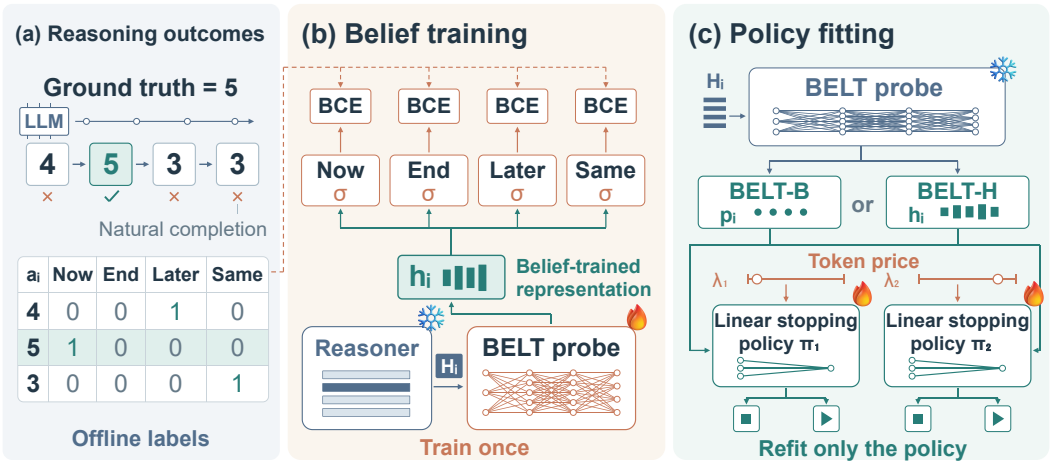

Figure 1: **BELT overview.** (a) Label offline traces to supervise outcome beliefs. (b) Jointly train the BELT probe with four objectives of outcome prediction. (c) Fit stopping policies on frozen features for different token prices λ. One representation for all preferences.

the option to stop before completion. We therefore also predict whether a correct answer will be available at a later boundary (*Later*, Wang et al., 2026a), even if the final answer is wrong. Finally, stopping may preserve the natural-completion answer even when its correctness is uncertain. We predict this agreement (*Same*, Wu et al., 2025). We call these predicted outcome probabilities *beliefs*.

During belief training, BELT learns these predictions from offline outcome labels, independently of the token price (Fig. 1). During policy fitting, it freezes the probe and fits a linear stopping policy to the sequential stopping objective (Ehab et al., 2026). The policy uses either the explicit beliefs (BELT-B) or the probe's learned representation (BELT-H). At each boundary, it decides whether to stop and answer or continue and reassess at the next boundary. The same probe supports a new token price by refitting only this small policy. We use this separation to test stopping performance and adaptation cost. We make two contributions.

(1) BELT learns reusable representations for stopping control. It combines a jointly supervised probe with a small linear policy that can be refitted for different token prices. On Qwen3, BELT-H saves 68.82% of generated tokens on out-of-distribution mathematics, averaged across datasets, at a test-frontier accuracy tolerance of one percentage point relative to natural completion, 2.2 pp more than OSP. On OLMo3, the same training recipe lets BELT-H save 8.1 pp more tokens than OSP-1 at the same tolerance. The math-trained probe also transfers to LiveCodeBench Pre without code-side training, where BELT-H saves 14.5 pp more than OSP (§ 5). Probe reuse lowers the estimated controller-training cost of covering five token prices by $25.5\times$ relative to OSP (§ 6.4).

(2) Comprehensive analyses clarify how outcome prediction supports stopping control. Predicting reasoning outcomes provides a useful basis for learning reusable control representations (§ 6.1). Target and input ablations distinguish the value of learning to predict an outcome from the value of directly using that prediction for decisions (§ 6.2). Policy diagnostics illustrate how changing token prices can be accommodated by reweighting the same beliefs (§ 6.3).

2 Background and Problem Formulation

Problem formulation. We study the metareasoning decision of whether to continue an LLM's reasoning or stop and request an answer. For a trajectory τ, let $i = 1, \ldots, T-1$ index candidate stopping boundaries and let T denote natural completion without early stopping. At boundary i, the controller observes H_i, the sequence of LLM hidden states at a selected layer up to that boundary. Stopping yields answer a_i, correctness $y_i \in \{0, 1\}$ and total token cost C_i. Continuing advances to the next boundary. Reaching T yields the final answer a_T, correctness y_T and total cost C_T.

A *stopping policy* with parameters ξ assigns the conditional probability $\pi_i = \Pr(\text{stop at } i \mid \text{reached } i, H_i; \xi)$. Its reach probability S_i and stopping probabilities ρ_i are

$$S_i = \prod_{j<i}(1-\pi_j), \qquad \rho_i = \begin{cases} S_i\pi_i, & i < T, \\ S_T, & i = T, \end{cases} \tag{1}$$

with $S_1 = 1$. Thus trajectories that continue past every stopping boundary complete at T, and $\sum_{i=1}^{T} \rho_i = 1$. For token price $\lambda \geq 0$, the *utility* of stopping at i is $u_i = y_i - \lambda C_i$. The stopping problem maximizes expected utility over trajectories and stopping decisions:

$$\max_{\xi} \mathcal{U}_\lambda(\xi), \qquad \mathcal{U}_\lambda(\xi) = \mathbb{E}_{\tau\sim\mathcal{D}}\left[\sum_{i=1}^{T} \rho_i(\xi) u_i\right], \tag{2}$$

where $\mathcal{D}$ is the trajectory distribution induced by the questions, frozen reasoner and decoding rule. The token price λ values correctness against computation, without imposing a hard token budget. For a fixed reasoner and decoding rule, changing λ changes these values but not the outcome prediction targets at a given prefix.

Direct utility training. Full rollouts and forced answers supply cached correctness labels and token costs for offline optimization of Eq. (2). Jointly training the representation and stopping head on this utility ties representation learning to the token price λ and the current policy (Ehab et al., 2026). A new price can require retraining the controller. Boundaries that the policy rarely reaches, or where its decisions saturate, provide weak learning signals even when outcome labels are available (§ E.2). BELT learns outcome predictions independently of the price and policy, then fits stopping decisions on the frozen features by optimizing utility.

Outcome prediction with probes. A *probe* predicts output properties from a frozen LLM's internal states. A *belief* is an outcome probability conditioned on the observed prefix H_i. Following UHeads (Shelmanov et al., 2025; Ni et al., 2026a), BELT uses a causal encoder $h_i = E_\phi(H_i)$ and sigmoid readouts $\mathbf{p}_i = \sigma(W_{\text{bel}} h_i + b_{\text{bel}})$. Architecture and adaptations are detailed in §§ A.4 and A.5.

3 BELT: Belief Training and Policy Fitting

BELT first trains a price-independent probe on outcome labels, then freezes it and fits a linear stopping policy to the utility in Eq. (2) (Fig. 1). At a new token price, we refit only the policy.

Outcome beliefs. Four outcomes describe the current answer and the prospects of continued reasoning, with binary labels at $i < T$:

$$\begin{aligned} Y_i^{\text{now}} &= y_i, & Y_i^{\text{end}} &= y_T, \\ Y_i^{\text{later}} &= \mathbf{1}\{\exists j \in \{i+1,\dots,T\} : y_j = 1\}, & Y_i^{\text{same}} &= \mathbf{1}\{a_i = a_T\}. \end{aligned} \tag{3}$$

Here $\mathbf{1}\{\cdot\}$ is the indicator function. *Now* and *End* measure current and natural-completion correctness. *Later* includes any subsequent correct answer, even if the final answer is wrong. *Same* measures parsed-answer agreement with natural completion, regardless of correctness or agreement at intervening boundaries. Grading and comparing forced answers from completed rollouts supplies these labels (Fig. 1(a)).

For each event $k \in \mathcal{K} = \{\text{now}, \text{end}, \text{later}, \text{same}\}$, the probe estimates its probability from the observed prefix: $p_i^k \approx \Pr(Y_i^k = 1 \mid H_i)$.

The four targets describe the correctness component of utility in Eq. (2) for both actions. Stopping at i has expected utility $p_i^{\text{now}} - \lambda\,\mathbb{E}[C_i \mid H_i]$. Continuing leaves natural completion as an option, with expected utility $p_i^{\text{end}} - \lambda\,\mathbb{E}[C_T \mid H_i]$. The belief p_i^{later} indicates whether a later boundary could offer a correct answer; p_i^{same} predicts agreement with the natural-completion answer. Together, they distinguish a possible future correction (low *Now*, high *Later* or *End*), likely final-answer agreement (high *Same*), and a correct answer that may be lost (high *Now*, low *Same*). The value of continuing depends on future stopping decisions and the token price. Predicting the four trajectory outcomes keeps the probe reusable across prices (§ 6.4). Stage 2 learns the price-dependent trade-off from recorded correctness and token costs, with observed prefix length supplied to the Qwen3 BELT-B

policy (§ 3). § 6.2 tests the contribution of *Later* and *Same*. These labels and predictions define the belief supervision for Stage 1.

Stage 1: Belief training. The BELT probe first projects each token's hidden state to width 768 with an MLP. Learned marker embeddings distinguish block-end tokens, and two causal Transformer layers encode the observed prefix (Fig. 1(b)). We read the current block-end representation $h_i \in \mathbb{R}^{768}$ and apply four linear heads followed by sigmoids to obtain $\mathbf{p}_i = (p_i^{\text{now}}, p_i^{\text{end}}, p_i^{\text{later}}, p_i^{\text{same}})^\top$. With the reasoner frozen, we jointly train the shared encoder and readouts $\Theta = (\phi, W_{\text{bel}}, b_{\text{bel}})$ on labelled prefixes using equally weighted binary cross-entropy:

$$\Theta^\star = \arg\min_{\Theta} \mathcal{L}_{\text{bel}}, \qquad \mathcal{L}_{\text{bel}} = \frac{1}{4} \sum_{k \in \mathcal{K}} \mathbb{E}_{\tau \sim \mathcal{D}_{\text{bel}}} \left[\operatorname*{mean}_{i \in \mathcal{I}_k(\tau)} \text{BCE}(p_i^k, Y_i^k) \right]. \tag{4}$$

Here $\mathcal{D}_{\text{bel}}$ is the empirical training-trajectory distribution and $\mathcal{I}_k(\tau)$ contains the labelled positions for event k, unweighted by policy reach probabilities. The loss and validation checkpoint selection are independent of λ; the selected probe $\Theta^\star$ is frozen for Stage 2.

Stage 2: Policy fitting. We apply the frozen probe to a trajectory bank with recorded correctness y_i and cost C_i. BELT-B uses the four beliefs and observed prefix length (Fig. 1(c)); BELT-H uses their shared representation; we refer to these two policy inputs as the BELT-B and BELT-H interfaces. The OLMo3 variant of BELT-B omits prefix length (§ A.9). Both use a linear logistic stopping policy:

$$x_i^{\text{B}} = \text{Std}([\text{logit}(\mathbf{p}_i); d_i]), \qquad x_i^{\text{H}} = \text{Std}(h_i), \qquad \pi_i = \sigma(w^\top x_i + b). \tag{5}$$

Here $[\,;\,]$ concatenates features, d_i is the observed prefix-token count divided by 12,000, and Std uses means and standard deviations from the fitting bank. The elementwise transform $\text{logit}(p) = \log[p/(1-p)]$ uses clipped probabilities. At inference, we sample a stop decision from $\text{Bernoulli}(\pi_i)$ at each boundary; otherwise, reasoning continues until the next boundary or natural completion.

For $\xi = (w, b)$ and the empirical bank distribution $\mathcal{D}_{\text{bank}}$ in Eq. (2), we fit

$$(w_\lambda, b_\lambda) = \arg\max_{w,b} J_\lambda(w, b), \qquad J_\lambda(w, b) = \mathcal{U}_\lambda(w, b) - \kappa \|w\|_2^2. \tag{6}$$

The empirical utility averages $\sum_i \rho_i(y_i - \lambda C_i)$ over banked trajectories, including natural completion, with ρ_i from Eq. (1). It marginalizes stopping decisions exactly, conditional on the recorded trajectories and their fixed features, labels and costs. Development utility selects $\kappa \geq 0$ and the optimization iterate (§ A.4).

4 Experimental Setup

We evaluate stopping performance across models and tasks (§ 5). We then compare representations under common policy fitting and examine the effects of supervision and price adaptation (§ 6).

Training data. We train the BELT probe on 3,918 questions from DAPO-Math-17k (Yu et al., 2025), NuminaMath-1.5 (LI et al., 2024) and DeepScaleR-Preview (Tan et al., 2026). We sample 8 trajectories per question for training and filter those with format errors. A separate 320-question validation bank selects the BELT probe checkpoint. We use 160 of these questions (1,278 trajectories) to fit BELT's linear stopping policies and the other 160 to select policy hyperparameters (Eqs. (5) and (6)). §§ A.1 and A.3 details dataset composition and filtering. We force and grade an answer every four reasoning blocks for belief supervision (§ 3; Fig. 1(a)).

Evaluation data. ID sets contain held-out DAPO-Math, NuminaMath and DeepScaleR questions (DAPO, Numina and DSR). OOD sets are MATH-500 (M500) (Lightman et al., 2023), OlympiadBench (Olymp.) (He et al., 2024), Gaokao2023-Math-En (GK) (Liao et al., 2024), AIME 2022–2025 (MAA, 2025), and AMC 2022–2023 (MAA, 2023). We test BELT and OS-Pruner variants (Ehab et al., 2026) on LiveCodeBench (Jain et al., 2025) (Pre: before August 2024; Post: August 2024 onward), transferring to coding without code-side training (§ A.10).

Metrics. Following Wu et al. (2025); Ehab et al. (2026), we compare test-frontier accuracy at matched token savings of 70%, 60%, 50%, and 40% relative to natural completion, and maximum token saving at matched accuracy tolerances of 0, 1, and 2 percentage points. For each method

Table 1: Token saving (%) at fixed accuracy tolerances on Qwen3 (macro averages). Cells report means with SD across independent training runs, or question-bootstrap SE for training-free methods (‡). Bold/underline denote best/second-best values; Probe Params counts trainable probe/controller parameters. Per-dataset results appear in § B.

Method	Probe Params	Acc. tolerance: 0 pp ID	Acc. tolerance: 0 pp OOD	Acc. tolerance: 1 pp ID	Acc. tolerance: 1 pp OOD	Acc. tolerance: 2 pp ID	Acc. tolerance: 2 pp OOD
CoDE-Stop‡	0	$27.98_{\pm 0.52}$	$41.15_{\pm 0.71}$	$36.15_{\pm 0.54}$	$45.55_{\pm 0.73}$	$41.59_{\pm 0.62}$	$47.61_{\pm 0.76}$
DEER‡	0	$29.40_{\pm 0.76}$	$30.72_{\pm 1.13}$	$32.21_{\pm 0.76}$	$32.47_{\pm 1.18}$	$34.42_{\pm 0.77}$	$33.64_{\pm 1.19}$
TC	514	$37.82_{\pm 0.68}$	$42.53_{\pm 1.09}$	$41.16_{\pm 0.66}$	$45.60_{\pm 0.32}$	$44.67_{\pm 0.43}$	$47.90_{\pm 1.76}$
ROM	5M	$31.23_{\pm 0.79}$	$32.44_{\pm 1.21}$	$34.50_{\pm 0.87}$	$35.30_{\pm 1.65}$	$37.10_{\pm 1.45}$	$37.95_{\pm 1.87}$
OSP	101M ×5	$\underline{60.00}_{\pm 1.82}$	$\underline{63.80}_{\pm 0.56}$	$\underline{62.34}_{\pm 1.06}$	$66.64_{\pm 0.76}$	$\underline{64.69}_{\pm 0.31}$	$68.27_{\pm 0.71}$
OSP-1	101M	$57.70_{\pm 3.93}$	$63.33_{\pm 1.09}$	$60.73_{\pm 3.26}$	$\underline{66.68}_{\pm 0.85}$	$63.76_{\pm 2.59}$	$\underline{69.01}_{\pm 0.83}$
BELT-B-1	16M	$53.58_{\pm 3.57}$	$49.70_{\pm 4.04}$	$56.12_{\pm 3.75}$	$51.59_{\pm 4.15}$	$58.29_{\pm 3.52}$	$53.43_{\pm 4.31}$
BELT-H-1	16M	$58.87_{\pm 0.59}$	$57.50_{\pm 2.52}$	$61.15_{\pm 0.70}$	$59.39_{\pm 2.85}$	$63.05_{\pm 0.67}$	$61.21_{\pm 3.14}$
BELT-B	16M	$59.05_{\pm 1.00}$	$63.07_{\pm 0.66}$	$61.76_{\pm 1.31}$	$65.60_{\pm 0.37}$	$64.22_{\pm 1.27}$	$67.58_{\pm 1.35}$
BELT-H	16M	$\mathbf{60.12}_{\pm 0.30}$	$\mathbf{66.18}_{\pm 0.95}$	$\mathbf{63.71}_{\pm 0.47}$	$\mathbf{68.82}_{\pm 0.42}$	$\mathbf{66.33}_{\pm 0.61}$	$\mathbf{71.26}_{\pm 0.11}$

and training run, micro-average test accuracy and cost define the frontier; interpolation between adjacent nondominated policies matches these constraints. We complement this comparison with split-half stability checks (§ E.1) and test utility, $\mathrm{Accuracy} - \lambda\,\mathrm{TokenCost}$, of policies selected by development utility at each λ (dev-selected utility). Full curves and protocols appear in Fig. 2 and §§ A.6 and A.7.

LLMs and baselines. We evaluate Qwen3-1.7B (Qwen3, reasoning mode) and OLMo-3-7B-Think (OLMo3). The BELT probe reads layer 16 of Qwen3 and layer 18 of OLMo3 (both between one half and two thirds of the depth). Training-free baselines are fixed token caps, CoDE-Stop (Hosseini et al., 2026) and DEER (Yang et al., 2026). Trained baselines are Thought Calibration (TC) (Wu et al., 2025), ROM (Wang et al., 2026a) and OS-Pruner (OSP) (Ehab et al., 2026). Full OSP uses five separately trained checkpoints, one per token price λ, to cover accuracy–cost trade-offs. For comparison with a single BELT probe, OSP-1 trains one checkpoint at an intermediate λ (10^{-5}) and adjusts the trade-off through an inference-time threshold. BELT-B-1/H-1 likewise fit one policy at $\lambda = 10^{-5}$ on Qwen3 (2×10^{-5} on OLMo3) and sweep 13 inference-time thresholds on its stopping probability instead of refitting the policy across λ, matching OSP-1's family size.

5 Main Results

BELT-H saves the most tokens on Qwen3. On Qwen3, BELT-H saves the most tokens at every reported accuracy tolerance on ID and OOD, improving token saving over OSP by 2.2 pp at the 1% OOD tolerance (Table 1). The choice of interface depends on the objective: BELT-H achieves 1–4 pp greater token saving at a fixed accuracy tolerance, while BELT-B has higher dev-selected utility and provides directly interpretable weights with just 6 policy parameters (§ 6.3). At shared token budgets, BELT achieves competitive or better accuracy than the OSP variants (Table 2). Fig. 2(a) shows the comparison as accuracy–cost curves. Under dev-selected utility, BELT-B leads full OSP in six of the ten Qwen3 math cells (five prices, ID and OOD) and both interfaces lead OSP-1 in all ten. A single BELT probe supports all of these operating points, whereas the OSP family uses five price-specific checkpoints. Threshold expansion, which sweeps inference-time thresholds on top of every fitted OSP and BELT policy, keeps this ordering: expanded BELT-H still saves 2.1 pp more than expanded OSP at the 1% OOD tolerance. See the appendix for dev-selected utility (§ B.5 and Table 37), threshold-expanded comparisons (§ B.6 and Table 38), and split-half stability checks (§ E.1 and Table 60).

BELT also outperforms OSP-1 on OLMo3. We replicate belief training with a new probe on OLMo3. Relative to OSP-1, BELT-H improves token saving by 8.1 pp at the 1% tolerance and accuracy by 2.0 pp at 50% token saving. BELT-B exceeds OSP-1 by 8.2 and 2.8 pp, respectively

Table 2: Accuracy (%) at fixed token-saving budgets on Qwen3 (macro averages). Reporting conventions follow Table 1. Natural† denotes natural completion; NF denotes a budget infeasible in at least one seed. Detailed per-dataset results are in § B.

Method	Probe Params	Saving: 70% ID	Saving: 70% OOD	Saving: 60% ID	Saving: 60% OOD	Saving: 50% ID	Saving: 50% OOD	Saving: 40% ID	Saving: 40% OOD
Natural†‡	–	$48.85_{\pm 1.0}$	$60.58_{\pm 1.4}$	$48.85_{\pm 1.0}$	$60.58_{\pm 1.4}$	$48.85_{\pm 1.0}$	$60.58_{\pm 1.4}$	$48.85_{\pm 1.0}$	$60.58_{\pm 1.4}$
Fixed cap‡	0	$29.15_{\pm 0.9}$	$38.61_{\pm 1.3}$	$34.25_{\pm 0.9}$	$44.60_{\pm 1.4}$	$39.18_{\pm 1.0}$	$49.40_{\pm 1.4}$	$42.86_{\pm 1.0}$	$52.86_{\pm 1.4}$
CoDE-Stop‡	0	$32.94_{\pm 1.0}$	$45.53_{\pm 1.4}$	$39.10_{\pm 1.0}$	$52.57_{\pm 1.4}$	$44.17_{\pm 1.0}$	$56.24_{\pm 1.4}$	$47.43_{\pm 1.0}$	$58.82_{\pm 1.3}$
DEER‡	0	$22.14_{\pm 0.8}$	$24.38_{\pm 1.0}$	$28.54_{\pm 0.8}$	$32.09_{\pm 1.1}$	$35.48_{\pm 0.9}$	$41.43_{\pm 1.2}$	$42.92_{\pm 1.0}$	$52.10_{\pm 1.3}$
Thought Calib.	514	$30.90_{\pm 0.8}$	$41.49_{\pm 1.4}$	$38.97_{\pm 0.2}$	$50.48_{\pm 0.9}$	$44.55_{\pm 0.0}$	$56.54_{\pm 0.3}$	$48.35_{\pm 0.2}$	$60.27_{\pm 0.2}$
ROM	5M	$28.75_{\pm 1.6}$	$38.59_{\pm 2.7}$	$34.94_{\pm 1.4}$	$45.54_{\pm 2.1}$	$40.35_{\pm 1.1}$	$51.49_{\pm 1.5}$	$45.31_{\pm 0.8}$	$56.50_{\pm 1.1}$
OS-Pruner	101M ×5	$\underline{44.58}_{\pm 0.5}$	NF	$\underline{48.89}_{\pm 0.5}$	$61.26_{\pm 0.5}$	$51.55_{\pm 0.1}$	$63.59_{\pm 0.5}$	$\underline{53.56}_{\pm 0.2}$	$65.42_{\pm 0.1}$
OS-Pruner-1	101M	$44.44_{\pm 0.6}$	$\underline{57.81}_{\pm 0.4}$	$48.17_{\pm 0.9}$	$60.88_{\pm 0.6}$	$50.60_{\pm 1.0}$	$62.99_{\pm 0.8}$	$51.98_{\pm 0.8}$	$63.93_{\pm 0.7}$
BELT-B-1	16M	$38.22_{\pm 3.7}$	$46.12_{\pm 5.7}$	$45.44_{\pm 2.0}$	$54.69_{\pm 3.5}$	$50.01_{\pm 1.0}$	$60.41_{\pm 1.9}$	$52.82_{\pm 0.4}$	$64.07_{\pm 0.8}$
BELT-H-1	16M	$42.66_{\pm 1.6}$	$51.84_{\pm 3.4}$	$48.45_{\pm 0.4}$	$59.53_{\pm 1.2}$	$\mathbf{51.84}_{\pm 0.1}$	$63.55_{\pm 0.2}$	$\mathbf{53.69}_{\pm 0.3}$	$\mathbf{65.76}_{\pm 0.3}$
BELT-B	16M	$44.06_{\pm 0.6}$	$56.97_{\pm 1.2}$	$48.58_{\pm 0.5}$	$\underline{61.33}_{\pm 0.7}$	$51.31_{\pm 0.4}$	$\underline{63.68}_{\pm 0.3}$	$53.01_{\pm 0.2}$	$65.13_{\pm 0.4}$
BELT-H	16M	$\mathbf{45.64}_{\pm 0.3}$	$\mathbf{58.49}_{\pm 0.2}$	$\mathbf{49.23}_{\pm 0.1}$	$\mathbf{61.90}_{\pm 0.2}$	$\underline{51.57}_{\pm 0.1}$	$\mathbf{63.92}_{\pm 0.3}$	$53.37_{\pm 0.1}$	$\underline{65.48}_{\pm 0.2}$

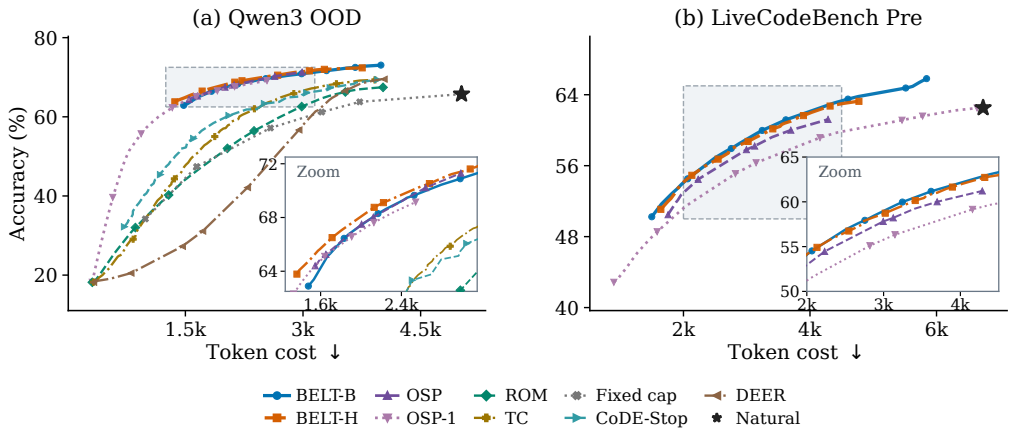

Figure 2: Accuracy–token cost frontiers on (a) Qwen3 OOD and (b) LiveCodeBench Pre. Stars mark natural completion. Additional curves: per-dataset (Fig. 6), Qwen3 ID and LiveCodeBench Post (Figs. 4 and 5).

(Table 3). BELT-H leads at the tightest budget, while BELT-B leads at the other three. The saving advantage over OSP-1 persists with BELT-B-1/H-1, and BELT-B/H exceed OSP-1 in the utility of dev-selected policies at every reported λ. See the appendix for utility results (§ B.5 and Table 37), full frontiers (§ B.10), and the comparator rationale (§ E.5).

Math-trained BELT probes transfer to coding. We reuse the math-trained probe and policy weights without code-side training. On LiveCodeBench Pre, BELT-H saves 14.5 pp and BELT-B 15.6 pp more tokens than OSP at the 1% tolerance (Fig. 2(b), Table 3). Both interfaces also exceed OSP in accuracy at 50% token saving, and BELT-B-1/H-1 retain a saving advantage over OSP-1. BELT remains competitive on LiveCodeBench Post and in the utility of dev-selected policies. See the appendix for Post results (§ B.3 and Tables 28 and 29), transfer checks (§ E.3), and dev-selected utility comparisons (§ B.5 and Table 37).

Price-specific policy fitting improves token saving. BELT-B-1/H-1 use inference-time thresholds on a single fitted policy, matching OSP-1. Refitting the linear policy for each λ raises BELT-H's OOD token saving by 9.4 pp at the 1% tolerance relative to BELT-H-1, with similar accuracy at 50% token saving. Full OSP also improves accuracy over OSP-1 at every shared token budget where both are feasible (Table 2). BELT obtains the benefit of price-specific fitting by reusing the frozen probe

Table 3: **Top:** OLMo3 performance on five OOD datasets (macro averages). Token saving is evaluated at accuracy tolerances and accuracy at token-saving budgets. Conventions follow Tables 1 and 2; trained methods use three independent runs. **Bottom:** math-to-code transfer on LiveCodeBench Pre with Qwen3 under the same conventions; Natural† is the deterministic natural-completion reference.

Method	Probe Params	Token saving (%) at tolerance 0.00	0.01	0.02	Accuracy (%) at saving 70%	60%	50%	40%
Natural†‡	–	–	–	–	78.35	78.35	78.35	78.35
Fixed cap‡	0	–	–	–	$53.60_{\pm 1.5}$	$59.98_{\pm 1.5}$	$65.55_{\pm 1.5}$	$70.39_{\pm 1.4}$
CoDE-Stop‡	0	$0.00_{\pm 0.0}$	$8.64_{\pm 0.3}$	$17.27_{\pm 0.6}$	$48.94_{\pm 1.4}$	$55.45_{\pm 1.4}$	$61.06_{\pm 1.5}$	$65.95_{\pm 1.4}$
DEER‡	0	$0.00_{\pm 0.0}$	$1.57_{\pm 0.3}$	$10.80_{\pm 0.7}$	$35.63_{\pm 1.2}$	$44.17_{\pm 1.1}$	$52.71_{\pm 1.2}$	$60.32_{\pm 1.3}$
Thought Calib.	514	$25.23_{\pm 0.3}$	$29.56_{\pm 0.5}$	$33.85_{\pm 0.1}$	$54.02_{\pm 0.2}$	$63.04_{\pm 0.2}$	$68.43_{\pm 0.1}$	$74.27_{\pm 0.1}$
ROM	8M	$29.98_{\pm 3.9}$	$36.06_{\pm 1.2}$	$38.24_{\pm 1.0}$	$32.17_{\pm 1.2}$	$43.08_{\pm 0.4}$	$59.79_{\pm 0.5}$	$74.76_{\pm 0.6}$
OS-Pruner-1	405M	$41.53_{\pm 1.6}$	$49.14_{\pm 1.7}$	$53.55_{\pm 1.6}$	$69.74_{\pm 0.6}$	$75.31_{\pm 0.8}$	$78.00_{\pm 0.4}$	$79.29_{\pm 0.1}$
BELT-B-1	19M	$52.21_{\pm 1.7}$	$54.61_{\pm 2.0}$	$56.69_{\pm 2.6}$	$66.73_{\pm 2.6}$	$75.90_{\pm 1.4}$	$80.31_{\pm 0.6}$	$82.07_{\pm 0.2}$
BELT-H-1	19M	$52.44_{\pm 1.1}$	$54.46_{\pm 0.7}$	$56.14_{\pm 0.8}$	$62.28_{\pm 1.0}$	$74.84_{\pm 1.1}$	$79.91_{\pm 0.6}$	$81.66_{\pm 0.4}$
BELT-B	19M	$\mathbf{54.28}_{\pm 1.6}$	$\mathbf{57.32}_{\pm 1.4}$	$59.67_{\pm 2.0}$	$70.74_{\pm 1.7}$	$\mathbf{77.52}_{\pm 0.8}$	$\mathbf{80.78}_{\pm 0.4}$	$\mathbf{82.22}_{\pm 0.3}$
BELT-H	19M	$53.34_{\pm 0.6}$	$57.22_{\pm 0.8}$	$\mathbf{59.91}_{\pm 1.8}$	$\mathbf{71.24}_{\pm 1.3}$	$76.85_{\pm 0.4}$	$80.04_{\pm 0.4}$	$81.91_{\pm 0.2}$

Method	Probe Params	Token saving (%) at tolerance 0.00	0.01	0.02	Accuracy (%) at saving 70%	60%	50%	40%
Natural†	–	–	–	–	62.52	62.52	62.52	62.52
OS-Pruner	101M ×5	$12.46_{\pm 21.6}$	$27.75_{\pm 14.5}$	$38.89_{\pm 13.0}$	$53.09_{\pm 1.5}$	$56.49_{\pm 0.1}$	$59.10_{\pm 1.0}$	$60.72_{\pm 1.3}$
OS-Pruner-1	101M	$1.25_{\pm 2.2}$	$15.76_{\pm 6.5}$	$27.37_{\pm 9.1}$	$51.33_{\pm 0.8}$	$54.51_{\pm 1.4}$	$56.94_{\pm 1.4}$	$58.83_{\pm 1.3}$
BELT-B-1	16M	$35.06_{\pm 7.1}$	$39.66_{\pm 6.7}$	$44.14_{\pm 6.3}$	$52.35_{\pm 0.5}$	$56.03_{\pm 0.9}$	$59.14_{\pm 1.4}$	$61.52_{\pm 1.4}$
BELT-H-1	16M	$32.73_{\pm 1.9}$	$39.08_{\pm 1.6}$	$44.10_{\pm 2.1}$	$52.81_{\pm 0.1}$	$56.46_{\pm 0.3}$	$59.20_{\pm 0.3}$	$61.32_{\pm 0.3}$
BELT-B	16M	$\mathbf{37.20}_{\pm 5.6}$	$\mathbf{43.33}_{\pm 4.3}$	$\mathbf{48.60}_{\pm 2.4}$	$\mathbf{54.19}_{\pm 0.5}$	$\mathbf{57.67}_{\pm 0.3}$	$\mathbf{60.31}_{\pm 0.5}$	$\mathbf{62.16}_{\pm 0.7}$
BELT-H	16M	$24.61_{\pm 21.4}$	$42.26_{\pm 1.2}$	$47.29_{\pm 1.0}$	$54.17_{\pm 0.8}$	$57.24_{\pm 0.6}$	$59.86_{\pm 0.3}$	$61.94_{\pm 0.3}$

and adapting only the linear policy. § 6.3 examines how the policy changes with λ, while § 6.4 quantifies its adaptation cost; § B.6 gives the threshold-only comparisons.

6 Analysis

We compare belief-trained representations with alternative features, separate the effects of supervision and policy inputs, and examine reuse across token prices. Utility ($Acc - \lambda Cost$) compares policies without matching accuracy or cost.

6.1 Belief-trained representations for stopping

To compare representations under common policy fitting, we freeze each feature set and fit linear stopping policies. Table 4 shows that on math OOD, features from the belief-trained representation yield higher mean utility than raw states, PCA, and mean-pooled states at all five reported token prices λ. Relative to utility-trained OS-Pruner-1 features, they improve mean utility by 0.014–0.019 under the same policy-fitting and development-selection procedure. Supervising the same four targets on a linear probe over the layer-16 state raises utility by 0.032–0.036 over raw states at every price. The BELT probe adds a further 0.013–0.034, suggesting that both the targets and the sequence probe contribute. The same ordering holds on both math ID and OOD when both methods fit linear stopping policies from four different initializations. Across these feature configurations, belief training yields features that support stopping at different token prices. Training the same probe architecture directly on the stopping utility reaches comparable operating points in one of three seeds and

Table 4: Math OOD average utility. Means: five fitting starts (rows 1–4), three training runs (OSP-1, linear probe, BELT). SDs in § C.1 and Table 46.

Features / λ ($\times 10^{-6}$)	**2**	**5**	**10**	**20**	**50**
Before belief training	0.655	0.643	0.623	0.584	0.491
Raw state (layer 16)	0.665	0.656	0.636	0.597	0.510
PCA (768-d)	0.664	0.652	0.632	0.592	0.502
Mean-pooled states	0.663	0.651	0.632	0.592	0.480
OSP-1 features	0.698	0.686	0.668	0.635	0.565
Linear probe 4 beliefs	0.701	0.689	0.668	0.629	0.545
BELT probe (BELT-H)	**0.714**	**0.702**	**0.683**	**0.654**	**0.579**

Table 5: BELT-probe and OSP-checkpoint parameter counts, inference FLOPs per token, and estimated controller-training GPU-hours on Qwen3.

Quantity	**BELT**	**OSP**
Parameters (M)	15.96	100.68
FLOPs/token (M)	85.5	347.6
One probe (GPU-h)	∼10.0	∼50.9
Policies for 5 λ (GPU-h)	∼10.0	∼254.5

Table 6: Average math OOD utility for (a) prediction-target and (b) policy-input ablations (fixed four-belief probe). Columns give token prices λ in 10^{-6}. Means over three BELT probe training runs. See SDs in the appendix (§ C.1 and Table 47).

Belief set	**(a) Prediction targets**					**(b) Policy inputs**				
	2	**5**	**10**	**20**	**50**	**2**	**5**	**10**	**20**	**50**
Now, *End*	0.7158	0.7034	0.6826	0.6382	0.5499	0.7222	0.7037	0.6746	0.6371	0.5553
Now, *End*, *Later*	0.7162	0.7068	0.6854	0.6458	0.5720	0.7197	0.7067	0.6823	0.6430	0.5692
Now, *End*, *Same*	0.7200	0.7084	0.6890	**0.6525**	0.5713	0.7226	0.7096	0.6900	0.6501	0.5725
All four	**0.7232**	**0.7104**	**0.6904**	0.6524	**0.5738**	**0.7232**	**0.7104**	**0.6904**	**0.6524**	**0.5738**

collapses to never-stop in the other two. See utility controls (§ C.1 and Table 46), matched feature comparisons (§ C.2 and Tables 48 and 49), and direct-training results (§ E.2 and Table 61).

6.2 Contributions of the *Later* and *Same* beliefs

Do Later and Same help stopping control? We compare removing their prediction targets during belief training with removing their inputs during policy fitting (Table 6).

Removing prediction targets. We retrain the BELT probe on the retained targets, changing both the learned representation and the belief outputs supplied to the policy. At $\lambda = 10^{-5}$, the two targets together raise mean utility from 0.6826 (*Now*/*End* only) to 0.6904. Keeping either target recovers part of the gain, while all four yield the highest mean utility. The four-target recipe also outperforms *Now*/*End* at every other reported token price (Table 6a).

Removing policy inputs. We keep belief training unchanged, freeze the four-belief BELT probe, and refit the policy using only the retained inputs (Table 6b). At the same token price, removing *Same* reduces mean utility by 0.0081, whereas removing *Later* reduces it by 0.0004. Adding *Later* to *Now*/*End* during training slightly improves mean utility at all five prices.

6.3 Interpreting the BELT-B logistic policy

BELT-B fits a logistic policy with six parameters, so its weights can be read directly. **(1) Beliefs have distinct conditional roles.** Holding other inputs fixed, *Now* and *Same* favor stopping (positive mean weights), while *End* and *Later* favor continuation (negative mean weights). **(2) Higher token prices encourage stopping.** As λ rises, stopping probabilities increase overall and the stopping region extends to lower *Now* beliefs. **(3) Relative belief weights change with price.** At low prices, the policy tends to stop when the current answer is likely correct but natural completion is not. At higher prices, relative weight shifts from *Now* toward *Same*, while *Later* becomes more negative. This favors stopping on

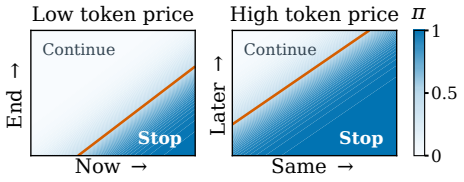

Figure 3: Illustrative policies on different belief pairs, not fitted estimates. Blue: stop probability; lines: $\pi = 0.5$.

expected final-answer agreement while continuing when future correction remains likely (Fig. 3). § C.4 and Fig. 8 give the fitted weights and stopping regions.

6.4 Training cost and reuse across token prices

New token prices reuse the BELT probe. For K token prices, we train the BELT probe once, share its exported features, and perform K CPU policy fits. Full OSP trains K checkpoints. Each BELT-B/H policy adds only 6/769 parameters (§ D.3). Threshold expansion can improve both methods while preserving BELT's training-cost advantage (§ B.6).

A compact probe reduces per-checkpoint cost. On Qwen3, the BELT probe has $6.3\times$ fewer parameters and requires approximately $4.1\times$ fewer inference FLOPs per token than one OS-Pruner checkpoint (Table 5 and § D.1). Under the recorded schedules, its estimated controller-training cost is approximately $5.1\times$ lower than one OS-Pruner checkpoint and $25.5\times$ lower than OS-Pruner's five-checkpoint configuration. These estimates exclude shared trajectory generation, feature export, and CPU policy fitting (§§ D and D.2).

6.5 BELT probe and policy design

A linear stopping policy matches an MLP on the same four belief logits and prefix length to within 0.001 mean utility. A top-two-layer probe branch yields configuration-dependent differences. Compressed concatenation of four reasoner layers raises mean utility by 0.0026–0.0082 across prices in the full-bank study, whereas scalar mixture and dynamic attention improve on a single layer only at some prices. These comparisons support using a linear policy with the single-layer probe, while multi-layer concatenation offers further utility gains. See policy capacity (§ C.1 and Table 46), probe capacity (§ C.3 and Table 54), and multi-layer comparisons (§ C.7 and Table 58).

7 Related Work

Efficient reasoning and early stopping. Efficient reasoning methods train on concise or shortest-correct traces (Chen et al., 2025; Yang et al., 2025) or control inference-time budgets (Muennighoff et al., 2025; Li et al., 2026a). Early-stopping methods use answer confidence (Yang et al., 2026; Hosseini et al., 2026), convergence (Li et al., 2026b), or hidden-state overthinking detectors (Wang et al., 2026a). Thought Calibration (Wu et al., 2025) learns correctness or final-answer agreement scores and separately calibrates thresholds with statistical risk control. OS-Pruner (Ehab et al., 2026) jointly trains a representation and stopping head on sequential utility at a chosen token price. BELT reuses a belief-trained representation across prices, refitting both feature weights and the decision threshold.

Predictive representations for control. Hidden-state probes support hallucination detection (Shelmanov et al., 2025), reasoning-step verification and search (ReProbe; Ni et al., 2026a), and intermediate-answer correctness prediction (Zhang et al., 2025). ZIP-RC predicts reward and remaining cost to allocate computation among samples (Manvi et al., 2026). AERA predicts recovery from cumulative response sets (Wang et al., 2026b). BELT builds on ReProbe to control stopping within a single trajectory. §§ 6.1 and 6.2 isolate the utility gains from belief supervision and policy inputs.

8 Conclusion

BELT learns a representation reusable across token prices by separating belief training from policy fitting. Under common policy fitting, belief-trained representations yield higher mean utility on mathematics than raw and utility-trained features. Ablations distinguish the contribution of supervision from the use of predictions as policy inputs. On Qwen3, a linear stopping policy on the frozen probe saves 68.82% of OOD tokens at a 1% accuracy tolerance, 2.2 pp more than OSP. Across the evaluated models and tasks, BELT achieves accuracy–token cost trade-offs competitive with or better than those of directly utility-trained controllers. Adapting to a new price requires refitting only the policy. Covering five prices this way has an estimated $25.5\times$ lower controller-training cost than training five OSP checkpoints.

Reproducibility statement

§ A specifies supervision, policy fitting, data partitions and metric conventions. Evaluation exports record question rosters, input hashes, frontier members and mixture weights. All reported evaluations reuse banked reasoning trajectories. We will release the code, trained BELT probe checkpoints, fitted stopping policies and evaluation records to reproduce the tables and figures.

Ethics statement

Our experiments use public reasoning and coding benchmarks. Reduced reasoning cost does not guarantee answer correctness, and matching an accuracy constraint on an evaluation set does not guarantee it on a new distribution. The method is not evaluated for safety-critical decisions.

AI use statement

We used generative AI tools (LLM coding assistants) for implementing and debugging parts of the experiment and evaluation code, for data-analysis scripts and table generation, and for editing the manuscript for clarity. We did not use generative AI tools to generate synthetic data, to clean or reformat datasets, to design the method or the experiments, to propose or refine hypotheses, to derive the formulation in §§ 2 and 3 or any mathematical claim, or to interpret the results beyond producing summaries that the authors checked against the exported tables; translation, proofs and qualitative data analysis are not applicable to this work. All AI-assisted code and text were reviewed by the authors, who take full responsibility for the contents of this paper.

References

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do not think that much for 2+3=? on the overthinking of long reasoning models. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Mohammed Ehab, Aymane El Gadarri, Vivek F. Farias, Adam Jozefiak, and Ciamac C. Moallemi. Os-pruner: Pruning chains-of-thought of reasoning models via optimal stopping, 2026. URL https://arxiv.org/abs/2607.11089.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Parsa Hosseini, Sumit Nawathe, Mahdi Salmani, Meisam Razaviyayn, and Soheil Feizi. Early stopping for large reasoning models via confidence dynamics, 2026. URL https://arxiv.org/abs/2604.04930.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida I. Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/94074dd5a072d28ff75a76dabed43767-Paper-Conference.pdf.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. [https://huggingface.co/datasets/AI-MO/NuminaMath-1.5](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

Junyan Li, Wenshuo Zhao, Yang Zhang, and Chuang Gan. Steering LLM thinking with budget guidance. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37437–37451, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1866. URL https://aclanthology.org/2026.findings-acl.1866/.

Junyan Li, Wenshuo Zhao, Yang Zhang, and Chuang Gan. Steering LLM thinking with budget guidance. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37437–37451, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1866. URL https://aclanthology.org/2026.findings-acl.1866/.

Minpeng Liao, Chengxi Li, Wei Luo, Wu Jing, and Kai Fan. MARIO: MAth reasoning with code interpreter output - a reproducible pipeline. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 905–924, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.53. URL https://aclanthology.org/2024.findings-acl.53/.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

MAA. American mathematics competitions (AMC), 2022–2023. Official competition website, 2023. URL https://maa.org/student-programs/amc/. Accessed September 22, 2026.

MAA. American invitational mathematics examination (AIME), 2022–2025. Official competition website, 2025. URL https://maa.org/maa-invitational-competitions/. Accessed September 22, 2026.

Rohin Manvi, Joey Hong, Tim Seyde, Maxime Labonne, Mathias Lechner, and Sergey Levine. Zero-overhead introspection for adaptive test-time compute. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GqZYGOYuF2.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Jingwei Ni, Ekaterina Fadeeva, Tianyi Wu, Mubashara Akhtar, Jiaheng Zhang, Elliott Ash, Markus Leippold, Timothy Baldwin, See-Kiong Ng, Artem Shelmanov, and Mrinmaya Sachan. ReProbe: Efficient test-time scaling of multi-step reasoning by probing internal states of large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11667–11689, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.536. URL https://aclanthology.org/2026.acl-long.536/.

Jingwei Ni, Yu Fan, Vilém Zouhar, Donya Rooein, Alexander Miserlis Hoyle, Mrinmaya Sachan, Markus Leippold, Dirk Hovy, and Elliott Ash. Can reasoning help large language models capture human annotator disagreement? In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36–54, Rabat, Morocco, March 2026b. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.3. URL https://aclanthology.org/2026.eacl-long.3/.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Stuart Russell and Eric Wefald. Principles of metareasoning. *Artificial Intelligence*, 49(1):361–395, 1991. ISSN 0004-3702. doi: https://doi.org/10.1016/0004-3702(91)90015-C. URL https://www.sciencedirect.com/science/article/pii/000437029190015C.

C. Nicolò De Sabbata, Theodore R. Sumers, Badr AlKhamissi, Antoine Bosselut, and Thomas L. Griffiths. Rational metareasoning for large language models, 2025. URL https://arxiv.org/abs/2410.05563.

Artem Shelmanov, Ekaterina Fadeeva, Akim Tsvigun, Ivan Tsvigun, Zhuohan Xie, Igor Kiselev, Nico Daheim, Caiqi Zhang, Artem Vazhentsev, Mrinmaya Sachan, Preslav Nakov, and Timothy Baldwin. A head to predict and a head to question: Pre-trained uncertainty quantification heads for hallucination detection in LLM outputs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 35712–35731, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1809. URL https://aclanthology.org/2025.emnlp-main.1809/.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 10131–10165, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1b623663fd9b874366f3ce019fdfdd44-Paper-Conference.pdf.

Sijun Tan, Michael Luo, Justin Wong, Colin Cai, Xiaoxiang Shi, William Yuan Tang, Manan Roongta, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Effective RL scaling of reasoning models via iterative context lengthening, 2026. URL https://openreview.net/forum?id=I6GzDCne7U.

Team Olmo. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Junlin Wang, Siddhartha Jain, Dejiao Zhang, Baishakhi Ray, Varun Kumar, and Ben Athiwaratkun. Reasoning in token economies: Budget-aware evaluation of LLM reasoning strategies. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 19916–19939, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1112. URL https://aclanthology.org/2024.emnlp-main.1112/.

Xinyan Wang, Xiaogeng Liu, Ming Pei, and Chaowei Xiao. Rom: Real-time overthinking mitigation via streaming detection and intervention, 2026a. URL https://arxiv.org/abs/2603.22016.

Ziming Wang, Ivor Tsang, and Hangwei Qian. Aera: Adaptive evidence residual allocation for efficient test-time reasoning, 2026b. URL https://arxiv.org/abs/2608.27964.

Menghua Wu, Cai Zhou, Stephen Bates, and Tommi Jaakkola. Thought calibration: Efficient and confident test-time scaling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14291–14305, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.722. URL https://aclanthology.org/2025.emnlp-main.722/.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 88170–88210, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8df90a1440ce782d1f5607b7a38f2531-Paper-Conference.pdf.

Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. Towards thinking-optimal scaling of test-time compute for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=6ICFqmixlS.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244. Curran Associates, Inc., 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a4277440d50f1f15d2cb4c14f7e0c0d2-Paper-Conference.pdf.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=O6I0Av7683.

A Experimental Protocol and Reproducibility

A.1 Evaluation grading

Admission and grading. To support consistent grading and reliable belief labels, we use easy-to-grade, text-only integer-answer subsets, following DAPO-Math's integer-answer transformation (Yu et al., 2025, Section 3.5) and the integer-answer TheoremQA evaluation of Wang et al. (2024, Appendix D.3). This reduces symbolic-equivalence and answer-format ambiguity for intermediate and final answers. We filter existing questions without rewriting them; scores describe the admitted populations in Table 7.

The reference parser removes whitespace, dollar delimiters, outer `\boxed{}` wrappers, and trailing periods, then requires an optional minus sign followed by digits. Source adapters convert integral numeric fields to integer strings. Fractions, symbolic expressions, decimal strings, and comma-separated numerals are rejected. We also exclude empty questions, questions longer than 2,000 characters, and those containing `[asy]`, `\includegraphics`, or `<img`; OlympiadBench must have a single answer. These deterministic gates do not exhaustively detect images.

Within each comparison, methods share the fixed roster, reasoner trajectories, strict grader, total generated-token accounting, and operating-point constraints. Admission depends on reference format rather than method performance; missing method coverage does not change the roster in Table 7.

Table 7: ID/OOD test questions after filtering and deduplication. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively.

Dataset	Distribution	Questions	Composition
DAPO	ID	740	Held-out DAPO source questions
Numina	ID	473	Held-out NuminaMath-hard source questions
DSR	ID	481	Held-out DeepScaleR source questions
M500	OOD	295	Retained MATH-500 questions
Olymp.	OOD	366	English text-only mathematics, single answer
GK	OOD	191	Gaokao 2023 mathematics, English
AIME	OOD	98	2022–25, nine known duplicates removed
AMC	OOD	75	35 from 2022 + 40 from 2023

Sources and deduplication. The ID bank draws from DAPO-Math-17k, NuminaMath-1.5, and DeepScaleR-Preview. NuminaMath retains valid problems tagged `olympiads`, `amc_aime`, or `aops_forum`; DAPO-Math prompt instructions are stripped. After prior-use exclusions, normalized-text deduplication, and exact/13-word overlap screening against recorded external references, two disjoint batches each admit 1,500 DAPO-Math, 1,000 NuminaMath, and 1,000 DeepScaleR questions. Salted question-key hashing assigns 60%/15%/25% to fitting, selection, and reporting. The reporting portions supply the ID set, combining batches by dataset.

MATH-500 retains $500 - 189 - 16 = 295$ questions after integer and figure filtering; Gaokao retains $385 - 194 = 191$ after integer parsing. OlympiadBench retains 366 of 581 single-answer candidates after integer, length, and figure gates; its adapter separately excludes 93 multi-answer entries. AIME contest identities remove nine repetitions, retaining the original 2024 record without combining alternative trajectories. Missing identities for some AIME-2025 and AMC-2023 records prevent exhaustive semantic deduplication. Partial-coverage methods are unavailable outside aligned pools.

Baseline implementations. We adapt the released implementations of TC, ROM, CoDE-Stop, and DEER to the shared trajectory and evaluation protocol. No public OSP code was available at the time of reproduction, so we independently implemented it, strictly following its published architecture and expected-utility objective (Ehab et al., 2026). Its training configuration and shared-protocol adaptations are specified below; § E.4 reports performance checks supporting reproduction quality.

OS-Pruner implementation and training. OSP trains copies of the reasoner's final two decoder blocks and a sigmoid stopping head on their final-normalized boundary states; the trajectory-

generating reasoner remains frozen. Training uses exact expected utility on banked trajectories with strict correctness labels and total generated-token cost, including the forced answer. AdamW uses learning rate 10^{-5}, weight decay 0.01, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, gradient clipping at norm 1, and three trajectories per optimizer step. Computation uses bfloat16 with float32 trainable weights. The head starts at stopping probability 0.1. The training price follows $\lambda_e = \lambda\, 10^{1-\min(e,4)/4}$ for zero-based epoch e. Validation utility at the final price selects checkpoints from $e \geq 4$, with patience two. Qwen3 allows eight epochs and 16,384-token sequences; its five training prices are $\{2, 5, 10, 20, 40\} \times 10^{-6}$, with OSP-1 using 10^{-5}. OLMo3 OSP-1 uses 2×10^{-5}, seven epochs, and a 20,480-token sequence limit. Both use training seeds 0, 1, and 2 and the 320-question validation bank. Evaluation integrates stopping probabilities exactly over each trajectory, including natural completion.

Qwen3 policy families. Full OSP supplies five checkpoints per utility-training run, OSP-1 supplies 13 threshold policies, and BELT-B/H each supply 17 linear stopping policies per probe training run. BELT-B-1/H-1 use one policy fitted at $\lambda = 10^{-5}$, OSP-1's training price, and 13 source-validation logit-quantile thresholds. BELT and OSP use three independent training runs; ROM uses two independent training runs with its layer-16 R1 configuration. TC uses two independent training runs and six exported configurations: correctness and stability targets at three smoothing weights. Fixed caps are deterministic.

CoDE-Stop and DEER decode a trial answer with their own prompt at each stopping boundary and read its confidence. Each family combines raw, logistic-calibrated, and isotonic-calibrated thresholds (279 members), with calibrators fitted on the shared validation bank. Total cost includes trial-answer tokens. Family sizes and training effort differ; § A.7 specifies how they are evaluated.

A.2 Generation and answer elicitation

We pin model weights and tokenizers to `Qwen/Qwen3-1.7B`, revision `70d244cc86ccca08cf5af4e1e306ecf908b1ad5e`, and `allenai/Olmo-3-7B-Think`, revision `d97e442d7cc678210054dbcc9b440894d62c89a4`. Both use the native chat template with an assistant generation prompt; Qwen3 sets `enable_thinking=True`, and OLMo3 uses its template's default system turn and opening thinking tag. Reasoning and forced-answer decoding use temperature 0.6 and top-$p = 0.95$. Qwen3 uses top-$k = 20$; OLMo3 disables top-k. Both disable min-p filtering and use repetition penalty 1.

The user message is `Answer the following question. {instruction}\n<Question>: {question}`, where `\n` denotes a newline. For mathematics, `{instruction}` is `Reply with the answer only. Answer enclosed with \boxed{}.` For code, it is `Reply with the complete Python 3 program only, inside one ```python code block.` Code questions retain their function signature or starter code, or append `Read the input from standard input and write the answer to standard output.` for standard-input tasks.

At a forced-answer boundary, we append `</think>\n\n\boxed{` to the reasoning prefix for math, or `</think>\n\n```python\n` for code. Decoding stops at `}` or ```` ``` ````, respectively, retaining the closing delimiter. The opening answer delimiter is part of the prompt; only newly generated answer tokens enter the cost. Math queries allow 32 answer tokens; code queries allow 2,048. The reasoning and replication budgets are specified in §§ A.3, A.9 and A.10.

A.3 Training trajectories and belief labels

The source bank starts with 4,241 questions from the three training datasets. Qwen3 rollouts allow 12,000 thinking tokens and 128 blank-line-delimited blocks, with a forced-answer query of at most 32 tokens after every fourth block. The archive requests eight trajectories per question, or 16 for a historical subset; belief training retains at most eight, choosing the lowest eligible replicate indices. After filtering, 4,238 questions and 33,802 trajectories remain: 3,918 questions (31,248 trajectories) for probe training and 320 (2,554) for probe validation. The full validation bank selects the BELT probe checkpoint; disjoint 160-question halves then support policy fitting and development selection.

Trainable trajectories require reasoning-block boundaries and aligned forced-answer queries. Missing boundaries or queries exclude a trajectory from belief training; inconsistent query positions and duplicate boundary assignments raise validation errors. Incorrect answers and length truncation do not themselves cause exclusion. Evaluation retains trajectories without eligible queries and lets them complete naturally.

The labels follow Eq. (3). *End* repeats natural-completion correctness at every block boundary; *Later* is positive if a strictly later recorded forced answer or the natural-completion answer is correct. *Now* and *Same* are labelled at forced-answer positions. An unparseable forced answer has *Now* label zero. *Same* is masked when the natural-completion answer is unparseable; otherwise it indicates equality with that parsed answer, with zero assigned to an unparseable forced answer. Only available labels enter the belief loss.

A.4 BELT PROBE AND POLICY FITTING

Probe architecture. ReProbe's causal architecture encodes token-level states into a boundary representation and predicts outcome probabilities through a classification head:

$$h_i = E_\phi(H_i) = \left[\mathrm{Transformer}_\theta^{\mathrm{causal}}(\mathrm{MLP}_\psi(H_i) + M_i)\right]_{n_i}, \qquad \mathbf{p}_i = \sigma(W_{\mathrm{bel}} h_i + b_{\mathrm{bel}}), \quad (7)$$

Here MLP_ψ projects each token state through two linear layers with LayerNorm and GELU, and M_i supplies learned embeddings distinguishing ordinary tokens from block-end tokens. The causal Transformer encodes the projected sequence. The term $[\cdot]_{n_i}$ selects its state at the current boundary, the last of the n_i tokens in H_i. The parameters ϕ include the projection, marker embeddings and Transformer. The readout parameters $W_{\mathrm{bel}}, b_{\mathrm{bel}}$ map h_i to the predicted beliefs $\mathbf{p}_i$, with elementwise sigmoid σ. We adapt this architecture to predict outcomes of answering now and continuing for stopping control.

Instantiation and training. The Qwen3 BELT probe implements Eq. (7) with 2,048-dimensional layer-16 states, a width-768 token-wise projection, block-end markers, and two causal Transformer layers with eight attention heads. It reads the 768-dimensional representation at marked block ends and has 15,957,765 trainable parameters including belief heads. Inference uses only observed prefix states and, for BELT-B, prefix length; future answers, correctness, and final trajectory length are unavailable.

Table 8 gives the belief training settings. The trainer updates parameters in evaluation mode, disabling configured dropout 0.1. Each optimizer step averages the listed number of trajectory losses. Target weights are equal: Qwen3 sums the four losses ($4\mathcal{L}_{\mathrm{bel}}$), while OLMo3 averages them ($\mathcal{L}_{\mathrm{bel}}$). The minimum validation belief loss selects one checkpoint for all token prices. All Qwen3 runs stop after three epochs and select epoch 1; the three independent OLMo3 training runs complete three epochs and select epochs 2, 1, and 1, respectively (one-based).

For Eq. (5), belief probabilities are clipped to $[10^{-4}, 1 - 10^{-4}]$ before logit conversion. BELT-B adds prefix tokens divided by 12,000; BELT-H uses all 768 representation coordinates. Standardization uses policy-fitting boundaries, replacing zero SD with one. Table 9 gives the Qwen3 fitting grid and optimizer. Development utility selects regularization and iterate; reported prices are $\{2, 5, 10, 20, 50\} \times 10^{-6}$. Evaluation integrates stochastic stopping exactly on banked trajectories, including natural completion for those without eligible queries.

A.5 EFFICIENT PROBE PROCESSING FOR LONG TRAJECTORIES

We adapt ReProbe's causal step head[1] to observed blank-line boundaries. It marks all boundaries in one sequence, encodes the prefix causally, and gathers their representations. The implementation uses causal scaled dot-product attention, right-padded trajectory batches, and a bounded store of frozen reasoner features. Exported BELT probe features are reused across token prices.

A.6 METRICS, AGGREGATION AND UNCERTAINTY

For training run r and dataset d, average trajectories within questions, then questions, to obtain accuracy $a_{r,d}$ and total generated-token cost $c_{r,d}$. With natural cost c_d^{nat}, token saving is

[1] `https://github.com/ReProbe/ReProbe`, revision `fac0c16`.

Table 8: BELT probe training hyperparameters.

Setting	Qwen3	OLMo3
Optimizer	AdamW	
Encoder learning rate	5×10^{-5}	
Belief-head learning rate	10^{-3}	
AdamW (β_1, β_2); ϵ	$(0.9, 0.999)$; 10^{-8}	
Weight decay (all parameter groups)	0.01	
Gradient clipping (global norm)	1.0	
Learning-rate schedule / warm-up	Constant / none	
Trajectories per optimizer step	8	1
Additional gradient accumulation	None	
Maximum epochs	8	3
Early-stopping patience (epochs)	2	3
Validation frequency	Every epoch	
Checkpoint selection	Minimum validation belief loss	
Coefficient per target loss	1	$1/4$
Dropout during training	Disabled (evaluation mode)	
Independent training runs	3	

Table 9: Qwen3 linear stopping policy fitting hyperparameters.

Setting	Value
Optimizer / precision	L-BFGS / float64
Learning rate / line search	1.0 / strong Wolfe
History size	20
Maximum iterations	500
Gradient / change tolerance	10^{-9} / 10^{-12}
Development evaluation	Every 25 iterations
Initial weights	$w = 0$
Initial intercepts	$b = \mathrm{logit}(p)$, $p \in \{0.1, 0.01\}$
Initial regularization grid	$\kappa \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$
Grid extension	If the largest κ is selected, extend by $\times 10$ up to 10^4
Penalized parameters	w only; intercept unpenalized
Token-cost-weight grid	$2 \times 10^{-6} 20^{k/12}$, $k = 0, \ldots, 16$
Selection criterion	Maximum development utility

$100(1 - c_{r,d}/c_d^{\mathrm{nat}})$. Macro averages weight dataset metrics equally. Micro accuracy weights questions equally; micro saving uses pooled costs, not averaged saving fractions. Main tables report macro averages; Tables 10, 16 and 24 also report micro averages. The Avg. columns pool ID and OOD. An aggregate is unavailable if a required dataset, question, or training run is missing.

Learned-method metrics are computed per independent training run, then summarized by mean and sample SD (denominator $S-1$). Prose values and main table entries are these means; table subscripts give sample SD, with variance retained in exports. For methods evaluated in one training run, we cannot estimate variation across training runs. Deterministic controls have no training-run variation. Policy-fitting starts, where varied instead, are identified separately; repeating evaluations does not create independent training runs.

For Natural, Fixed cap, CoDE-Stop, and DEER, ‡ denotes a question-bootstrap standard error. We draw 5,000 samples independently within each dataset, preserving its size, from a pseudorandom sequence fixed by the implementation, and retain each question's trajectory-averaged outcomes. Method and natural costs use the same sampled identities. Each draw recomputes dataset and aggregate metrics, including pooled-cost micro saving. Frontier members and mixture weights remain fixed, so these errors exclude frontier-selection uncertainty and variation across training runs.

Independent training runs are not paired across methods. Question resampling conditions on trained models and cannot replace uncertainty estimated from variation across training runs.

A.7 Operating points on accuracy–cost frontiers

Test-frontier operating points. For each method, independent training run, and distribution, we evaluate the complete exported family at micro average accuracy and mean total generated-token cost, adding natural completion as the never-stop policy. Nondominated members form the frontier in ascending cost order. Linear interpolation between adjacent points corresponds to randomizing between their policies with fixed probabilities.

At an accuracy tolerance in percentage points, we report the cheapest frontier point within that drop from natural accuracy. At a token-saving target, we report accuracy at the corresponding fraction of natural cost; costs below the cheapest member are infeasible. The single-member comparison in Tables 43 to 44 instead selects the most accurate member within budget, without interpolation. Per-dataset and macro values evaluate the same distribution-level policy or mixture. Each independent training run is read separately before computing the mean and sample SD across runs.

Two utility protocols. Test-family-max utility selects the member or natural completion with the highest test utility at each price (§ B.4). Utility of dev-selected policies instead selects on development data before test evaluation (§§ B.5 and C.1). Utility fixes the accuracy–cost trade-off without requiring a matched constraint. In contrast, test-frontier operating points hold test accuracy or cost fixed; a constraint selected on source development data need not hold on the target distribution.

Frontier plots. Curves average interpolated accuracy at common costs over the shared support of the independent training runs, without extrapolation. Per-dataset panels project the distribution-level mixtures. Uniform markers identify selected mean-curve knots, including endpoints; they are not the table operating points or necessarily single policies. Tables average their constrained readings separately, so a mean cost at fixed accuracy need not lie on the plotted mean-accuracy curve. Validation supports checkpoint, calibration, and hyperparameter selection; test data determine frontier readings after the families are frozen.

A.8 Compute resources

Experiments use NVIDIA GH200, RTX 4090, and RTX PRO 6000 GPUs. Qwen3 generation on RTX 4090 uses vLLM 0.9.0.1, Transformers 4.52.4, and PyTorch `2.7.0+cu126`. The archived GH200 generation image contains vLLM build `0.1.dev1+gf7e62e3d6.d20260427.cu132`, Transformers 4.57.6, and PyTorch `2.11.0a0+a6c236b9fd.nv26.3.46836102`. A versioned roster fixes question identities, benchmark membership, and duplicate exclusions. Primary Qwen3 exports record this roster, frontier members and mixture weights, grading, and cost conventions. Table rendering formats supplied cells without combining incomplete populations or inferring variance. § D.2 specifies the scope of the training-cost estimates.

A.9 OLMo3 cross-family replication

OLMo3 uses its native thinking template, up to four trajectories per question, and a 16,384-token generation cap. We train a new BELT probe on layer-18 states using the same two-layer, width-768 architecture and four belief targets. This replicates the training recipe without transferring Qwen3 weights. Of 4,241 source questions, 320 are held out for probe validation and split into 160-question policy-fitting and development sets. Archived policy fits use loose utility labels; test evaluation uses strict grading and total generated tokens.

All comparisons use the main 1,025-question OOD roster, with natural accuracy 81.83% (micro average; 78.4% macro, Table 3) and mean cost 6,868 tokens. Metrics are aggregated separately for each backbone. OLMo3 BELT-B omits prefix length, giving four standardized belief logits and five policy parameters including the intercept.

Full BELT-B/H families have 221 members per probe training run. TC has 150 across three smoothing settings, ROM 21 thresholds, OSP-1 13, CoDE-Stop 75 configurations, DEER 18, and Fixed cap eight. BELT-B-1/H-1 fit one policy at $\lambda = 2 \times 10^{-5}$ and use 13 thresholds. Every family includes natural completion. BELT, TC, ROM, and OSP-1 each use three independent training runs; OSP-1

trains a two-block branch and selects its checkpoint by validation utility. Runs are not paired across methods. Operating points and test-family-max utility follow § A.7, with full results in Tables 24 and 36.

A.10 MATH TO CODE DOMAIN TRANSFER

LiveCodeBench provides 549 problems before August 2024 (Pre) and 416 from August 2024 onward (Post). Pre precedes the backbone's stated cutoff and may overlap its training period. Qwen3 generates eight trajectories per question with a 12,000-token thinking cap and a forced-answer query every fourth reasoning block. Forced answers and natural completion produce programs graded by unit tests; programs that fail to complete within 2,048 tokens count as wrong. Total generated cost includes the forced program.

Tests come from the public and decoded private tests in `livecodebench/code_generation_lite`, revision `0fe84c3912ea0c4d4a78037083943e8f0c4dd505`. We retain up to 15 tests per problem, considering longer serialized inputs first (ties by source order), excluding inputs over 1 MiB and enforcing a 4 MiB total serialized-test limit. Admission requires at least five retained tests. Each test runs in a fresh `python -I` subprocess with 10-second CPU and wall-clock limits, 1 GiB address-space limit, and 8 MiB file-size limit. Standard-output comparison normalizes whitespace, checks integer tokens exactly, and allows floating-point error $10^{-6}\max(1, |y|)$ for reference value y. Function-call tests use the supplied function entry point. A program is correct only if all retained tests pass; malformed programs, runtime errors, timeouts, and memory failures count as incorrect.

All stopping-controller and policy weights are trained on mathematics; the pretrained reasoner remains frozen. Each pool receives 17 BELT-B/H policies per probe training run, 13 OSP-1 thresholds per utility-training run, five full-OSP checkpoints per utility-training run, and 13 BELT-B-1/H-1 thresholds per probe training run, plus natural completion. Frontier operating points are selected on code test outcomes; dev-selected utility instead retains mathematics-side selection (§ B.5). For each method, we compute every metric separately for three independent training runs and report the mean and sample SD across those runs; runs are not paired across methods.

B DETAILED RESULTS BY DATASET AND DISTRIBUTION

The tables below expand the main results using the aggregation and uncertainty conventions in § A.6. Each dataset inherits its distribution-level operating point, so its accuracy or cost need not independently satisfy that constraint. Avg. columns pool all ID/OOD questions for micro averages and weight all datasets equally for macro averages. Natural and Fixed cap appear in accuracy tables but are omitted from token-saving tables.

B.1 PER-DATASET RESULTS

Table 10: Token saving given the accuracy tolerance, reported by dataset. Reporting convention follows Table 1. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 0 pp tolerance, ID and pooled averages.

Method	Probe Params	DAPO	Numina	DSR	Micro	Macro
CoDE-Stop‡	0	$30.31_{\pm 0.65}$	$24.35_{\pm 1.02}$	$29.27_{\pm 0.99}$	$32.14_{\pm 0.39}$	$36.21_{\pm 0.48}$
DEER‡	0	$24.80_{\pm 1.02}$	$29.22_{\pm 1.50}$	$34.17_{\pm 1.39}$	$29.16_{\pm 0.58}$	$30.22_{\pm 0.76}$
TC	514	$35.10_{\pm 0.39}$	$39.08_{\pm 1.04}$	$39.28_{\pm 0.59}$	$39.07_{\pm 0.73}$	$40.76_{\pm 0.93}$
ROM	5M	$26.71_{\pm 1.29}$	$31.51_{\pm 1.18}$	$35.46_{\pm 0.11}$	$30.84_{\pm 1.30}$	$31.98_{\pm 1.05}$
OSP	101M $\times 5$	$56.19_{\pm 0.53}$	$66.17_{\pm 3.66}$	$57.64_{\pm 1.83}$	$61.25_{\pm 0.89}$	$62.38_{\pm 0.36}$
OSP-1	101M	$53.29_{\pm 3.72}$	$65.05_{\pm 2.76}$	$54.75_{\pm 5.34}$	$59.41_{\pm 2.90}$	$61.22_{\pm 2.12}$
BELT-B-1	16M	$49.51_{\pm 2.35}$	$59.32_{\pm 4.70}$	$51.90_{\pm 4.11}$	$52.25_{\pm 3.67}$	$51.16_{\pm 3.80}$
BELT-H-1	16M	$56.34_{\pm 1.01}$	$64.81_{\pm 0.89}$	$55.47_{\pm 0.62}$	$58.72_{\pm 1.18}$	$58.01_{\pm 1.75}$
BELT-B	16M	$56.24_{\pm 1.83}$	$64.13_{\pm 1.49}$	$56.79_{\pm 0.85}$	$60.57_{\pm 0.97}$	$61.56_{\pm 0.79}$
BELT-H	16M	$59.79_{\pm 0.65}$	$64.86_{\pm 0.51}$	$55.71_{\pm 0.63}$	$62.64_{\pm 0.55}$	$63.91_{\pm 0.71}$

Table 11: Token saving given the accuracy tolerance, reported by dataset. Reporting convention follows Table 1. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 0 pp tolerance, OOD.

Method	Probe Params	M500	Olymp.	GK	AIME	AMC
CoDE-Stop‡	0	$44.21_{\pm 1.61}$	$37.84_{\pm 0.98}$	$40.98_{\pm 1.50}$	$39.51_{\pm 1.63}$	$43.23_{\pm 2.02}$
DEER‡	0	$44.64_{\pm 2.41}$	$33.78_{\pm 1.57}$	$35.13_{\pm 2.55}$	$12.58_{\pm 2.07}$	$27.45_{\pm 3.49}$
TC	514	$44.31_{\pm 0.43}$	$45.02_{\pm 0.92}$	$44.55_{\pm 0.68}$	$38.00_{\pm 1.83}$	$40.76_{\pm 2.42}$
ROM	5M	$43.69_{\pm 0.53}$	$33.93_{\pm 4.32}$	$38.06_{\pm 0.47}$	$16.47_{\pm 1.05}$	$30.04_{\pm 0.75}$
OSP	101M $\times 5$	$53.48_{\pm 1.59}$	$69.65_{\pm 0.26}$	$55.89_{\pm 1.91}$	$73.04_{\pm 0.55}$	$66.97_{\pm 0.68}$
OSP-1	101M	$53.48_{\pm 3.01}$	$69.16_{\pm 1.08}$	$54.85_{\pm 2.91}$	$71.27_{\pm 2.44}$	$67.91_{\pm 1.57}$
BELT-B-1	16M	$51.95_{\pm 1.81}$	$55.89_{\pm 6.39}$	$50.33_{\pm 3.31}$	$42.00_{\pm 4.70}$	$48.34_{\pm 4.33}$
BELT-H-1	16M	$53.24_{\pm 2.80}$	$64.10_{\pm 2.50}$	$54.30_{\pm 2.76}$	$58.37_{\pm 1.73}$	$57.48_{\pm 3.20}$
BELT-B	16M	$53.06_{\pm 1.15}$	$69.67_{\pm 0.92}$	$57.48_{\pm 1.00}$	$70.16_{\pm 0.99}$	$64.96_{\pm 0.31}$
BELT-H	16M	$54.95_{\pm 1.13}$	$71.89_{\pm 0.95}$	$58.57_{\pm 0.74}$	$76.10_{\pm 1.23}$	$69.40_{\pm 1.55}$

Table 12: Token saving given the accuracy tolerance, reported by dataset. Reporting convention follows Table 1. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 1 pp tolerance, ID and pooled averages.

Method	Probe Params	DAPO	Numina	DSR	Micro	Macro
CoDE-Stop‡	0	$35.95_{\pm 0.67}$	$34.48_{\pm 1.08}$	$38.01_{\pm 1.01}$	$38.77_{\pm 0.40}$	$42.03_{\pm 0.50}$
DEER‡	0	$27.15_{\pm 1.06}$	$32.50_{\pm 1.52}$	$36.99_{\pm 1.37}$	$31.59_{\pm 0.60}$	$32.37_{\pm 0.79}$
TC	514	$38.48_{\pm 0.29}$	$42.61_{\pm 0.75}$	$42.38_{\pm 0.95}$	$42.34_{\pm 0.45}$	$43.93_{\pm 0.45}$
ROM	5M	$29.54_{\pm 1.74}$	$35.54_{\pm 1.01}$	$38.42_{\pm 0.14}$	$33.93_{\pm 1.48}$	$35.00_{\pm 1.36}$
OSP	101M $\times 5$	$58.86_{\pm 0.37}$	$68.50_{\pm 2.59}$	$59.68_{\pm 0.93}$	$63.77_{\pm 0.34}$	$65.03_{\pm 0.13}$
OSP-1	101M	$56.82_{\pm 3.07}$	$67.77_{\pm 2.23}$	$57.59_{\pm 4.49}$	$62.61_{\pm 2.38}$	$64.45_{\pm 1.75}$
BELT-B-1	16M	$52.24_{\pm 2.93}$	$62.00_{\pm 4.76}$	$54.13_{\pm 4.00}$	$54.63_{\pm 3.89}$	$53.29_{\pm 3.95}$
BELT-H-1	16M	$58.81_{\pm 1.31}$	$67.09_{\pm 0.84}$	$57.54_{\pm 0.64}$	$60.89_{\pm 1.42}$	$60.05_{\pm 2.04}$
BELT-B	16M	$59.44_{\pm 1.88}$	$66.66_{\pm 1.94}$	$59.18_{\pm 0.92}$	$63.29_{\pm 1.07}$	$64.16_{\pm 0.69}$
BELT-H	16M	$63.18_{\pm 0.81}$	$68.56_{\pm 0.34}$	$59.39_{\pm 0.32}$	$65.86_{\pm 0.47}$	$66.90_{\pm 0.43}$

Table 13: Token saving given the accuracy tolerance, reported by dataset. Reporting convention follows Table 1. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 1 pp tolerance, OOD.

Method	Probe Params	M500	Olymp.	GK	AIME	AMC
CoDE-Stop‡	0	$49.30_{\pm 1.52}$	$43.18_{\pm 1.06}$	$46.76_{\pm 1.48}$	$41.08_{\pm 1.72}$	$47.44_{\pm 2.20}$
DEER‡	0	$45.95_{\pm 2.42}$	$35.86_{\pm 1.60}$	$36.53_{\pm 2.57}$	$14.12_{\pm 2.23}$	$29.90_{\pm 3.73}$
TC	514	$46.53_{\pm 0.62}$	$48.26_{\pm 0.05}$	$47.16_{\pm 0.03}$	$41.38_{\pm 0.85}$	$44.65_{\pm 1.35}$
ROM	5M	$45.72_{\pm 0.83}$	$36.95_{\pm 4.25}$	$42.46_{\pm 2.63}$	$18.74_{\pm 0.81}$	$32.65_{\pm 1.39}$
OSP	101M ×5	$55.79_{\pm 2.14}$	$72.60_{\pm 0.74}$	$58.21_{\pm 1.90}$	$74.94_{\pm 2.35}$	$71.68_{\pm 1.27}$
OSP-1	101M	$55.67_{\pm 2.33}$	$72.51_{\pm 0.55}$	$58.13_{\pm 2.12}$	$75.15_{\pm 2.60}$	$71.92_{\pm 0.91}$
BELT-B-1	16M	$53.25_{\pm 1.81}$	$57.80_{\pm 6.29}$	$51.94_{\pm 3.13}$	$44.63_{\pm 5.23}$	$50.36_{\pm 4.51}$
BELT-H-1	16M	$54.43_{\pm 2.86}$	$65.84_{\pm 2.82}$	$55.89_{\pm 3.02}$	$60.94_{\pm 2.30}$	$59.87_{\pm 3.58}$
BELT-B	16M	$54.70_{\pm 0.99}$	$71.97_{\pm 0.55}$	$59.73_{\pm 0.61}$	$73.65_{\pm 0.30}$	$67.91_{\pm 0.61}$
BELT-H	16M	$56.78_{\pm 0.67}$	$74.23_{\pm 0.20}$	$\underline{61.63}_{\pm 0.57}$	$\underline{78.99}_{\pm 0.31}$	$72.46_{\pm 1.05}$

Table 14: Token saving given the accuracy tolerance, reported by dataset. Reporting convention follows Table 1. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 2 pp tolerance, ID and pooled averages.

Method	Probe Params	DAPO	Numina	DSR	Micro	Macro
CoDE-Stop‡	0	$40.23_{\pm 0.75}$	$41.03_{\pm 1.26}$	$43.50_{\pm 1.14}$	$42.94_{\pm 0.44}$	$45.35_{\pm 0.53}$
DEER‡	0	$28.99_{\pm 1.09}$	$35.12_{\pm 1.52}$	$39.16_{\pm 1.35}$	$33.42_{\pm 0.60}$	$33.93_{\pm 0.80}$
TC	514	$42.23_{\pm 0.18}$	$46.21_{\pm 0.70}$	$45.58_{\pm 0.41}$	$45.51_{\pm 0.79}$	$46.69_{\pm 1.26}$
ROM	5M	$31.75_{\pm 2.26}$	$38.49_{\pm 1.42}$	$41.06_{\pm 0.68}$	$36.48_{\pm 1.87}$	$37.63_{\pm 1.71}$
OSP	101M ×5	$61.53_{\pm 0.57}$	$\underline{70.83}_{\pm 1.53}$	$\underline{61.71}_{\pm 0.08}$	$\underline{65.94}_{\pm 0.31}$	$66.93_{\pm 0.56}$
OSP-1	101M	$60.35_{\pm 2.42}$	$70.50_{\pm 1.70}$	$60.44_{\pm 3.65}$	$65.50_{\pm 1.92}$	$\underline{67.04}_{\pm 1.48}$
BELT-B-1	16M	$54.54_{\pm 2.92}$	$64.24_{\pm 4.36}$	$56.09_{\pm 3.74}$	$56.72_{\pm 3.81}$	$55.25_{\pm 3.97}$
BELT-H-1	16M	$60.89_{\pm 1.27}$	$68.95_{\pm 0.92}$	$59.32_{\pm 0.69}$	$62.78_{\pm 1.47}$	$61.90_{\pm 2.20}$
BELT-B	16M	$62.32_{\pm 1.87}$	$68.89_{\pm 1.86}$	$61.45_{\pm 0.77}$	$65.67_{\pm 1.33}$	$66.32_{\pm 1.23}$
BELT-H	16M	$\mathbf{65.60}_{\pm 1.04}$	$\mathbf{71.13}_{\pm 0.28}$	$\mathbf{62.26}_{\pm 0.56}$	$\mathbf{68.36}_{\pm 0.46}$	$\mathbf{69.41}_{\pm 0.29}$

Table 15: Token saving given the accuracy tolerance, reported by dataset. Reporting convention follows Table 1. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 2 pp tolerance, OOD.

Method	Probe Params	M500	Olymp.	GK	AIME	AMC
CoDE-Stop‡	0	$44.72_{\pm 1.79}$	$44.04_{\pm 1.00}$	$44.90_{\pm 1.88}$	$53.16_{\pm 1.52}$	$51.23_{\pm 2.13}$
DEER‡	0	$46.98_{\pm 2.40}$	$37.18_{\pm 1.60}$	$37.72_{\pm 2.54}$	$15.01_{\pm 2.30}$	$31.33_{\pm 3.80}$
TC	514	$48.93_{\pm 0.21}$	$50.51_{\pm 1.56}$	$49.32_{\pm 1.33}$	$44.17_{\pm 2.99}$	$46.55_{\pm 3.12}$
ROM	5M	$47.55_{\pm 1.35}$	$39.66_{\pm 3.90}$	$46.18_{\pm 4.15}$	$21.26_{\pm 0.64}$	$35.11_{\pm 1.99}$
OSP	101M ×5	$56.90_{\pm 1.10}$	$73.80_{\pm 0.80}$	$60.41_{\pm 0.25}$	$77.96_{\pm 0.90}$	$72.28_{\pm 1.58}$
OSP-1	101M	$\underline{57.50}_{\pm 1.76}$	$\underline{74.73}_{\pm 0.55}$	$60.45_{\pm 1.89}$	$78.10_{\pm 2.51}$	$\underline{74.28}_{\pm 1.30}$
BELT-B-1	16M	$54.52_{\pm 1.84}$	$59.63_{\pm 6.27}$	$53.46_{\pm 3.07}$	$47.21_{\pm 5.80}$	$52.33_{\pm 4.72}$
BELT-H-1	16M	$55.58_{\pm 2.90}$	$67.47_{\pm 3.09}$	$57.41_{\pm 3.26}$	$63.38_{\pm 2.84}$	$62.18_{\pm 3.90}$
BELT-B	16M	$56.13_{\pm 1.51}$	$73.92_{\pm 1.46}$	$61.43_{\pm 1.12}$	$76.39_{\pm 1.69}$	$70.05_{\pm 1.53}$
BELT-H	16M	$\mathbf{58.86}_{\pm 0.42}$	$\mathbf{76.32}_{\pm 0.38}$	$\mathbf{63.98}_{\pm 0.49}$	$\mathbf{81.88}_{\pm 0.23}$	$\mathbf{75.25}_{\pm 0.46}$

Table 16: Acc. controlling the token budget, reported by dataset. Reporting convention follows Table 2. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 70% token saving, ID and pooled averages.

Method	Probe Params	DAPO	Numina	DSR	Micro	Macro
Natural†‡	–	$52.91_{\pm 1.51}$	$37.74_{\pm 1.94}$	$55.90_{\pm 1.86}$	$55.63_{\pm 0.77}$	$56.18_{\pm 0.97}$
Fixed cap‡	0	$26.08_{\pm 1.30}$	$26.73_{\pm 1.66}$	$34.63_{\pm 1.78}$	$34.70_{\pm 0.71}$	$35.06_{\pm 0.87}$
CoDE-Stop‡	0	$30.75_{\pm 1.31}$	$30.40_{\pm 1.74}$	$37.68_{\pm 1.83}$	$40.02_{\pm 0.73}$	$40.81_{\pm 0.92}$
DEER‡	0	$19.43_{\pm 0.96}$	$21.96_{\pm 1.45}$	$25.03_{\pm 1.48}$	$24.07_{\pm 0.59}$	$23.54_{\pm 0.70}$
TC	514	$26.59_{\pm 0.58}$	$30.39_{\pm 1.10}$	$35.71_{\pm 0.73}$	$36.89_{\pm 1.09}$	$37.52_{\pm 1.20}$
ROM	5M	$26.57_{\pm 1.19}$	$26.80_{\pm 1.40}$	$32.86_{\pm 2.06}$	$34.39_{\pm 2.44}$	$34.90_{\pm 2.29}$
OSP	101M ×5	$46.17_{\pm 0.85}$	$38.22_{\pm 0.80}$	$49.35_{\pm 0.17}$	NF	NF
OSP-1	101M	$46.20_{\pm 0.98}$	$37.88_{\pm 0.75}$	$49.23_{\pm 0.22}$	$52.04_{\pm 0.55}$	$52.80_{\pm 0.45}$
BELT-B-1	16M	$39.23_{\pm 3.76}$	$32.58_{\pm 3.55}$	$42.86_{\pm 4.06}$	$43.04_{\pm 4.58}$	$43.16_{\pm 4.94}$
BELT-H-1	16M	$43.26_{\pm 0.91}$	$36.69_{\pm 1.86}$	$48.03_{\pm 2.01}$	$48.29_{\pm 2.31}$	$48.39_{\pm 2.65}$
BELT-B	16M	$45.45_{\pm 0.46}$	$37.46_{\pm 0.64}$	$49.28_{\pm 0.93}$	$51.41_{\pm 0.81}$	$52.13_{\pm 0.97}$
BELT-H	16M	$46.76_{\pm 0.21}$	$38.67_{\pm 0.54}$	$51.49_{\pm 0.41}$	$53.05_{\pm 0.16}$	$53.67_{\pm 0.19}$

Table 17: Acc. controlling the token budget, reported by dataset. Reporting convention follows Table 2. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 70% token saving, OOD.

Method	Probe Params	M500	Olymp.	GK	AIME	AMC
Natural†‡	–	$79.15_{\pm 1.80}$	$65.57_{\pm 2.19}$	$64.53_{\pm 2.92}$	$29.46_{\pm 3.71}$	$64.17_{\pm 4.47}$
Fixed cap‡	0	$62.84_{\pm 2.05}$	$40.44_{\pm 2.06}$	$47.77_{\pm 2.88}$	$6.18_{\pm 1.93}$	$35.84_{\pm 4.54}$
CoDE-Stop‡	0	$73.65_{\pm 1.84}$	$46.19_{\pm 2.15}$	$57.27_{\pm 2.97}$	$8.97_{\pm 2.43}$	$41.57_{\pm 4.71}$
DEER‡	0	$38.66_{\pm 2.11}$	$23.28_{\pm 1.60}$	$35.32_{\pm 2.68}$	$5.17_{\pm 1.22}$	$19.47_{\pm 3.35}$
TC	514	$64.38_{\pm 2.21}$	$43.96_{\pm 1.41}$	$54.69_{\pm 1.76}$	$9.25_{\pm 0.44}$	$35.17_{\pm 1.39}$
ROM	5M	$62.36_{\pm 7.76}$	$38.75_{\pm 3.16}$	$48.92_{\pm 4.40}$	$9.16_{\pm 2.16}$	$33.76_{\pm 0.55}$
OSP	101M ×5	NF	NF	NF	NF	NF
OSP-1	101M	$85.44_{\pm 0.21}$	$57.28_{\pm 0.61}$	$69.51_{\pm 1.12}$	$21.30_{\pm 1.47}$	$55.54_{\pm 0.39}$
BELT-B-1	16M	$62.14_{\pm 7.70}$	$49.28_{\pm 5.31}$	$53.61_{\pm 6.97}$	$18.90_{\pm 3.30}$	$46.65_{\pm 5.48}$
BELT-H-1	16M	$72.01_{\pm 7.30}$	$55.12_{\pm 2.37}$	$60.67_{\pm 3.92}$	$20.62_{\pm 0.39}$	$50.75_{\pm 3.30}$
BELT-B	16M	$82.91_{\pm 2.59}$	$57.62_{\pm 0.23}$	$67.94_{\pm 2.23}$	$20.72_{\pm 0.63}$	$55.66_{\pm 1.04}$
BELT-H	16M	$85.84_{\pm 0.35}$	$59.02_{\pm 0.23}$	$69.88_{\pm 0.51}$	$20.83_{\pm 0.45}$	$56.90_{\pm 0.29}$

Table 18: Acc. controlling the token budget, reported by dataset. Reporting convention follows Table 2. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 60% token saving, ID and pooled averages.

Method	Probe Params	DAPO	Numina	DSR	Micro	Macro
Natural†‡	–	$52.91_{\pm 1.51}$	$37.74_{\pm 1.94}$	$55.90_{\pm 1.86}$	$55.63_{\pm 0.77}$	$56.18_{\pm 0.97}$
Fixed cap‡	0	$32.94_{\pm 1.31}$	$30.22_{\pm 1.71}$	$39.60_{\pm 1.76}$	$40.53_{\pm 0.71}$	$40.72_{\pm 0.91}$
CoDE-Stop‡	0	$39.70_{\pm 1.37}$	$34.10_{\pm 1.79}$	$43.49_{\pm 1.79}$	$47.00_{\pm 0.73}$	$47.52_{\pm 0.94}$
DEER‡	0	$28.31_{\pm 1.14}$	$25.93_{\pm 1.56}$	$31.36_{\pm 1.62}$	$31.59_{\pm 0.64}$	$30.76_{\pm 0.74}$
TC	514	$36.84_{\pm 0.06}$	$36.05_{\pm 0.41}$	$44.03_{\pm 0.31}$	$45.62_{\pm 0.39}$	$46.17_{\pm 0.64}$
ROM	5M	$34.41_{\pm 1.17}$	$31.54_{\pm 1.31}$	$38.85_{\pm 1.60}$	$41.19_{\pm 1.86}$	$41.56_{\pm 1.84}$
OSP	101M ×5	$51.43_{\pm 0.97}$	$41.47_{\pm 0.39}$	$53.77_{\pm 0.51}$	$56.17_{\pm 0.55}$	$56.62_{\pm 0.49}$
OSP-1	101M	$50.98_{\pm 1.23}$	$40.87_{\pm 1.02}$	$52.66_{\pm 0.48}$	$55.58_{\pm 0.70}$	$56.11_{\pm 0.69}$
BELT-B-1	16M	$47.88_{\pm 2.40}$	$38.28_{\pm 1.44}$	$50.17_{\pm 2.72}$	$51.20_{\pm 2.55}$	$51.22_{\pm 2.91}$
BELT-H-1	16M	$50.74_{\pm 0.28}$	$40.70_{\pm 0.76}$	$53.90_{\pm 0.63}$	$55.03_{\pm 0.72}$	$55.37_{\pm 0.93}$
BELT-B	16M	$50.89_{\pm 0.53}$	$40.65_{\pm 0.37}$	$54.20_{\pm 1.07}$	$55.94_{\pm 0.49}$	$56.55_{\pm 0.62}$
BELT-H	16M	$51.17_{\pm 0.04}$	$40.94_{\pm 0.25}$	$55.59_{\pm 0.14}$	$56.56_{\pm 0.12}$	$57.15_{\pm 0.14}$

Table 19: Acc. controlling the token budget, reported by dataset. Reporting convention follows Table 2. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 60% token saving, OOD.

Method	Probe Params	M500	Olymp.	GK	AIME	AMC
Natural†‡	–	$79.15_{\pm 1.80}$	$65.57_{\pm 2.19}$	$64.53_{\pm 2.92}$	$29.46_{\pm 3.71}$	$64.17_{\pm 4.47}$
Fixed cap‡	0	$68.95_{\pm 1.99}$	$49.21_{\pm 2.12}$	$52.25_{\pm 2.92}$	$9.03_{\pm 2.31}$	$43.54_{\pm 4.76}$
CoDE-Stop‡	0	$81.36_{\pm 1.63}$	$54.90_{\pm 2.18}$	$64.57_{\pm 2.89}$	$12.31_{\pm 2.73}$	$49.73_{\pm 4.97}$
DEER‡	0	$48.65_{\pm 2.01}$	$33.57_{\pm 1.78}$	$41.93_{\pm 2.55}$	$9.01_{\pm 1.69}$	$27.29_{\pm 3.61}$
TC	514	$73.06_{\pm 0.25}$	$54.87_{\pm 1.08}$	$62.19_{\pm 0.19}$	$16.23_{\pm 1.67}$	$46.06_{\pm 1.25}$
ROM	5M	$70.27_{\pm 5.80}$	$47.16_{\pm 1.34}$	$54.69_{\pm 3.09}$	$13.94_{\pm 0.64}$	$41.64_{\pm 1.07}$
OSP	101M ×5	$87.58_{\pm 0.13}$	$62.05_{\pm 0.57}$	$72.06_{\pm 1.26}$	$23.88_{\pm 0.38}$	$60.72_{\pm 0.44}$
OSP-1	101M	$87.18_{\pm 0.50}$	$61.18_{\pm 0.47}$	$72.01_{\pm 0.48}$	$25.13_{\pm 2.08}$	$58.89_{\pm 0.88}$
BELT-B-1	16M	$73.65_{\pm 5.16}$	$58.67_{\pm 2.03}$	$62.32_{\pm 4.18}$	$23.50_{\pm 2.88}$	$55.30_{\pm 3.30}$
BELT-H-1	16M	$82.07_{\pm 2.86}$	$61.86_{\pm 0.70}$	$68.38_{\pm 2.41}$	$25.82_{\pm 0.50}$	$59.50_{\pm 0.74}$
BELT-B	16M	$87.23_{\pm 0.52}$	$61.88_{\pm 0.37}$	$72.20_{\pm 0.48}$	$25.20_{\pm 1.24}$	$60.15_{\pm 1.39}$
BELT-H	16M	$87.65_{\pm 0.44}$	$62.81_{\pm 0.13}$	$73.10_{\pm 0.13}$	$24.60_{\pm 0.23}$	$61.32_{\pm 0.29}$

Table 20: Acc. controlling the token budget, reported by dataset. Reporting convention follows Table 2. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 50% token saving, ID and pooled averages.

Method	Probe Params	DAPO	Numina	DSR	Micro	Macro
Natural†‡	–	$52.91_{\pm 1.51}$	$37.74_{\pm 1.94}$	$55.90_{\pm 1.86}$	$55.63_{\pm 0.77}$	$56.18_{\pm 0.97}$
Fixed cap‡	0	$39.53_{\pm 1.44}$	$33.60_{\pm 1.86}$	$44.41_{\pm 1.84}$	$45.72_{\pm 0.76}$	$45.57_{\pm 0.96}$
CoDE-Stop‡	0	$45.92_{\pm 1.45}$	$37.44_{\pm 1.88}$	$49.15_{\pm 1.81}$	$51.58_{\pm 0.75}$	$51.72_{\pm 0.94}$
DEER‡	0	$37.40_{\pm 1.27}$	$31.44_{\pm 1.67}$	$37.62_{\pm 1.69}$	$40.01_{\pm 0.68}$	$39.20_{\pm 0.80}$
TC	514	$44.46_{\pm 0.04}$	$39.63_{\pm 0.25}$	$49.56_{\pm 0.11}$	$51.60_{\pm 0.15}$	$52.04_{\pm 0.20}$
ROM	5M	$41.30_{\pm 0.68}$	$35.47_{\pm 1.64}$	$44.29_{\pm 0.95}$	$47.07_{\pm 1.23}$	$47.32_{\pm 1.37}$
OSP	101M $\times 5$	$54.48_{\pm 0.33}$	$43.48_{\pm 0.43}$	$56.70_{\pm 0.26}$	$58.69_{\pm 0.21}$	$59.07_{\pm 0.32}$
OSP-1	101M	$54.00_{\pm 1.40}$	$42.65_{\pm 1.35}$	$55.16_{\pm 0.44}$	$57.90_{\pm 0.83}$	$58.35_{\pm 0.87}$
BELT-B-1	16M	$53.01_{\pm 1.87}$	$41.69_{\pm 0.72}$	$55.34_{\pm 0.95}$	$56.36_{\pm 1.34}$	$56.51_{\pm 1.55}$
BELT-H-1	16M	$55.00_{\pm 0.24}$	$43.14_{\pm 0.29}$	$57.40_{\pm 0.13}$	$58.76_{\pm 0.14}$	$59.16_{\pm 0.16}$
BELT-B	16M	$54.52_{\pm 0.39}$	$42.44_{\pm 0.64}$	$56.98_{\pm 0.54}$	$58.54_{\pm 0.29}$	$59.04_{\pm 0.26}$
BELT-H	16M	$54.19_{\pm 0.03}$	$42.70_{\pm 0.20}$	$57.82_{\pm 0.19}$	$58.81_{\pm 0.18}$	$59.29_{\pm 0.22}$

Table 21: Acc. controlling the token budget, reported by dataset. Reporting convention follows Table 2. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 50% token saving, OOD.

Method	Probe Params	M500	Olymp.	GK	AIME	AMC
Natural†‡	–	$79.15_{\pm 1.80}$	$65.57_{\pm 2.19}$	$64.53_{\pm 2.92}$	$29.46_{\pm 3.71}$	$64.17_{\pm 4.47}$
Fixed cap‡	0	$73.34_{\pm 1.97}$	$56.63_{\pm 2.21}$	$55.25_{\pm 3.00}$	$11.89_{\pm 2.72}$	$49.90_{\pm 5.00}$
CoDE-Stop‡	0	$83.71_{\pm 1.51}$	$59.18_{\pm 2.17}$	$67.24_{\pm 2.79}$	$17.26_{\pm 3.02}$	$53.83_{\pm 4.89}$
DEER‡	0	$60.89_{\pm 1.93}$	$44.21_{\pm 1.98}$	$52.14_{\pm 2.55}$	$13.57_{\pm 2.16}$	$36.35_{\pm 3.97}$
TC	514	$81.22_{\pm 0.94}$	$60.11_{\pm 0.25}$	$66.94_{\pm 0.35}$	$21.75_{\pm 0.07}$	$52.67_{\pm 0.07}$
ROM	5M	$76.41_{\pm 3.49}$	$54.37_{\pm 0.04}$	$60.01_{\pm 1.96}$	$18.69_{\pm 0.88}$	$47.98_{\pm 1.34}$
OSP	101M $\times 5$	$89.11_{\pm 0.42}$	$64.58_{\pm 0.35}$	$73.75_{\pm 0.92}$	$26.90_{\pm 0.81}$	$63.59_{\pm 0.32}$
OSP-1	101M	$88.25_{\pm 0.25}$	$63.86_{\pm 0.49}$	$73.41_{\pm 0.47}$	$27.59_{\pm 2.00}$	$61.86_{\pm 1.48}$
BELT-B-1	16M	$81.70_{\pm 2.94}$	$63.39_{\pm 0.77}$	$68.84_{\pm 2.66}$	$27.15_{\pm 1.57}$	$60.96_{\pm 2.02}$
BELT-H-1	16M	$86.71_{\pm 1.04}$	$65.28_{\pm 0.21}$	$73.12_{\pm 0.75}$	$28.75_{\pm 0.40}$	$63.90_{\pm 0.40}$
BELT-B	16M	$88.45_{\pm 0.68}$	$64.56_{\pm 0.36}$	$73.83_{\pm 0.37}$	$28.72_{\pm 0.42}$	$62.84_{\pm 0.87}$
BELT-H	16M	$88.88_{\pm 0.30}$	$65.16_{\pm 0.24}$	$74.89_{\pm 0.54}$	$27.13_{\pm 0.26}$	$63.53_{\pm 0.27}$

Table 22: Acc. controlling the token budget, reported by dataset. Reporting convention follows Table 2. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 40% token saving, ID and pooled averages.

Method	Probe Params	DAPO	Numina	DSR	Micro	Macro
Natural†‡	–	$52.91_{\pm 1.51}$	$37.74_{\pm 1.94}$	$55.90_{\pm 1.86}$	$55.63_{\pm 0.77}$	$56.18_{\pm 0.97}$
Fixed cap‡	0	$44.28_{\pm 1.47}$	$36.15_{\pm 1.86}$	$48.15_{\pm 1.81}$	$49.41_{\pm 0.76}$	$49.11_{\pm 0.97}$
CoDE-Stop‡	0	$50.54_{\pm 1.44}$	$39.66_{\pm 1.90}$	$52.10_{\pm 1.81}$	$54.66_{\pm 0.74}$	$54.55_{\pm 0.92}$
DEER‡	0	$46.35_{\pm 1.40}$	$36.96_{\pm 1.81}$	$45.44_{\pm 1.77}$	$49.03_{\pm 0.74}$	$48.66_{\pm 0.87}$
TC	514	$50.31_{\pm 0.19}$	$41.36_{\pm 0.02}$	$53.37_{\pm 0.37}$	$55.47_{\pm 0.25}$	$55.80_{\pm 0.19}$
ROM	5M	$47.33_{\pm 0.38}$	$39.06_{\pm 1.49}$	$49.54_{\pm 0.64}$	$52.12_{\pm 0.88}$	$52.30_{\pm 1.03}$
OSP	101M ×5	$\underline{56.93}_{\pm 0.17}$	$\mathbf{44.88}_{\pm 0.38}$	$58.86_{\pm 0.42}$	$\underline{60.61}_{\pm 0.16}$	$\underline{60.97}_{\pm 0.15}$
OSP-1	101M	$55.95_{\pm 1.06}$	$43.26_{\pm 1.60}$	$56.72_{\pm 0.40}$	$59.15_{\pm 0.60}$	$59.45_{\pm 0.65}$
BELT-B-1	16M	$56.08_{\pm 1.29}$	$43.65_{\pm 0.10}$	$58.73_{\pm 0.26}$	$59.57_{\pm 0.57}$	$59.85_{\pm 0.62}$
BELT-H-1	16M	$\mathbf{57.19}_{\pm 0.32}$	$\underline{44.43}_{\pm 0.28}$	$\underline{59.44}_{\pm 0.50}$	$\mathbf{60.81}_{\pm 0.30}$	$\mathbf{61.23}_{\pm 0.27}$
BELT-B	16M	$56.60_{\pm 0.45}$	$43.46_{\pm 0.29}$	$58.97_{\pm 0.16}$	$60.15_{\pm 0.12}$	$60.58_{\pm 0.21}$
BELT-H	16M	$56.78_{\pm 0.16}$	$43.85_{\pm 0.11}$	$\mathbf{59.48}_{\pm 0.09}$	$60.54_{\pm 0.07}$	$60.94_{\pm 0.13}$

Table 23: Acc. controlling the token budget, reported by dataset. Reporting convention follows Table 2. Micro and Macro aggregate ID and OOD. DAPO, Numina, DSR, M500, Olymp., and GK denote DAPO-Math, NuminaMath, DeepScaleR, MATH-500, OlympiadBench, and Gaokao2023-Math-En, respectively. Panel: 40% token saving, OOD.

Method	Probe Params	M500	Olymp.	GK	AIME	AMC
Natural†‡	–	$79.15_{\pm 1.80}$	$65.57_{\pm 2.19}$	$64.53_{\pm 2.92}$	$29.46_{\pm 3.71}$	$64.17_{\pm 4.47}$
Fixed cap‡	0	$75.95_{\pm 1.90}$	$61.05_{\pm 2.19}$	$57.41_{\pm 3.01}$	$15.49_{\pm 3.01}$	$54.38_{\pm 4.98}$
CoDE-Stop‡	0	$85.72_{\pm 1.37}$	$62.33_{\pm 2.13}$	$67.93_{\pm 2.65}$	$20.58_{\pm 3.15}$	$57.57_{\pm 4.62}$
DEER‡	0	$75.17_{\pm 1.77}$	$54.08_{\pm 2.11}$	$63.98_{\pm 2.65}$	$19.31_{\pm 2.73}$	$47.97_{\pm 4.27}$
TC	514	$84.20_{\pm 1.10}$	$63.60_{\pm 0.12}$	$70.04_{\pm 0.05}$	$26.15_{\pm 0.42}$	$57.38_{\pm 0.07}$
ROM	5M	$81.04_{\pm 1.49}$	$59.47_{\pm 0.49}$	$64.97_{\pm 1.31}$	$23.35_{\pm 1.91}$	$53.67_{\pm 0.52}$
OSP	101M ×5	$\mathbf{90.06}_{\pm 0.34}$	$66.55_{\pm 0.39}$	$75.39_{\pm 0.28}$	$29.52_{\pm 0.75}$	$\underline{65.60}_{\pm 0.07}$
OSP-1	101M	$88.65_{\pm 0.19}$	$65.12_{\pm 0.47}$	$74.05_{\pm 1.07}$	$28.85_{\pm 1.48}$	$62.96_{\pm 0.81}$
BELT-B-1	16M	$86.46_{\pm 1.29}$	$66.23_{\pm 0.13}$	$73.86_{\pm 1.56}$	$29.75_{\pm 0.58}$	$64.06_{\pm 0.89}$
BELT-H-1	16M	$89.03_{\pm 0.75}$	$\mathbf{67.40}_{\pm 0.19}$	$\mathbf{76.25}_{\pm 0.30}$	$\mathbf{30.42}_{\pm 0.12}$	$\mathbf{65.70}_{\pm 0.51}$
BELT-B	16M	$89.07_{\pm 0.80}$	$66.20_{\pm 0.42}$	$75.47_{\pm 0.59}$	$\underline{30.26}_{\pm 0.46}$	$64.64_{\pm 1.06}$
BELT-H	16M	$\underline{89.41}_{\pm 0.49}$	$\underline{67.06}_{\pm 0.03}$	$\underline{75.99}_{\pm 0.98}$	$29.59_{\pm 0.15}$	$65.37_{\pm 0.22}$

B.2 OLMo3 MICRO- AND MACRO-AVERAGE RESULTS

Tables 24 to 27 retain both aggregation schemes for the OLMo3 comparison.

Table 24: Token saving and accuracy on OLMo3, reported as Micro and Macro averages. Reporting convention follows Table 3. Panel: token saving, micro averages.

Method	Probe Params	0 pp	1 pp	2 pp
CoDE-Stop‡	0	$0.00_{\pm 0.00}$	$8.43_{\pm 0.27}$	$16.86_{\pm 0.53}$
DEER‡	0	$0.00_{\pm 0.00}$	$1.71_{\pm 0.29}$	$10.98_{\pm 0.64}$
TC	514	$25.26_{\pm 0.33}$	$29.65_{\pm 0.71}$	$33.79_{\pm 0.43}$
ROM	8M	$28.39_{\pm 3.89}$	$34.61_{\pm 1.15}$	$36.90_{\pm 0.98}$
OSP-1	405M	$42.47_{\pm 1.66}$	$50.62_{\pm 2.05}$	$55.18_{\pm 1.92}$
BELT-B-1	19M	$54.49_{\pm 1.82}$	$56.93_{\pm 2.03}$	$59.02_{\pm 2.57}$
BELT-H-1	19M	$54.08_{\pm 1.06}$	$56.20_{\pm 0.86}$	$57.93_{\pm 1.00}$
BELT-B	19M	$\mathbf{56.81}_{\pm 1.33}$	$\mathbf{59.89}_{\pm 1.12}$	$\mathbf{62.14}_{\pm 1.69}$
BELT-H	19M	$\underline{55.21}_{\pm 0.57}$	$\underline{59.04}_{\pm 0.90}$	$\underline{61.96}_{\pm 1.67}$

Table 25: Token saving and accuracy on OLMo3, reported as Micro and Macro averages. Reporting convention follows Table 3. Panel: token saving, macro averages.

Method	Probe Params	0 pp	1 pp	2 pp
CoDE-Stop‡	0	$0.00_{\pm 0.00}$	$8.64_{\pm 0.31}$	$17.27_{\pm 0.63}$
DEER‡	0	$0.00_{\pm 0.00}$	$1.57_{\pm 0.25}$	$10.80_{\pm 0.68}$
TC	514	$25.23_{\pm 0.31}$	$29.56_{\pm 0.47}$	$33.85_{\pm 0.06}$
ROM	8M	$29.98_{\pm 3.90}$	$36.06_{\pm 1.16}$	$38.24_{\pm 1.04}$
OSP-1	405M	$41.53_{\pm 1.58}$	$49.14_{\pm 1.72}$	$53.55_{\pm 1.63}$
BELT-B-1	19M	$52.21_{\pm 1.68}$	$54.61_{\pm 2.02}$	$56.69_{\pm 2.64}$
BELT-H-1	19M	$52.44_{\pm 1.12}$	$54.46_{\pm 0.72}$	$56.14_{\pm 0.77}$
BELT-B	19M	$\mathbf{54.28}_{\pm 1.57}$	$\mathbf{57.32}_{\pm 1.36}$	$\underline{59.67}_{\pm 1.98}$
BELT-H	19M	$\underline{53.34}_{\pm 0.58}$	$\underline{57.22}_{\pm 0.81}$	$\mathbf{59.91}_{\pm 1.78}$

Table 26: Token saving and accuracy on OLMo3, reported as Micro and Macro averages. Reporting convention follows Table 3. Panel: accuracy, micro averages.

Method	Probe Params	70%	60%	50%	40%
Natural†‡	–	81.83	81.83	81.83	81.83
Fixed cap‡	0	$61.33_{\pm 1.22}$	$67.74_{\pm 1.19}$	$72.42_{\pm 1.14}$	$76.36_{\pm 1.07}$
CoDE-Stop‡	0	$56.70_{\pm 1.22}$	$63.43_{\pm 1.17}$	$69.07_{\pm 1.15}$	$73.23_{\pm 1.09}$
DEER‡	0	$40.26_{\pm 1.05}$	$49.05_{\pm 1.00}$	$57.83_{\pm 1.05}$	$65.36_{\pm 1.04}$
TC	514	$60.67_{\pm 0.19}$	$69.24_{\pm 0.11}$	$74.05_{\pm 0.08}$	$78.30_{\pm 0.08}$
ROM	8M	$34.13_{\pm 1.03}$	$45.03_{\pm 0.80}$	$62.60_{\pm 1.10}$	$77.87_{\pm 0.69}$
OSP-1	405M	$74.06_{\pm 0.98}$	$78.65_{\pm 0.61}$	$80.89_{\pm 0.36}$	$82.06_{\pm 0.09}$
BELT-B-1	19M	$70.71_{\pm 2.47}$	$79.13_{\pm 1.27}$	$83.21_{\pm 0.44}$	$\underline{84.95}_{\pm 0.15}$
BELT-H-1	19M	$66.55_{\pm 0.76}$	$78.40_{\pm 0.68}$	$83.09_{\pm 0.36}$	$84.72_{\pm 0.32}$
BELT-B	19M	$\underline{74.88}_{\pm 1.45}$	$\mathbf{80.68}_{\pm 0.65}$	$\mathbf{83.58}_{\pm 0.37}$	$\mathbf{85.10}_{\pm 0.29}$
BELT-H	19M	$\mathbf{75.68}_{\pm 0.94}$	$\underline{80.38}_{\pm 0.47}$	$\underline{83.36}_{\pm 0.27}$	$84.91_{\pm 0.19}$

Table 27: Token saving and accuracy on OLMo3, reported as Micro and Macro averages. Reporting convention follows Table 3. Panel: accuracy, macro averages.

Method	Probe Params	70%	60%	50%	40%
Natural†‡	–	78.35	78.35	78.35	78.35
Fixed cap‡	0	$53.60_{\pm 1.46}$	$59.98_{\pm 1.51}$	$65.55_{\pm 1.47}$	$70.39_{\pm 1.42}$
CoDE-Stop‡	0	$48.94_{\pm 1.40}$	$55.45_{\pm 1.44}$	$61.06_{\pm 1.46}$	$65.95_{\pm 1.40}$
DEER‡	0	$35.63_{\pm 1.15}$	$44.17_{\pm 1.12}$	$52.71_{\pm 1.22}$	$60.32_{\pm 1.25}$
TC	514	$54.02_{\pm 0.21}$	$63.04_{\pm 0.19}$	$68.43_{\pm 0.10}$	$74.27_{\pm 0.11}$
ROM	8M	$32.17_{\pm 1.22}$	$43.08_{\pm 0.37}$	$59.79_{\pm 0.49}$	$74.76_{\pm 0.56}$
OSP-1	405M	$69.74_{\pm 0.63}$	$75.31_{\pm 0.77}$	$78.00_{\pm 0.42}$	$79.29_{\pm 0.09}$
BELT-B-1	19M	$66.73_{\pm 2.58}$	$75.90_{\pm 1.44}$	$\underline{80.31}_{\pm 0.61}$	$\underline{82.07}_{\pm 0.17}$
BELT-H-1	19M	$62.28_{\pm 0.97}$	$74.84_{\pm 1.05}$	$79.91_{\pm 0.57}$	$81.66_{\pm 0.42}$
BELT-B	19M	$\underline{70.74}_{\pm 1.67}$	$\mathbf{77.52}_{\pm 0.77}$	$\mathbf{80.78}_{\pm 0.44}$	$\mathbf{82.22}_{\pm 0.30}$
BELT-H	19M	$\mathbf{71.24}_{\pm 1.31}$	$\underline{76.85}_{\pm 0.43}$	$80.04_{\pm 0.44}$	$81.91_{\pm 0.23}$

B.3 LiveCodeBench Post results

LiveCodeBench Post contains 416 problems, with natural accuracy 28.67%. Most trajectories reach the generation cap. Model training and transfer follow § A.10. OSP has the highest accuracy at the 70% and 60% token-saving targets, but BELT remains competitive across the evaluated trade-offs: both interfaces exceed OSP at 40% saving and save more than OSP-1 at the one-point tolerance (Tables 28 and 29 and Fig. 4). These operating points reuse one 16M BELT probe, compared with five 101M checkpoints for full OSP (§ D). Refitting the policy across prices also improves token saving over the corresponding BELT-B-1/H-1 families. Zero-tolerance saving varies across independent training runs because some frontiers reach natural accuracy only near natural cost.

Table 28: Token saving given the accuracy tolerance and accuracy given the token budget on LiveCodeBench Post (from August 2024). Reporting convention follows Table 3. Panel: token saving.

Method	Probe Params	0 pp	1 pp	2 pp
Natural†	–	–	–	–
OSP	101M ×5	$19.49_{\pm 33.75}$	$\mathbf{65.03}_{\pm 1.75}$	$\mathbf{69.99}_{\pm 0.82}$
OSP-1	101M	$33.48_{\pm 29.12}$	$55.93_{\pm 8.76}$	$65.92_{\pm 5.72}$
BELT-B	16M	$\mathbf{51.66}_{\pm 3.40}$	$\underline{61.90}_{\pm 0.96}$	$\underline{66.56}_{\pm 1.85}$
BELT-H	16M	$\underline{49.75}_{\pm 4.97}$	$59.60_{\pm 1.17}$	$66.00_{\pm 0.87}$
BELT-B-1	16M	$44.32_{\pm 8.28}$	$52.13_{\pm 5.32}$	$58.47_{\pm 3.78}$
BELT-H-1	16M	$45.80_{\pm 7.86}$	$54.97_{\pm 6.33}$	$62.68_{\pm 5.00}$

Table 29: Token saving given the accuracy tolerance and accuracy given the token budget on LiveCodeBench Post (from August 2024). Reporting convention follows Table 3. Panel: accuracy.

Method	Probe Params	70%	60%	50%	40%
Natural†	–	28.67	28.67	28.67	28.67
OSP	101M ×5	**26.70**$_{\pm 0.13}$	**28.21**$_{\pm 0.38}$	28.83$_{\pm 1.06}$	28.89$_{\pm 1.00}$
OSP-1	101M	26.01$_{\pm 1.03}$	27.34$_{\pm 0.65}$	28.25$_{\pm 0.71}$	28.71$_{\pm 0.64}$
BELT-B	16M	25.85$_{\pm 0.50}$	27.97$_{\pm 0.13}$	**28.87**$_{\pm 0.35}$	**29.76**$_{\pm 0.60}$
BELT-H	16M	25.99$_{\pm 0.19}$	27.62$_{\pm 0.23}$	28.77$_{\pm 0.27}$	29.25$_{\pm 0.57}$
BELT-B-1	16M	24.35$_{\pm 0.88}$	26.44$_{\pm 0.78}$	28.10$_{\pm 0.98}$	29.50$_{\pm 1.32}$
BELT-H-1	16M	25.72$_{\pm 0.40}$	27.16$_{\pm 0.77}$	28.25$_{\pm 0.79}$	29.10$_{\pm 0.54}$

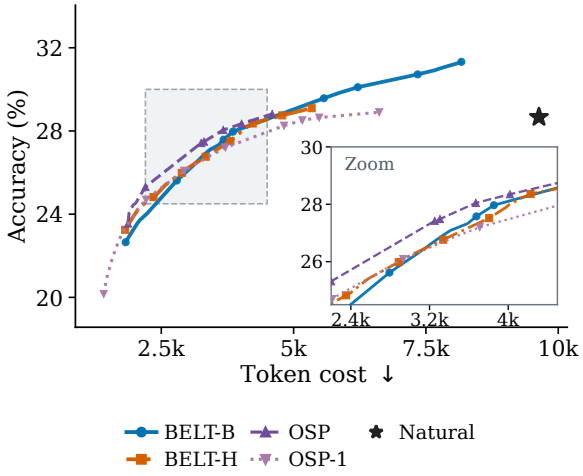

Figure 4: LiveCodeBench Post frontiers. The presentation follows Figure 2(b).

B.4 Test-family-max utility

At $\lambda = 2 \times 10^{-5}$ on Qwen3 OOD, micro average test-family-max utility is 0.654/0.656 for BELT-B/H, 0.656 for OSP, 0.643 for OSP-1, and 0.557 for natural completion (Table 32). These values maximize test utility within each family as defined in § A.7; uncertainty follows § A.6.

On LiveCodeBench Post, OSP leads in test-family-max utility from $\lambda = 10^{-5}$ upward, but BELT-B leads at the two lowest Post prices and the three lowest Pre prices (Tables 34 to 35). BELT thus offers competitive trade-offs across the two splits with one shared 16M probe, compared with five 101M OSP checkpoints. These family maxima describe attainable test trade-offs. The next subsection evaluates policies selected before observing test outcomes.

On OLMo3, both BELT interfaces exceed OSP-1 in mean test-family-max utility at every reported price (Table 36).

Table 30: Utility at fixed token prices on Qwen3, reported as Micro and Macro averages. Reporting convention follows Table 1; each policy family is evaluated at its highest micro average test utility for each price. Panel: ID, micro averages. Token prices λ are in 10^{-6} units.

Method	Probe Params	2	5	10	20	50
Natural†‡	–	$0.4821_{\pm 0.0102}$	$0.4624_{\pm 0.0103}$	$0.4296_{\pm 0.0106}$	$0.3640_{\pm 0.0111}$	$0.1670_{\pm 0.0129}$
Fixed cap‡	0	$0.4821_{\pm 0.0102}$	$0.4624_{\pm 0.0103}$	$0.4296_{\pm 0.0106}$	$0.3775_{\pm 0.0104}$	$0.2358_{\pm 0.0108}$
CoDE-Stop‡	0	$0.5070_{\pm 0.0100}$	$0.4909_{\pm 0.0102}$	$0.4640_{\pm 0.0104}$	$0.4103_{\pm 0.0109}$	$0.2827_{\pm 0.0119}$
DEER‡	0	$0.5087_{\pm 0.0100}$	$0.4924_{\pm 0.0102}$	$0.4662_{\pm 0.0104}$	$0.4142_{\pm 0.0110}$	$0.2602_{\pm 0.0127}$
TC	514	$0.5166_{\pm 0.0001}$	$0.5007_{\pm 0.0000}$	$0.4746_{\pm 0.0004}$	$0.4237_{\pm 0.0005}$	$0.2923_{\pm 0.0014}$
ROM	5M	$0.5061_{\pm 0.0015}$	$0.4895_{\pm 0.0019}$	$0.4634_{\pm 0.0018}$	$0.4125_{\pm 0.0009}$	$0.2667_{\pm 0.0034}$
OSP	101M $\times 5$	$0.5436_{\pm 0.0057}$	$\underline{0.5305}_{\pm 0.0051}$	$\mathbf{0.5088}_{\pm 0.0039}$	$\mathbf{0.4653}_{\pm 0.0016}$	$\underline{0.3630}_{\pm 0.0058}$
OSP-1	101M	$0.5227_{\pm 0.0076}$	$0.5092_{\pm 0.0089}$	$0.4880_{\pm 0.0086}$	$0.4495_{\pm 0.0098}$	$0.3563_{\pm 0.0086}$
BELT-B-1	16M	$\underline{0.5450}_{\pm 0.0031}$	$0.5298_{\pm 0.0028}$	$0.5056_{\pm 0.0030}$	$0.4591_{\pm 0.0030}$	$0.3421_{\pm 0.0099}$
BELT-H-1	16M	$0.5427_{\pm 0.0027}$	$0.5285_{\pm 0.0025}$	$0.5061_{\pm 0.0027}$	$\underline{0.4646}_{\pm 0.0033}$	$0.3627_{\pm 0.0010}$
BELT-B	16M	$\mathbf{0.5472}_{\pm 0.0029}$	$\mathbf{0.5321}_{\pm 0.0032}$	$\underline{0.5073}_{\pm 0.0034}$	$0.4609_{\pm 0.0023}$	$0.3606_{\pm 0.0028}$
BELT-H	16M	$0.5420_{\pm 0.0013}$	$0.5273_{\pm 0.0013}$	$0.5034_{\pm 0.0013}$	$0.4610_{\pm 0.0012}$	$\mathbf{0.3663}_{\pm 0.0018}$

Table 31: Utility at fixed token prices on Qwen3, reported as Micro and Macro averages. Reporting convention follows Table 1; each policy family is evaluated at its highest micro average test utility for each price. Panel: ID, macro averages. Token prices λ are in 10^{-6} units.

Method	Probe Params	2	5	10	20	50
Natural†‡	–	$0.4757_{\pm 0.0104}$	$0.4564_{\pm 0.0105}$	$0.4243_{\pm 0.0107}$	$0.3602_{\pm 0.0112}$	$0.1677_{\pm 0.0130}$
Fixed cap‡	0	$0.4757_{\pm 0.0104}$	$0.4564_{\pm 0.0105}$	$0.4243_{\pm 0.0107}$	$0.3738_{\pm 0.0106}$	$0.2348_{\pm 0.0110}$
CoDE-Stop‡	0	$0.5022_{\pm 0.0103}$	$0.4864_{\pm 0.0104}$	$0.4601_{\pm 0.0107}$	$0.4075_{\pm 0.0112}$	$0.2828_{\pm 0.0121}$
DEER‡	0	$0.5027_{\pm 0.0103}$	$0.4868_{\pm 0.0105}$	$0.4609_{\pm 0.0107}$	$0.4103_{\pm 0.0112}$	$0.2612_{\pm 0.0128}$
TC	514	$0.5110_{\pm 0.0004}$	$0.4955_{\pm 0.0003}$	$0.4703_{\pm 0.0002}$	$0.4208_{\pm 0.0003}$	$0.2950_{\pm 0.0011}$
ROM	5M	$0.4989_{\pm 0.0014}$	$0.4829_{\pm 0.0021}$	$0.4582_{\pm 0.0009}$	$0.4092_{\pm 0.0006}$	$0.2688_{\pm 0.0051}$
OSP	101M $\times 5$	$0.5378_{\pm 0.0058}$	$\underline{0.5251}_{\pm 0.0051}$	$\mathbf{0.5041}_{\pm 0.0040}$	$\mathbf{0.4619}_{\pm 0.0016}$	$\underline{0.3639}_{\pm 0.0062}$
OSP-1	101M	$0.5159_{\pm 0.0077}$	$0.5028_{\pm 0.0090}$	$0.4828_{\pm 0.0082}$	$0.4462_{\pm 0.0098}$	$0.3569_{\pm 0.0085}$
BELT-B-1	16M	$\underline{0.5384}_{\pm 0.0031}$	$0.5242_{\pm 0.0028}$	$0.5007_{\pm 0.0029}$	$0.4562_{\pm 0.0024}$	$0.3433_{\pm 0.0098}$
BELT-H-1	16M	$0.5365_{\pm 0.0028}$	$0.5230_{\pm 0.0026}$	$0.5014_{\pm 0.0028}$	$\underline{0.4613}_{\pm 0.0033}$	$0.3628_{\pm 0.0009}$
BELT-B	16M	$\mathbf{0.5409}_{\pm 0.0031}$	$\mathbf{0.5262}_{\pm 0.0034}$	$\underline{0.5023}_{\pm 0.0034}$	$0.4575_{\pm 0.0020}$	$0.3612_{\pm 0.0031}$
BELT-H	16M	$0.5356_{\pm 0.0013}$	$0.5214_{\pm 0.0012}$	$0.4984_{\pm 0.0013}$	$0.4575_{\pm 0.0013}$	$\mathbf{0.3664}_{\pm 0.0016}$

Table 32: Utility at fixed token prices on Qwen3, reported as Micro and Macro averages. Reporting convention follows Table 1; each policy family is evaluated at its highest micro average test utility for each price. Panel: OOD, micro averages. Token prices λ are in 10^{-6} units.

Method	Probe Params	2	5	10	20	50
Natural†‡	–	$0.6473_{\pm 0.0120}$	$0.6322_{\pm 0.0122}$	$0.6071_{\pm 0.0126}$	$0.5569_{\pm 0.0133}$	$0.4062_{\pm 0.0158}$
Fixed cap‡	0	$0.6473_{\pm 0.0120}$	$0.6322_{\pm 0.0122}$	$0.6071_{\pm 0.0126}$	$0.5633_{\pm 0.0127}$	$0.4516_{\pm 0.0140}$
CoDE-Stop‡	0	$0.6859_{\pm 0.0111}$	$0.6740_{\pm 0.0114}$	$0.6542_{\pm 0.0117}$	$0.6147_{\pm 0.0124}$	$0.5091_{\pm 0.0138}$
DEER‡	0	$0.6870_{\pm 0.0110}$	$0.6750_{\pm 0.0112}$	$0.6550_{\pm 0.0116}$	$0.6150_{\pm 0.0125}$	$0.4973_{\pm 0.0149}$
TC	514	$0.6859_{\pm 0.0013}$	$0.6738_{\pm 0.0019}$	$0.6548_{\pm 0.0011}$	$0.6176_{\pm 0.0005}$	$0.5168_{\pm 0.0032}$
ROM	5M	$0.6666_{\pm 0.0008}$	$0.6545_{\pm 0.0010}$	$0.6343_{\pm 0.0010}$	$0.5954_{\pm 0.0006}$	$0.4890_{\pm 0.0034}$
OSP	101M ×5	$0.7146_{\pm 0.0078}$	$0.7047_{\pm 0.0071}$	$0.6884_{\pm 0.0061}$	$\underline{0.6559}_{\pm 0.0045}$	$\underline{0.5755}_{\pm 0.0049}$
OSP-1	101M	$0.6968_{\pm 0.0068}$	$0.6873_{\pm 0.0062}$	$0.6719_{\pm 0.0056}$	$0.6427_{\pm 0.0044}$	$0.5706_{\pm 0.0032}$
BELT-B-1	16M	$\underline{0.7211}_{\pm 0.0014}$	$\underline{0.7087}_{\pm 0.0013}$	$0.6887_{\pm 0.0007}$	$0.6512_{\pm 0.0012}$	$0.5516_{\pm 0.0069}$
BELT-H-1	16M	$0.7195_{\pm 0.0024}$	$0.7081_{\pm 0.0020}$	$\underline{0.6903}_{\pm 0.0017}$	$\mathbf{0.6575}_{\pm 0.0032}$	$0.5693_{\pm 0.0022}$
BELT-B	16M	$\mathbf{0.7232}_{\pm 0.0004}$	$\mathbf{0.7110}_{\pm 0.0004}$	$\mathbf{0.6912}_{\pm 0.0015}$	$0.6536_{\pm 0.0023}$	$0.5752_{\pm 0.0044}$
BELT-H	16M	$0.7167_{\pm 0.0035}$	$0.7055_{\pm 0.0030}$	$0.6877_{\pm 0.0015}$	$0.6555_{\pm 0.0008}$	$\mathbf{0.5815}_{\pm 0.0025}$

Table 33: Utility at fixed token prices on Qwen3, reported as Micro and Macro averages. Reporting convention follows Table 1; each policy family is evaluated at its highest micro average test utility for each price. Panel: OOD, macro averages. Token prices λ are in 10^{-6} units.

Method	Probe Params	2	5	10	20	50
Natural†‡	–	$0.5943_{\pm 0.0144}$	$0.5772_{\pm 0.0147}$	$0.5486_{\pm 0.0152}$	$0.4913_{\pm 0.0163}$	$0.3197_{\pm 0.0196}$
Fixed cap‡	0	$0.5943_{\pm 0.0144}$	$0.5772_{\pm 0.0147}$	$0.5486_{\pm 0.0152}$	$0.4927_{\pm 0.0153}$	$0.3727_{\pm 0.0169}$
CoDE-Stop‡	0	$0.6221_{\pm 0.0137}$	$0.6086_{\pm 0.0140}$	$0.5860_{\pm 0.0145}$	$0.5407_{\pm 0.0154}$	$0.4215_{\pm 0.0171}$
DEER‡	0	$0.6249_{\pm 0.0137}$	$0.6106_{\pm 0.0140}$	$0.5867_{\pm 0.0146}$	$0.5391_{\pm 0.0158}$	$0.3987_{\pm 0.0191}$
TC	514	$0.6274_{\pm 0.0006}$	$0.6113_{\pm 0.0028}$	$0.5892_{\pm 0.0018}$	$0.5451_{\pm 0.0001}$	$0.4265_{\pm 0.0033}$
ROM	5M	$0.6074_{\pm 0.0011}$	$0.5931_{\pm 0.0010}$	$0.5688_{\pm 0.0000}$	$0.5213_{\pm 0.0000}$	$0.3925_{\pm 0.0013}$
OSP	101M ×5	$0.6547_{\pm 0.0067}$	$0.6433_{\pm 0.0059}$	$0.6241_{\pm 0.0047}$	$\underline{0.5869}_{\pm 0.0027}$	$\underline{0.4996}_{\pm 0.0047}$
OSP-1	101M	$0.6368_{\pm 0.0068}$	$0.6256_{\pm 0.0060}$	$0.6067_{\pm 0.0066}$	$0.5725_{\pm 0.0060}$	$0.4967_{\pm 0.0035}$
BELT-B-1	16M	$\underline{0.6614}_{\pm 0.0017}$	$\underline{0.6471}_{\pm 0.0013}$	$0.6236_{\pm 0.0007}$	$0.5793_{\pm 0.0009}$	$0.4635_{\pm 0.0078}$
BELT-H-1	16M	$0.6593_{\pm 0.0023}$	$0.6462_{\pm 0.0018}$	$\underline{0.6251}_{\pm 0.0022}$	$\mathbf{0.5871}_{\pm 0.0036}$	$0.4879_{\pm 0.0006}$
BELT-B	16M	$\mathbf{0.6631}_{\pm 0.0003}$	$\mathbf{0.6487}_{\pm 0.0007}$	$\mathbf{0.6260}_{\pm 0.0016}$	$0.5815_{\pm 0.0025}$	$0.4953_{\pm 0.0056}$
BELT-H	16M	$0.6570_{\pm 0.0033}$	$0.6437_{\pm 0.0028}$	$0.6230_{\pm 0.0011}$	$0.5862_{\pm 0.0010}$	$\mathbf{0.5080}_{\pm 0.0030}$

Table 34: Utility at fixed token prices on LiveCodeBench Pre and Post. Reporting convention follows Table 3; the highest test utility is reported for each policy family and price. Panel: Pre. Token prices λ are in 10^{-6} units.

Method	Probe Params	2	5	10	20	50
Natural†	–	0.6118	0.5915	0.5579	0.4905	0.2884
OSP	101M ×5	$0.6144_{\pm 0.0045}$	$0.5967_{\pm 0.0087}$	$0.5722_{\pm 0.0138}$	$0.5327_{\pm 0.0121}$	$0.4367_{\pm 0.0091}$
OSP-1	101M	$0.6121_{\pm 0.0006}$	$0.5924_{\pm 0.0008}$	$0.5609_{\pm 0.0028}$	$0.5098_{\pm 0.0118}$	$0.4139_{\pm 0.0096}$
BELT-B-1	16M	$\underline{0.6420}_{\pm 0.0168}$	$\underline{0.6258}_{\pm 0.0160}$	$\underline{0.5988}_{\pm 0.0147}$	$\mathbf{0.5458}_{\pm 0.0132}$	$0.4288_{\pm 0.0105}$
BELT-H-1	16M	$0.6290_{\pm 0.0018}$	$0.6118_{\pm 0.0018}$	$0.5849_{\pm 0.0031}$	$0.5360_{\pm 0.0032}$	$0.4312_{\pm 0.0018}$
BELT-B	16M	$\mathbf{0.6482}_{\pm 0.0166}$	$\mathbf{0.6303}_{\pm 0.0168}$	$\mathbf{0.6004}_{\pm 0.0171}$	$\underline{0.5455}_{\pm 0.0109}$	$\underline{0.4441}_{\pm 0.0051}$
BELT-H	16M	$0.6237_{\pm 0.0079}$	$0.6096_{\pm 0.0072}$	$0.5861_{\pm 0.0059}$	$0.5404_{\pm 0.0038}$	$\mathbf{0.4445}_{\pm 0.0050}$

Table 35: Utility at fixed token prices on LiveCodeBench Pre and Post. Reporting convention follows Table 3; the highest test utility is reported for each policy family and price. Panel: Post. Token prices λ are in 10^{-6} units.

Method	Probe Params	2	5	10	20	50
Natural†	–	0.2674	0.2385	0.1903	0.0940	−0.1951
OSP	101M ×5	$0.2796_{\pm 0.0104}$	$0.2680_{\pm 0.0083}$	$\mathbf{0.2488}_{\pm 0.0049}$	$\mathbf{0.2124}_{\pm 0.0014}$	$\mathbf{0.1496}_{\pm 0.0049}$
OSP-1	101M	$0.2762_{\pm 0.0067}$	$0.2598_{\pm 0.0068}$	$0.2369_{\pm 0.0065}$	$0.2033_{\pm 0.0096}$	$\underline{0.1435}_{\pm 0.0063}$
BELT-B-1	16M	$\underline{0.2925}_{\pm 0.0119}$	$\underline{0.2710}_{\pm 0.0118}$	$0.2393_{\pm 0.0122}$	$0.1893_{\pm 0.0077}$	$0.1329_{\pm 0.0015}$
BELT-H-1	16M	$0.2843_{\pm 0.0052}$	$0.2649_{\pm 0.0051}$	$0.2361_{\pm 0.0063}$	$0.1999_{\pm 0.0037}$	$0.1375_{\pm 0.0023}$
BELT-B	16M	$\mathbf{0.2972}_{\pm 0.0118}$	$\mathbf{0.2738}_{\pm 0.0100}$	$\underline{0.2431}_{\pm 0.0035}$	$\underline{0.2046}_{\pm 0.0019}$	$0.1368_{\pm 0.0015}$
BELT-H	16M	$0.2817_{\pm 0.0060}$	$0.2655_{\pm 0.0043}$	$0.2426_{\pm 0.0015}$	$0.2038_{\pm 0.0006}$	$0.1431_{\pm 0.0022}$

Table 36: Micro average utility at fixed token prices on OLMo3. Reporting convention follows Table 3; the highest test utility is reported for each policy family and price.

Method	Probe Params	OOD Micro: utility ↑ (test-curve reading)				
λ		2×10^{-6}	5×10^{-6}	10^{-5}	2×10^{-5}	5×10^{-5}
Natural†	–	0.8046	0.7840	0.7496	0.6809	0.4749
Fixed cap‡	0	$0.8046_{\pm 0.0103}$	$0.7840_{\pm 0.0107}$	$0.7509_{\pm 0.0106}$	$0.7006_{\pm 0.0110}$	$0.5589_{\pm 0.0122}$
CoDE-Stop‡	0	$0.8046_{\pm 0.0103}$	$0.7840_{\pm 0.0107}$	$0.7496_{\pm 0.0112}$	$0.6862_{\pm 0.0114}$	$0.5365_{\pm 0.0133}$
DEER‡	0	$0.8046_{\pm 0.0103}$	$0.7840_{\pm 0.0107}$	$0.7496_{\pm 0.0112}$	$0.6809_{\pm 0.0123}$	$0.5029_{\pm 0.0152}$
TC	514	$0.8232_{\pm 0.0005}$	$0.8054_{\pm 0.0003}$	$0.7757_{\pm 0.0003}$	$0.7182_{\pm 0.0011}$	$0.5802_{\pm 0.0008}$
ROM	8M	$0.8118_{\pm 0.0014}$	$0.7962_{\pm 0.0027}$	$0.7706_{\pm 0.0044}$	$0.7221_{\pm 0.0049}$	$0.5842_{\pm 0.0043}$
OSP-1	405M	$0.8162_{\pm 0.0036}$	$0.8021_{\pm 0.0020}$	$0.7799_{\pm 0.0009}$	$0.7408_{\pm 0.0030}$	$0.6506_{\pm 0.0064}$
BELT-B-1	19M	$0.8431_{\pm 0.0011}$	$\underline{0.8300}_{\pm 0.0015}$	$\underline{0.8086}_{\pm 0.0016}$	$0.7690_{\pm 0.0016}$	$0.6623_{\pm 0.0063}$
BELT-H-1	19M	$0.8410_{\pm 0.0021}$	$0.8277_{\pm 0.0028}$	$0.8066_{\pm 0.0033}$	$0.7661_{\pm 0.0029}$	$0.6617_{\pm 0.0051}$
BELT-B	19M	$\mathbf{0.8449}_{\pm 0.0027}$	$\mathbf{0.8317}_{\pm 0.0027}$	$\mathbf{0.8102}_{\pm 0.0029}$	$\mathbf{0.7705}_{\pm 0.0020}$	$\mathbf{0.6728}_{\pm 0.0041}$
BELT-H	19M	$\underline{0.8435}_{\pm 0.0019}$	$0.8296_{\pm 0.0016}$	$0.8083_{\pm 0.0021}$	$\underline{0.7691}_{\pm 0.0032}$	$\underline{0.6695}_{\pm 0.0051}$

B.5 Utility of dev-selected policies

On math OOD, BELT-B has higher mean utility than OSP at the three lowest and the highest reported prices, while OSP leads at $\lambda = 2 \times 10^{-5}$ (Table 37). Both BELT interfaces have higher mean utility than OSP at every reported price on LiveCodeBench Pre. On Post, OSP leads at the three intermediate prices, but BELT remains competitive: BELT-B leads at the lowest price and BELT-H at the highest, reusing the same compact probe across prices. Both BELT interfaces exceed OSP-1 across reported prices on Qwen3 OOD and OLMo3. This protocol evaluates transfer at a fixed token price, selecting policies on mathematics development utility before target-population evaluation.

Table 37: Utility of dev-selected policies. Reporting convention follows Table 1; all values are micro averages.

Method	Utility by λ ($\times 10^{-6}$)				
	2	5	10	20	50
Math ID					
OSP	$0.5436_{\pm 0.0057}$	$\mathbf{0.5305}_{\pm 0.0051}$	$\mathbf{0.5088}_{\pm 0.0039}$	$\mathbf{0.4653}_{\pm 0.0016}$	$0.3536_{\pm 0.0113}$
OSP-1	$0.5227_{\pm 0.0076}$	$0.5079_{\pm 0.0084}$	$0.4865_{\pm 0.0109}$	$0.4436_{\pm 0.0174}$	$0.3537_{\pm 0.0130}$
BELT-B	$\mathbf{0.5463}_{\pm 0.0035}$	$0.5303_{\pm 0.0043}$	$0.5048_{\pm 0.0042}$	$0.4603_{\pm 0.0019}$	$\mathbf{0.3603}_{\pm 0.0028}$
BELT-H	$0.5411_{\pm 0.0016}$	$0.5262_{\pm 0.0018}$	$0.5014_{\pm 0.0015}$	$0.4595_{\pm 0.0013}$	$0.3599_{\pm 0.0036}$
Math OOD					
OSP	$0.7146_{\pm 0.0078}$	$0.7047_{\pm 0.0071}$	$0.6884_{\pm 0.0061}$	$\mathbf{0.6557}_{\pm 0.0045}$	$0.5695_{\pm 0.0066}$
OSP-1	$0.6940_{\pm 0.0110}$	$0.6873_{\pm 0.0062}$	$0.6713_{\pm 0.0063}$	$0.6395_{\pm 0.0100}$	$0.5695_{\pm 0.0048}$
BELT-B	$\mathbf{0.7232}_{\pm 0.0004}$	$\mathbf{0.7104}_{\pm 0.0012}$	$\mathbf{0.6904}_{\pm 0.0010}$	$0.6524_{\pm 0.0040}$	$0.5738_{\pm 0.0042}$
BELT-H	$0.7141_{\pm 0.0064}$	$0.7024_{\pm 0.0068}$	$0.6835_{\pm 0.0048}$	$0.6543_{\pm 0.0021}$	$\mathbf{0.5791}_{\pm 0.0037}$
LiveCodeBench Pre					
OSP	$0.5966_{\pm 0.0232}$	$0.5859_{\pm 0.0213}$	$0.5681_{\pm 0.0181}$	$0.5325_{\pm 0.0120}$	$0.4250_{\pm 0.0046}$
OSP-1	$0.6019_{\pm 0.0069}$	$0.5827_{\pm 0.0083}$	$0.5565_{\pm 0.0061}$	$0.5040_{\pm 0.0075}$	$0.4085_{\pm 0.0168}$
BELT-B	$\mathbf{0.6363}_{\pm 0.0211}$	$\mathbf{0.6179}_{\pm 0.0213}$	$\mathbf{0.5866}_{\pm 0.0226}$	$\mathbf{0.5387}_{\pm 0.0155}$	$\mathbf{0.4404}_{\pm 0.0027}$
BELT-H	$0.6144_{\pm 0.0164}$	$0.5982_{\pm 0.0174}$	$0.5718_{\pm 0.0109}$	$0.5366_{\pm 0.0078}$	$0.4390_{\pm 0.0040}$
LiveCodeBench Post					
OSP	$0.2796_{\pm 0.0104}$	$\mathbf{0.2680}_{\pm 0.0083}$	$\mathbf{0.2488}_{\pm 0.0049}$	$\mathbf{0.2104}_{\pm 0.0020}$	$0.1015_{\pm 0.0153}$
OSP-1	$0.2756_{\pm 0.0078}$	$0.2548_{\pm 0.0092}$	$0.2214_{\pm 0.0143}$	$0.1545_{\pm 0.0267}$	$0.1004_{\pm 0.0230}$
BELT-B	$\mathbf{0.2909}_{\pm 0.0162}$	$0.2653_{\pm 0.0166}$	$0.2215_{\pm 0.0189}$	$0.1580_{\pm 0.0243}$	$0.1045_{\pm 0.0064}$
BELT-H	$0.2784_{\pm 0.0047}$	$0.2580_{\pm 0.0039}$	$0.2328_{\pm 0.0067}$	$0.1964_{\pm 0.0082}$	$\mathbf{0.1196}_{\pm 0.0098}$
OLMo3 OOD					
OSP-1	$0.8162_{\pm 0.0036}$	$0.8021_{\pm 0.0020}$	$0.7776_{\pm 0.0043}$	$0.7386_{\pm 0.0025}$	$0.6476_{\pm 0.0063}$
BELT-B	$0.8393_{\pm 0.0076}$	$\mathbf{0.8295}_{\pm 0.0020}$	$\mathbf{0.8080}_{\pm 0.0009}$	$\mathbf{0.7684}_{\pm 0.0053}$	$\mathbf{0.6680}_{\pm 0.0037}$
BELT-H	$\mathbf{0.8407}_{\pm 0.0007}$	$0.8266_{\pm 0.0031}$	$0.8076_{\pm 0.0009}$	$0.7650_{\pm 0.0039}$	$0.6649_{\pm 0.0091}$

B.6 Inference-time threshold sweeps

Threshold expansion adds candidate policies at inference time without retraining weights. For each of OSP's five checkpoints and each of BELT-B/H's 17 policies, we take 13 equally spaced source-validation score quantiles and stop at the first threshold crossing. Retaining the original stochastic policies gives 70 OSP and 238 candidates per BELT interface before deduplication, plus natural completion. Test-frontier constraints and utility selection use micro averages; Qwen3 macro values evaluate the same policies or mixtures. Micro and macro coincide for each single-dataset LiveCodeBench split.

Price-specific fitting and threshold-only adaptation. On Qwen3, BELT-H-1 falls below OSP-1 in OOD token saving at the 1% tolerance, but remains close to full BELT-H in accuracy at 50% saving. Refitting the small policy across prices adds 9.4 pp of saving over BELT-H-1 and places BELT-H 2.2 pp ahead of full OSP at that tolerance (Tables 1 and 2). All BELT operating points reuse one probe, with only 6 or 769 parameters per BELT-B/H policy (§ D.3).

Performance across distributions. Expanded BELT-H saves 64.14%/69.01% at the one-point tolerance on Qwen3 ID/OOD, versus 63.73%/66.89% for expanded OSP, and also has higher macro accuracy at half natural cost (Tables 38 to 39). Among expanded families, OSP leads in macro utility at four of five prices on ID, while BELT-B or BELT-H leads at all five on OOD (Table 42).

On LiveCodeBench Pre, expansion raises OSP's one-point-tolerance saving from 27.8% to 40.11%, below BELT-B's 46.14% (Tables 40 to 41). On Post, expanded OSP leads at that tolerance and at the

three higher utility prices, but BELT-B remains competitive: at half natural cost, its 29.61% accuracy is within 0.12 pp of OSP's 29.73%. Expanded BELT-B exceeds OSP in utility at every price on Pre and the two lowest on Post, while retaining the single-probe training advantage quantified below.

Threshold expansion leaves controller-training costs unchanged: one 15.96M-parameter BELT probe versus five 100.68M-parameter OSP checkpoints. The accounting in § D.2 gives approximately 10.0 versus 254.5 GPU-hours, a 25.5-fold difference.

Table 38: Original and threshold-expanded families on Qwen3 (Macro). Reporting convention follows Table 1. NF denotes an infeasible budget in at least one seed. Panel: token saving.

Method	0 pp	1 pp	2 pp
	ID		
OSP	$60.00_{\pm 1.82}$	$62.34_{\pm 1.06}$	$64.69_{\pm 0.31}$
OSP + thresholds	$61.46_{\pm 0.99}$	$63.73_{\pm 1.43}$	$66.20_{\pm 1.09}$
BELT-B	$59.05_{\pm 1.00}$	$61.76_{\pm 1.31}$	$64.22_{\pm 1.27}$
BELT-B + thresholds	$58.32_{\pm 3.00}$	$61.96_{\pm 0.68}$	$63.54_{\pm 1.37}$
BELT-H	$60.12_{\pm 0.30}$	$63.71_{\pm 0.47}$	$66.33_{\pm 0.61}$
BELT-H + thresholds	$\mathbf{61.65}_{\pm 0.64}$	$\mathbf{64.14}_{\pm 0.78}$	$\mathbf{66.96}_{\pm 0.74}$
	OOD		
OSP	$63.80_{\pm 0.56}$	$66.64_{\pm 0.76}$	$68.27_{\pm 0.71}$
OSP + thresholds	$64.00_{\pm 0.56}$	$66.89_{\pm 0.79}$	$70.05_{\pm 0.16}$
BELT-B	$63.07_{\pm 0.66}$	$65.60_{\pm 0.37}$	$67.58_{\pm 1.35}$
BELT-B + thresholds	$61.16_{\pm 2.88}$	$65.66_{\pm 0.58}$	$67.45_{\pm 1.82}$
BELT-H	$66.18_{\pm 0.95}$	$68.82_{\pm 0.42}$	$\mathbf{71.26}_{\pm 0.11}$
BELT-H + thresholds	$\mathbf{66.66}_{\pm 0.76}$	$\mathbf{69.01}_{\pm 0.99}$	$71.25_{\pm 0.54}$

Table 39: Original and threshold-expanded families on Qwen3 (Macro). Reporting convention follows Table 1. NF denotes an infeasible budget in at least one seed. Panel: accuracy.

Method	70%	60%	50%	40%
		ID		
OSP	$44.58_{\pm 0.52}$	$48.89_{\pm 0.53}$	$51.55_{\pm 0.12}$	$53.56_{\pm 0.24}$
OSP + thresholds	$44.71_{\pm 0.33}$	$49.44_{\pm 0.18}$	$51.90_{\pm 0.32}$	$53.66_{\pm 0.25}$
BELT-B	$44.06_{\pm 0.61}$	$48.58_{\pm 0.50}$	$51.31_{\pm 0.43}$	$53.01_{\pm 0.20}$
BELT-B + thresholds	$44.09_{\pm 0.68}$	$48.64_{\pm 0.25}$	$51.49_{\pm 0.27}$	$53.20_{\pm 0.28}$
BELT-H	$45.64_{\pm 0.27}$	$49.23_{\pm 0.11}$	$51.57_{\pm 0.14}$	$53.37_{\pm 0.06}$
BELT-H + thresholds	$\mathbf{45.79}_{\pm 0.48}$	$\mathbf{49.63}_{\pm 0.06}$	$\mathbf{52.25}_{\pm 0.22}$	$\mathbf{53.82}_{\pm 0.19}$
		OOD		
OSP	NF	$61.26_{\pm 0.51}$	$63.59_{\pm 0.45}$	$65.42_{\pm 0.10}$
OSP + thresholds	$57.96_{\pm 0.10}$	$61.77_{\pm 0.49}$	$64.03_{\pm 0.28}$	$65.73_{\pm 0.46}$
BELT-B	$56.97_{\pm 1.20}$	$61.33_{\pm 0.70}$	$63.68_{\pm 0.27}$	$65.13_{\pm 0.40}$
BELT-B + thresholds	$57.14_{\pm 1.15}$	$61.07_{\pm 1.00}$	$63.76_{\pm 0.37}$	$65.30_{\pm 0.62}$
BELT-H	$\mathbf{58.49}_{\pm 0.16}$	$61.90_{\pm 0.20}$	$63.92_{\pm 0.27}$	$65.48_{\pm 0.23}$
BELT-H + thresholds	$\mathbf{58.49}_{\pm 0.32}$	$\mathbf{62.20}_{\pm 0.09}$	$\mathbf{64.46}_{\pm 0.22}$	$\mathbf{65.84}_{\pm 0.29}$

Table 40: Original and threshold-expanded families on LiveCodeBench. Reporting convention follows Table 1. Panel: token saving.

Method	0 pp	1 pp	2 pp
Pre			
OSP	$12.46_{\pm 21.57}$	$27.75_{\pm 14.52}$	$38.89_{\pm 13.01}$
OSP + thresholds	$36.36_{\pm 2.38}$	$40.11_{\pm 3.25}$	$44.42_{\pm 5.13}$
BELT-B	$37.20_{\pm 5.56}$	$43.33_{\pm 4.31}$	$48.60_{\pm 2.37}$
BELT-B + thresholds	$\mathbf{42.56}_{\pm 1.11}$	$\mathbf{46.14}_{\pm 2.69}$	$\mathbf{51.16}_{\pm 1.50}$
BELT-H	$24.61_{\pm 21.39}$	$42.26_{\pm 1.18}$	$47.29_{\pm 1.00}$
BELT-H + thresholds	$37.06_{\pm 1.19}$	$42.55_{\pm 1.77}$	$50.10_{\pm 2.97}$
Post			
OSP	$19.49_{\pm 33.75}$	$65.03_{\pm 1.75}$	$69.99_{\pm 0.82}$
OSP + thresholds	$\mathbf{57.65}_{\pm 6.01}$	$\mathbf{66.19}_{\pm 1.81}$	$\mathbf{71.11}_{\pm 2.57}$
BELT-B	$51.66_{\pm 3.40}$	$61.90_{\pm 0.96}$	$66.56_{\pm 1.85}$
BELT-B + thresholds	$57.36_{\pm 0.87}$	$63.19_{\pm 1.97}$	$67.21_{\pm 1.29}$
BELT-H	$49.75_{\pm 4.97}$	$59.60_{\pm 1.17}$	$66.00_{\pm 0.87}$
BELT-H + thresholds	$53.65_{\pm 2.57}$	$62.59_{\pm 1.51}$	$68.46_{\pm 0.79}$

Table 41: Original and threshold-expanded families on LiveCodeBench. Reporting convention follows Table 1. Panel: accuracy.

Method	70%	60%	50%	40%
Pre				
OSP	$53.09_{\pm 1.48}$	$56.49_{\pm 0.13}$	$59.10_{\pm 0.97}$	$60.72_{\pm 1.34}$
OSP + thresholds	$53.77_{\pm 0.97}$	$57.10_{\pm 0.70}$	$59.49_{\pm 0.87}$	$61.28_{\pm 1.20}$
BELT-B	$54.19_{\pm 0.52}$	$57.67_{\pm 0.30}$	$60.31_{\pm 0.50}$	$62.16_{\pm 0.68}$
BELT-B + thresholds	$\mathbf{54.23}_{\pm 0.33}$	$\mathbf{57.77}_{\pm 0.19}$	$\mathbf{60.70}_{\pm 0.35}$	$\mathbf{63.09}_{\pm 0.43}$
BELT-H	$54.17_{\pm 0.76}$	$57.24_{\pm 0.59}$	$59.86_{\pm 0.32}$	$61.94_{\pm 0.30}$
BELT-H + thresholds	$54.11_{\pm 0.84}$	$57.71_{\pm 0.54}$	$60.41_{\pm 0.84}$	$62.07_{\pm 0.37}$
Post				
OSP	$26.70_{\pm 0.13}$	$28.21_{\pm 0.38}$	$28.83_{\pm 1.06}$	$28.89_{\pm 1.00}$
OSP + thresholds	$\mathbf{26.78}_{\pm 0.48}$	$\mathbf{28.48}_{\pm 0.46}$	$\mathbf{29.73}_{\pm 0.46}$	$30.24_{\pm 0.22}$
BELT-B	$25.85_{\pm 0.50}$	$27.97_{\pm 0.13}$	$28.87_{\pm 0.35}$	$29.76_{\pm 0.60}$
BELT-B + thresholds	$25.91_{\pm 0.46}$	$28.13_{\pm 0.11}$	$29.61_{\pm 0.12}$	$\mathbf{30.64}_{\pm 0.39}$
BELT-H	$25.99_{\pm 0.19}$	$27.62_{\pm 0.23}$	$28.77_{\pm 0.27}$	$29.25_{\pm 0.57}$
BELT-H + thresholds	$26.19_{\pm 0.18}$	$28.16_{\pm 0.05}$	$29.07_{\pm 0.03}$	$29.90_{\pm 0.08}$

Table 42: Utility of the original and threshold-expanded families. Qwen3 uses Macro averages. Reporting convention follows Table 1.

Method	2×10^{-6}	5×10^{-6}	10^{-5}	2×10^{-5}	5×10^{-5}
Qwen3 ID (Macro)					
OSP	$0.5378_{\pm 0.0058}$	$0.5251_{\pm 0.0051}$	$\underline{0.5041}_{\pm 0.0040}$	$0.4619_{\pm 0.0016}$	$0.3639_{\pm 0.0062}$
OSP + thresholds	$\mathbf{0.5430}_{\pm 0.0040}$	$\mathbf{0.5280}_{\pm 0.0025}$	$\mathbf{0.5050}_{\pm 0.0028}$	$\mathbf{0.4631}_{\pm 0.0030}$	$\underline{0.3702}_{\pm 0.0014}$
BELT-B	$0.5409_{\pm 0.0031}$	$0.5262_{\pm 0.0034}$	$0.5023_{\pm 0.0034}$	$0.4575_{\pm 0.0020}$	$0.3612_{\pm 0.0031}$
BELT-B + thresholds	$\underline{0.5429}_{\pm 0.0042}$	$\underline{0.5274}_{\pm 0.0038}$	$0.5030_{\pm 0.0030}$	$0.4588_{\pm 0.0016}$	$0.3617_{\pm 0.0037}$
BELT-H	$0.5356_{\pm 0.0013}$	$0.5214_{\pm 0.0012}$	$0.4984_{\pm 0.0013}$	$0.4575_{\pm 0.0013}$	$0.3664_{\pm 0.0016}$
BELT-H + thresholds	$0.5398_{\pm 0.0032}$	$0.5249_{\pm 0.0029}$	$0.5026_{\pm 0.0031}$	$\underline{0.4625}_{\pm 0.0025}$	$\mathbf{0.3711}_{\pm 0.0019}$
Qwen3 OOD (Macro)					
OSP	$0.6547_{\pm 0.0067}$	$0.6433_{\pm 0.0059}$	$0.6241_{\pm 0.0047}$	$\underline{0.5869}_{\pm 0.0027}$	$0.4996_{\pm 0.0047}$
OSP + thresholds	$0.6605_{\pm 0.0043}$	$0.6471_{\pm 0.0032}$	$0.6253_{\pm 0.0033}$	$0.5867_{\pm 0.0061}$	$0.5059_{\pm 0.0017}$
BELT-B	$\underline{0.6631}_{\pm 0.0003}$	$\underline{0.6487}_{\pm 0.0007}$	$0.6260_{\pm 0.0016}$	$0.5815_{\pm 0.0025}$	$0.4953_{\pm 0.0056}$
BELT-B + thresholds	$\mathbf{0.6660}_{\pm 0.0029}$	$\mathbf{0.6504}_{\pm 0.0022}$	$\mathbf{0.6271}_{\pm 0.0021}$	$0.5825_{\pm 0.0012}$	$0.5000_{\pm 0.0037}$
BELT-H	$0.6570_{\pm 0.0033}$	$0.6437_{\pm 0.0028}$	$0.6230_{\pm 0.0011}$	$0.5862_{\pm 0.0010}$	$\underline{0.5080}_{\pm 0.0030}$
BELT-H + thresholds	$0.6619_{\pm 0.0035}$	$0.6479_{\pm 0.0034}$	$\underline{0.6267}_{\pm 0.0020}$	$\mathbf{0.5901}_{\pm 0.0010}$	$\mathbf{0.5123}_{\pm 0.0033}$
LiveCodeBench Pre					
OSP	$0.6144_{\pm 0.0045}$	$0.5967_{\pm 0.0087}$	$0.5722_{\pm 0.0138}$	$0.5327_{\pm 0.0121}$	$0.4367_{\pm 0.0091}$
OSP + thresholds	$\underline{0.6522}_{\pm 0.0218}$	$\underline{0.6345}_{\pm 0.0208}$	$\underline{0.6061}_{\pm 0.0174}$	$\underline{0.5511}_{\pm 0.0078}$	$0.4421_{\pm 0.0042}$
BELT-B	$0.6482_{\pm 0.0166}$	$0.6303_{\pm 0.0168}$	$0.6004_{\pm 0.0171}$	$0.5455_{\pm 0.0109}$	$0.4441_{\pm 0.0051}$
BELT-B + thresholds	$\mathbf{0.6562}_{\pm 0.0052}$	$\mathbf{0.6384}_{\pm 0.0052}$	$\mathbf{0.6106}_{\pm 0.0045}$	$\mathbf{0.5572}_{\pm 0.0021}$	$\mathbf{0.4494}_{\pm 0.0038}$
BELT-H	$0.6237_{\pm 0.0079}$	$0.6096_{\pm 0.0072}$	$0.5861_{\pm 0.0059}$	$0.5404_{\pm 0.0038}$	$0.4445_{\pm 0.0050}$
BELT-H + thresholds	$0.6318_{\pm 0.0025}$	$0.6159_{\pm 0.0017}$	$0.5905_{\pm 0.0023}$	$0.5434_{\pm 0.0041}$	$\underline{0.4470}_{\pm 0.0055}$
LiveCodeBench Post					
OSP	$0.2796_{\pm 0.0104}$	$0.2680_{\pm 0.0083}$	$0.2488_{\pm 0.0049}$	$\underline{0.2124}_{\pm 0.0014}$	$\underline{0.1496}_{\pm 0.0049}$
OSP + thresholds	$\underline{0.3003}_{\pm 0.0087}$	$\underline{0.2798}_{\pm 0.0046}$	$\mathbf{0.2520}_{\pm 0.0025}$	$\mathbf{0.2145}_{\pm 0.0048}$	$\mathbf{0.1507}_{\pm 0.0043}$
BELT-B	$0.2972_{\pm 0.0118}$	$0.2738_{\pm 0.0100}$	$0.2431_{\pm 0.0035}$	$0.2046_{\pm 0.0019}$	$0.1368_{\pm 0.0015}$
BELT-B + thresholds	$\mathbf{0.3059}_{\pm 0.0029}$	$\mathbf{0.2834}_{\pm 0.0024}$	$\underline{0.2510}_{\pm 0.0026}$	$0.2071_{\pm 0.0036}$	$0.1378_{\pm 0.0009}$
BELT-H	$0.2817_{\pm 0.0060}$	$0.2655_{\pm 0.0043}$	$0.2426_{\pm 0.0015}$	$0.2038_{\pm 0.0006}$	$0.1431_{\pm 0.0022}$
BELT-H + thresholds	$0.2894_{\pm 0.0018}$	$0.2703_{\pm 0.0010}$	$0.2446_{\pm 0.0014}$	$0.2076_{\pm 0.0020}$	$0.1443_{\pm 0.0021}$

B.7 BUDGET CONTROL WITHOUT POLICY MIXTURES

Tables 43 to 44 select a single feasible policy without interpolation. Comparing it with the mixture protocol shows the effect of member spacing: a single policy can leave budget unused, while a mixture reaches costs between adjacent frontier members. Exports retain realized costs for both.

Table 43: Macro accuracy (%) under single-member token budgets. At each target, the most accurate feasible member is selected without interpolation. Entries are mean$_{\pm SD}$ across seeds; NF denotes an infeasible target. Panel: ID.

Method	70%	60%	50%	40%
OSP	$42.60_{\pm 1.54}$	$44.93_{\pm 0.50}$	$48.44_{\pm 2.70}$	$50.97_{\pm 1.77}$
OSP-1	$42.42_{\pm 1.28}$	$46.34_{\pm 0.07}$	$49.33_{\pm 0.42}$	$50.93_{\pm 1.62}$
BELT-B-1	$35.28_{\pm 3.02}$	$42.42_{\pm 1.64}$	$48.43_{\pm 2.27}$	$51.92_{\pm 1.45}$
BELT-H-1	$38.50_{\pm 1.54}$	$47.93_{\pm 0.83}$	$51.20_{\pm 0.23}$	$52.84_{\pm 0.21}$
BELT-B	$42.80_{\pm 0.03}$	$46.94_{\pm 1.62}$	$50.58_{\pm 1.19}$	$51.88_{\pm 0.63}$
BELT-H	$45.20_{\pm 0.39}$	$48.16_{\pm 0.29}$	$50.48_{\pm 0.11}$	$53.03_{\pm 0.15}$

Table 44: Macro accuracy (%) under single-member token budgets. At each target, the most accurate feasible member is selected without interpolation. Entries are mean$_{\pm SD}$ across seeds; NF denotes an infeasible target. Panel: OOD.

Method	70%	60%	50%	40%
OSP	NF	$59.79_{\pm 1.73}$	$60.65_{\pm 1.65}$	$63.84_{\pm 2.15}$
OSP-1	$55.89_{\pm 1.04}$	$59.57_{\pm 0.29}$	$61.64_{\pm 0.15}$	$63.62_{\pm 0.23}$
BELT-B-1	$42.19_{\pm 8.75}$	$50.45_{\pm 6.09}$	$58.57_{\pm 1.85}$	$63.09_{\pm 0.74}$
BELT-H-1	$48.58_{\pm 3.23}$	$56.46_{\pm 1.74}$	$62.23_{\pm 0.62}$	$64.94_{\pm 0.35}$
BELT-B	$56.43_{\pm 1.40}$	$60.42_{\pm 1.66}$	$62.98_{\pm 0.25}$	$64.53_{\pm 0.99}$
BELT-H	$57.96_{\pm 0.49}$	$60.63_{\pm 0.41}$	$62.79_{\pm 0.23}$	$65.30_{\pm 0.26}$

B.8 IN-DISTRIBUTION ACCURACY–COST TRADE-OFFS

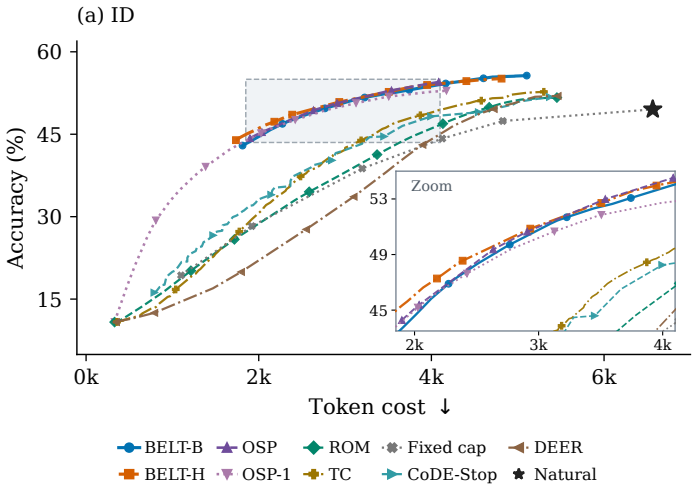

Figure 5: Qwen3 ID frontiers. The presentation follows Figure 2.

B.9 ACCURACY–COST TRADE-OFFS BY DATASET

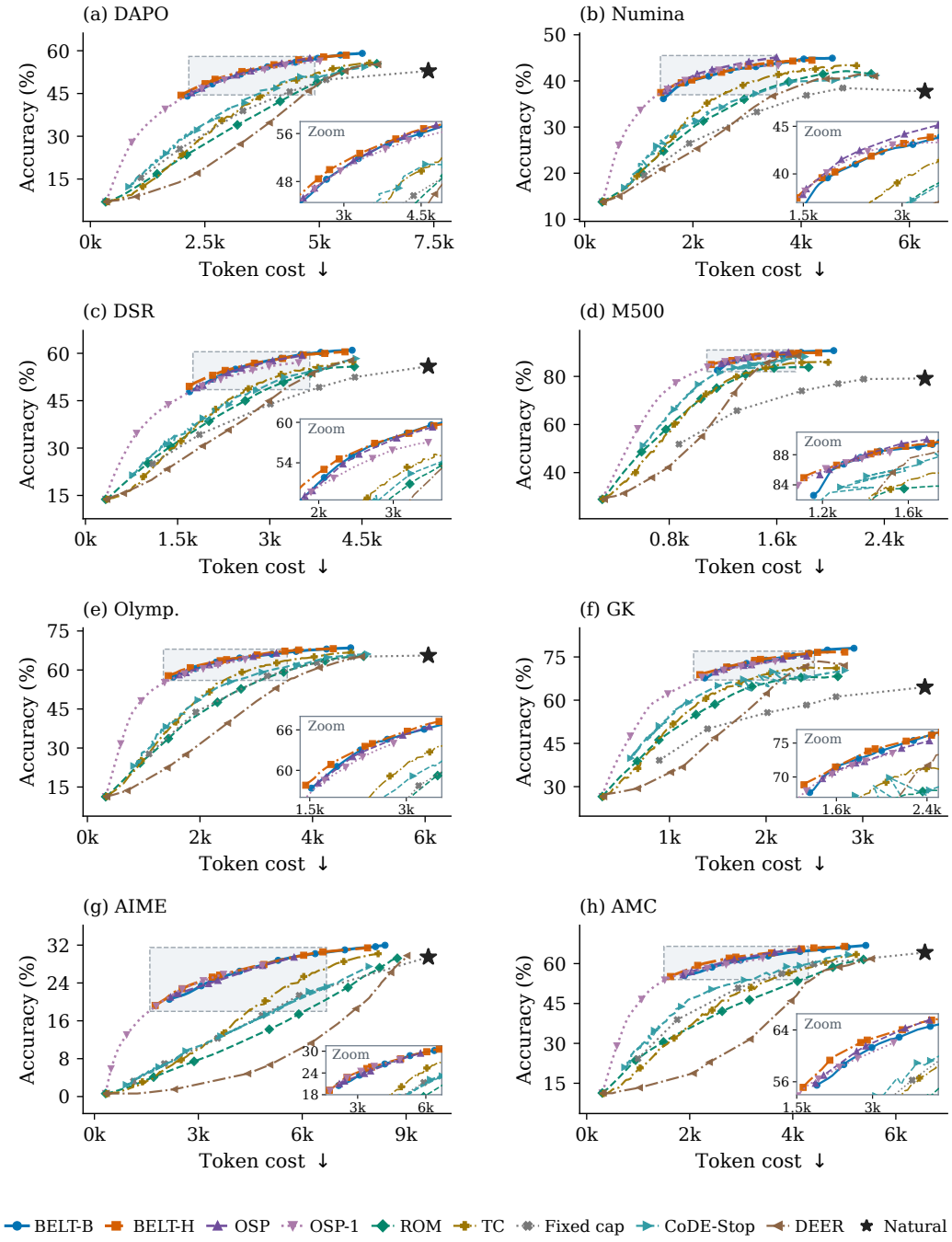

Figure 6: Dataset-level frontiers. The presentation follows Figure 2. Axes differ across panels.

B.10 OLMo3 ACCURACY–TOKEN COST TRADE-OFFS

Fig. 7 shows the OLMo3 frontiers and operating points of Table 3, evaluated on the test frontier of each method.

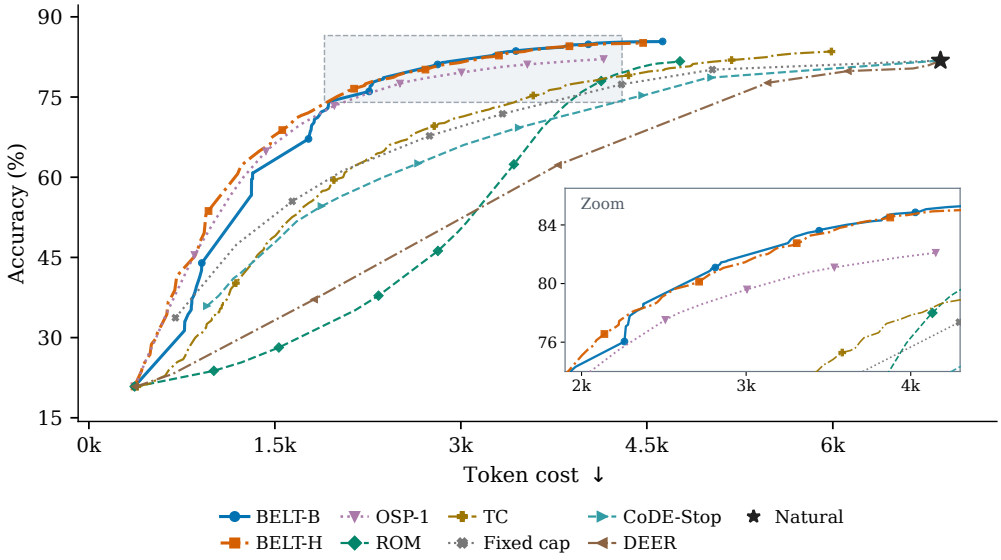

Figure 7: OLMo3 OOD frontiers. The presentation follows Figure 2. Candidate-set-size ablations appear only in Table 3.

C Belief and Policy Analysis

C.1 Features, policy capacity, and belief ablations

Shared evaluation protocol. The utility controls in Tables 46, 47, 54 and 55 evaluate dev-selected policies on the main 1,025-question OOD roster, with strict test grading, total generated-token cost, and micro average utility. They use the 160-question policy-fitting and 160-question policy development sets in § A.3. Fitting and development labels use archived numeric-equality grading; the matched OSP-1 comparison below instead selects by strict-grading development utility. Each reported token price selects a policy before test evaluation.

Table 45 summarizes the grading conventions and sources of variation.

Table 45: Grading and variability in utility comparisons. The 320-question bank first selects the BELT probe checkpoint; its disjoint 160-question halves then fit and select policies. The final column identifies whether sample SD is computed across independent training runs or policy-fitting starts. All test costs count total generated tokens.

Comparison	Policy fitting / development selection	Test grading	SD across
Qwen3 BELT utility controls	Numeric-equality labels	Strict math	3 independent probe training runs
Fixed feature controls	Numeric-equality labels	Strict math	5 policy-fitting starts
Linear four-belief probe	Numeric-equality labels	Strict math	3 probe training seeds
Matched OSP-1/BELT features	Shared fitter; strict math development utility	Strict math; code unit tests	3 independent training runs per method
OLMo3 BELT	Archived loose-label utility	Strict math	3 independent probe training runs

Features and policy capacity. Table 46a reports the four fixed-feature controls alongside BELT. Each control fixes its feature bank and varies five policy-fitting starts, so its SD measures optimization variability. The BELT row instead varies three independent probe training runs; these axes are not paired. Raw-state and pooling controls change prefix access and architecture; PCA matches the 768-dimensional representation. The before-training control uses the pretrained ReProbe checkpoint with the same architecture and no belief training. It measures what belief supervision adds to the inherited representation. The linear four-belief probe maps the 2,048-dimensional layer-16 block-end state directly to the four targets, using the loss in Eq. (4). We train it with three seeds and select checkpoints by minimum validation belief loss. Its validation *Now* AUROC is 0.77, compared with 0.90 for the BELT probe. We then fit the BELT-B policy on its four logits and prefix length. This tests the targets without the sequence probe; SD spans the three probe seeds.

The MLP and linear policies in Table 46b use the same four belief logits, prefix length, and three independent BELT probe training runs; the MLP follows the recorded optimization recipe. The hand rules change the available information, using only their named beliefs.

Training targets and policy inputs. Target ablations retrain the BELT probe and reselect its checkpoint, then fit a policy on the retained beliefs. Input ablations keep the four-target BELT probe fixed and refit the policy after removing belief coordinates. Both retain prefix length and report mean and sample SD over three independent probe training runs. Table 47 compares both interventions across all five prices. The common full-input row is the same experiment. Target ablations measure the combined effect of supervision, checkpoint selection, and the associated output interface; input ablations measure feature removal with the trained probe held fixed.

Adding *Later* to *Now*/*End* yields higher mean utility at all five reported token prices, with gains of 0.0004–0.0222, even though *Now* AP falls from 0.7719 to 0.7587 (Table 57). With the four-target BELT probe fixed, removing *Later* reduces mean utility by 0.0004–0.0023 across prices, compared with 0.0035–0.0094 for *Same*.

Table 46: Utility comparing features (a) and controller capacity (b). Mean and SD are reported over five policy-fitting starts for the first four feature controls, and three training seeds for OSP-1, the linear four-belief probe and BELT policies.

Control	**Utility by λ ($\times 10^{-6}$)**				
	2	**5**	**10**	**20**	**50**
(a) Features for a linear stopping policy					
BELT probe before belief training	$0.655_{\pm 0.001}$	$0.643_{\pm 0.001}$	$0.623_{\pm 0.001}$	$0.584_{\pm 0.001}$	$0.491_{\pm 0.005}$
Reasoner state (layer 16)	$0.665_{\pm 0.002}$	$0.656_{\pm 0.005}$	$0.636_{\pm 0.006}$	$0.597_{\pm 0.006}$	$0.510_{\pm 0.001}$
Reasoner state (PCA, 768-d)	$0.664_{\pm 0.005}$	$0.652_{\pm 0.005}$	$0.632_{\pm 0.006}$	$0.592_{\pm 0.005}$	$0.502_{\pm 0.002}$
Mean-pooled reasoner states	$0.663_{\pm 0.002}$	$0.651_{\pm 0.002}$	$0.632_{\pm 0.002}$	$0.592_{\pm 0.002}$	$0.480_{\pm 0.004}$
OSP-1 features	$0.698_{\pm 0.003}$	$0.686_{\pm 0.003}$	$0.668_{\pm 0.002}$	$0.635_{\pm 0.005}$	$0.565_{\pm 0.007}$
OSP-1 features (multi-start)	$0.698_{\pm 0.003}$	$0.686_{\pm 0.003}$	$\underline{0.669}_{\pm 0.005}$	$\underline{0.637}_{\pm 0.006}$	$\underline{0.567}_{\pm 0.004}$
Linear four-belief probe (2048→4)	$\underline{0.701}_{\pm 0.001}$	$\underline{0.689}_{\pm 0.001}$	$0.668_{\pm 0.001}$	$0.629_{\pm 0.001}$	$0.545_{\pm 0.003}$
BELT probe	$\mathbf{0.714}_{\pm 0.006}$	$\mathbf{0.702}_{\pm 0.007}$	$\mathbf{0.683}_{\pm 0.005}$	$\mathbf{0.654}_{\pm 0.002}$	$\mathbf{0.579}_{\pm 0.004}$
(b) Controllers on a fixed BELT probe					
Rule: $p^{\text{now}} \geq \tau$	$0.680_{\pm 0.002}$	$0.668_{\pm 0.002}$	$0.647_{\pm 0.002}$	$0.607_{\pm 0.002}$	$0.467_{\pm 0.009}$
Rule: $p^{\text{now}} \geq p^{\text{end}} - m$	$\underline{0.701}_{\pm 0.014}$	$0.687_{\pm 0.014}$	$\underline{0.665}_{\pm 0.014}$	$\underline{0.625}_{\pm 0.010}$	$\underline{0.556}_{\pm 0.011}$
Linear	$\mathbf{0.723}_{\pm 0.000}$	$\underline{0.710}_{\pm 0.001}$	$\mathbf{0.690}_{\pm 0.001}$	$\mathbf{0.652}_{\pm 0.004}$	$\mathbf{0.574}_{\pm 0.004}$
MLP	$\mathbf{0.723}_{\pm 0.002}$	$\mathbf{0.711}_{\pm 0.002}$	$\mathbf{0.690}_{\pm 0.002}$	$\mathbf{0.652}_{\pm 0.002}$	$\mathbf{0.574}_{\pm 0.005}$

Table 47: Utility comparing belief training targets and policy inputs across five token prices. Mean and sample SD cover three BELT probe training seeds. The two panels respectively retrain the BELT probe and refit policies on a fixed four-target BELT probe. Values at $\lambda = 10^{-5}$ retain four decimal places.

Control	**Utility by λ ($\times 10^{-6}$)**				
	2	**5**	**10**	**20**	**50**
BELT probe training targets					
Now, *End*	$0.716_{\pm 0.004}$	$0.703_{\pm 0.003}$	$0.6826_{\pm 0.0012}$	$0.638_{\pm 0.008}$	$0.550_{\pm 0.007}$
Now, *End*, *Later*	$0.716_{\pm 0.007}$	$0.707_{\pm 0.005}$	$0.6854_{\pm 0.0048}$	$0.646_{\pm 0.004}$	$\underline{0.572}_{\pm 0.007}$
Now, *End*, *Same*	$\underline{0.720}_{\pm 0.003}$	$\underline{0.708}_{\pm 0.003}$	$\underline{0.6890}_{\pm 0.0026}$	$\mathbf{0.653}_{\pm 0.005}$	$0.571_{\pm 0.007}$
All four	$\mathbf{0.723}_{\pm 0.000}$	$\mathbf{0.710}_{\pm 0.001}$	$\mathbf{0.6904}_{\pm 0.0010}$	$\underline{0.652}_{\pm 0.004}$	$\mathbf{0.574}_{\pm 0.004}$
Policy inputs from a fixed four-target BELT probe					
Now, *End*	$\underline{0.722}_{\pm 0.001}$	$0.704_{\pm 0.007}$	$0.6746_{\pm 0.0150}$	$0.637_{\pm 0.015}$	$0.555_{\pm 0.016}$
Now, *End*, *Later*	$0.720_{\pm 0.004}$	$\underline{0.707}_{\pm 0.003}$	$\underline{0.6823}_{\pm 0.0038}$	$0.643_{\pm 0.008}$	$0.569_{\pm 0.005}$
Now, *End*, *Same*	$\mathbf{0.723}_{\pm 0.001}$	$\mathbf{0.710}_{\pm 0.002}$	$\mathbf{0.6900}_{\pm 0.0013}$	$\underline{0.650}_{\pm 0.002}$	$\underline{0.572}_{\pm 0.004}$
All four	$\mathbf{0.723}_{\pm 0.000}$	$\mathbf{0.710}_{\pm 0.001}$	$\mathbf{0.6904}_{\pm 0.0010}$	$\mathbf{0.652}_{\pm 0.004}$	$\mathbf{0.574}_{\pm 0.004}$

C.2 Matched feature comparison and transfer

Matched policy fitting. We freeze three OSP-1 checkpoints trained at $\lambda = 10^{-5}$ and extract their 2,048-dimensional boundary states after final normalization. OSP-1 features and BELT-B/H receive the same 17-price grid, linear fitter, standardization, and regularization search. At each reported price, strict-grading micro average utility on the policy development set selects among the 17 policies and natural completion. The main fit starts from zero weights; a four-start sensitivity adds three random initializations for each feature set. Mean and SD cover three independent training

runs per method, without cross-method pairing. This matches fitting opportunities while retaining different architectures and feature dimensions.

BELT-H exceeds OSP-1 features at every reported price on math ID and OOD, including with four starts (Tables 48 to 51). The same fits and source-development selections transfer to LiveCodeBench without code-side fitting or selection. In the main fits, BELT-H has higher mean utility on both splits, while BELT-B leads on Pre and at the two lowest prices on Post. The ordering varies at some code operating points: at $\lambda = 2 \times 10^{-5}$ on Pre, four-start mean utility is 0.5268 for BELT-H and 0.5288 for OSP-1 features, and BELT-B trails OSP-1 at intermediate prices on Post. BELT nevertheless remains competitive in these comparisons, with a 0.0020 utility gap in the four-start Pre example, while BELT-H leads across all reported math prices and retains its main-fit advantage on both code splits. Its probe uses $6.3\times$ fewer parameters than one OSP checkpoint (§ 6.4). Tables 52 and 53 reports every independent training run so that the spread underlying each mean is visible. Question-bootstrap intervals use 5,000 shared resamples with policies and selections fixed, and therefore exclude training-run and selector uncertainty.

Table 48: Utility under matched fitting and selection budgets. Reporting convention follows Table 1; all values are micro averages. Panel: Math ID. Token prices λ are in 10^{-6} units.

Features / fitting	**2**	**5**	**10**	**20**	**50**
OSP-1 features	$0.5240_{\pm 0.0038}$	$0.5098_{\pm 0.0034}$	$0.4861_{\pm 0.0027}$	$0.4446_{\pm 0.0042}$	$0.3520_{\pm 0.0121}$
OSP-1 features (four starts)	$0.5239_{\pm 0.0036}$	$0.5098_{\pm 0.0034}$	$0.4877_{\pm 0.0035}$	$0.4472_{\pm 0.0022}$	$0.3539_{\pm 0.0097}$
BELT-H	$\underline{0.5411}_{\pm 0.0016}$	$0.5262_{\pm 0.0018}$	$0.5014_{\pm 0.0015}$	$\underline{0.4595}_{\pm 0.0013}$	$0.3599_{\pm 0.0036}$
BELT-H (four starts)	$0.5399_{\pm 0.0004}$	$0.5255_{\pm 0.0007}$	$0.5028_{\pm 0.0010}$	$0.4591_{\pm 0.0009}$	$\mathbf{0.3629}_{\pm 0.0031}$
BELT-B	$\mathbf{0.5463}_{\pm 0.0035}$	$\underline{0.5303}_{\pm 0.0043}$	$\underline{0.5048}_{\pm 0.0042}$	$\mathbf{0.4603}_{\pm 0.0019}$	$\underline{0.3603}_{\pm 0.0028}$
BELT-B (four starts)	$\mathbf{0.5463}_{\pm 0.0042}$	$\mathbf{0.5311}_{\pm 0.0040}$	$\mathbf{0.5057}_{\pm 0.0034}$	$0.4573_{\pm 0.0032}$	$\underline{0.3603}_{\pm 0.0029}$

Table 49: Utility under matched fitting and selection budgets. Reporting convention follows Table 1; all values are micro averages. Panel: Math OOD. Token prices λ are in 10^{-6} units.

Features / fitting	**2**	**5**	**10**	**20**	**50**
OSP-1 features	$0.6977_{\pm 0.0032}$	$0.6862_{\pm 0.0032}$	$0.6677_{\pm 0.0021}$	$0.6351_{\pm 0.0050}$	$0.5651_{\pm 0.0066}$
OSP-1 features (four starts)	$0.6976_{\pm 0.0033}$	$0.6864_{\pm 0.0029}$	$0.6689_{\pm 0.0051}$	$0.6375_{\pm 0.0056}$	$0.5665_{\pm 0.0042}$
BELT-H	$0.7141_{\pm 0.0064}$	$0.7024_{\pm 0.0068}$	$0.6835_{\pm 0.0048}$	$\mathbf{0.6543}_{\pm 0.0021}$	$\underline{0.5791}_{\pm 0.0037}$
BELT-H (four starts)	$0.7128_{\pm 0.0044}$	$0.7013_{\pm 0.0055}$	$0.6839_{\pm 0.0052}$	$\mathbf{0.6543}_{\pm 0.0021}$	$\mathbf{0.5817}_{\pm 0.0025}$
BELT-B	$\mathbf{0.7232}_{\pm 0.0004}$	$\mathbf{0.7104}_{\pm 0.0012}$	$\mathbf{0.6904}_{\pm 0.0010}$	$\underline{0.6524}_{\pm 0.0040}$	$0.5738_{\pm 0.0042}$
BELT-B (four starts)	$\underline{0.7222}_{\pm 0.0010}$	$\underline{0.7102}_{\pm 0.0009}$	$\underline{0.6872}_{\pm 0.0032}$	$0.6478_{\pm 0.0007}$	$0.5742_{\pm 0.0040}$

Table 50: Utility under matched fitting and selection budgets. Reporting convention follows Table 1; all values are micro averages. Panel: LiveCodeBench Pre. Token prices λ are in 10^{-6} units.

Features / fitting	**2**	**5**	**10**	**20**	**50**
OSP-1 features	$0.5934_{\pm 0.0478}$	$0.5809_{\pm 0.0435}$	$0.5590_{\pm 0.0353}$	$0.5261_{\pm 0.0141}$	$0.4222_{\pm 0.0090}$
OSP-1 features (four starts)	$0.5935_{\pm 0.0477}$	$0.5809_{\pm 0.0434}$	$0.5640_{\pm 0.0395}$	$0.5288_{\pm 0.0169}$	$0.4200_{\pm 0.0068}$
BELT-H	$0.6144_{\pm 0.0164}$	$0.5982_{\pm 0.0174}$	$0.5718_{\pm 0.0109}$	$\underline{0.5366}_{\pm 0.0078}$	$0.4390_{\pm 0.0040}$
BELT-H (four starts)	$0.6167_{\pm 0.0180}$	$0.6005_{\pm 0.0197}$	$0.5781_{\pm 0.0162}$	$0.5268_{\pm 0.0237}$	$0.4386_{\pm 0.0040}$
BELT-B	$\mathbf{0.6363}_{\pm 0.0211}$	$\mathbf{0.6179}_{\pm 0.0213}$	$\underline{0.5866}_{\pm 0.0226}$	$\mathbf{0.5387}_{\pm 0.0155}$	$\underline{0.4404}_{\pm 0.0027}$
BELT-B (four starts)	$\underline{0.6362}_{\pm 0.0208}$	$\underline{0.6175}_{\pm 0.0215}$	$\mathbf{0.5887}_{\pm 0.0214}$	$0.5277_{\pm 0.0226}$	$\mathbf{0.4417}_{\pm 0.0004}$

Table 51: Utility under matched fitting and selection budgets. Reporting convention follows Table 1; all values are micro averages. Panel: LiveCodeBench Post. Token prices λ are in 10^{-6} units.

Features / fitting	**2**	**5**	**10**	**20**	**50**
OSP-1 features	$0.2721_{\pm 0.0183}$	$0.2562_{\pm 0.0128}$	$0.2311_{\pm 0.0070}$	$0.1826_{\pm 0.0220}$	$0.1084_{\pm 0.0354}$
OSP-1 features (four starts)	$0.2723_{\pm 0.0184}$	$0.2564_{\pm 0.0129}$	$\underline{0.2333}_{\pm 0.0052}$	$0.1900_{\pm 0.0115}$	$0.1164_{\pm 0.0188}$
BELT-H	$0.2784_{\pm 0.0047}$	$0.2580_{\pm 0.0039}$	$0.2328_{\pm 0.0067}$	$\mathbf{0.1964}_{\pm 0.0082}$	$\underline{0.1196}_{\pm 0.0098}$
BELT-H (four starts)	$0.2799_{\pm 0.0062}$	$0.2608_{\pm 0.0070}$	$\mathbf{0.2336}_{\pm 0.0080}$	$\underline{0.1963}_{\pm 0.0079}$	$\mathbf{0.1211}_{\pm 0.0111}$
BELT-B	$\mathbf{0.2909}_{\pm 0.0162}$	$\mathbf{0.2653}_{\pm 0.0166}$	$0.2215_{\pm 0.0189}$	$0.1580_{\pm 0.0243}$	$0.1045_{\pm 0.0064}$
BELT-B (four starts)	$\underline{0.2908}_{\pm 0.0157}$	$\underline{0.2646}_{\pm 0.0170}$	$0.2209_{\pm 0.0203}$	$0.1439_{\pm 0.0374}$	$0.1070_{\pm 0.0090}$

Table 52: Individual training runs for the matched feature comparison on LiveCodeBench Pre. Seed indices are not paired across methods.

Features / fitting / seed	Utility by λ ($\times 10^{-6}$)				
	2	5	10	20	50
OSP-1 features, main, s0	0.6106	0.5973	0.5753	0.5324	0.4169
OSP-1 features, main, s1	0.6302	0.6139	0.5832	0.5359	0.4325
OSP-1 features, main, s2	0.5394	0.5316	0.5185	0.5100	0.4171
OSP-1 features, four starts, s0	0.6112	0.5979	0.5874	0.5426	0.4179
OSP-1 features, four starts, s1	0.6297	0.6132	0.5862	0.5339	0.4275
OSP-1 features, four starts, s2	0.5394	0.5316	0.5185	0.5100	0.4145
BELT-H, main, s0	0.5955	0.5781	0.5603	0.5293	0.4346
BELT-H, main, s1	0.6253	0.6085	0.5821	0.5448	0.4403
BELT-H, main, s2	0.6224	0.6081	0.5731	0.5358	0.4422
BELT-H, four starts, s0	0.5959	0.5781	0.5603	0.4999	0.4346
BELT-H, four starts, s1	0.6253	0.6085	0.5820	0.5448	0.4388
BELT-H, four starts, s2	0.6288	0.6150	0.5920	0.5358	**0.4425**
BELT-B, main, s0	**0.6573**	**0.6395**	**0.6099**	**0.5503**	0.4373
BELT-B, main, s1	0.6152	0.5970	0.5647	0.5211	0.4416
BELT-B, main, s2	0.6364	0.6172	0.5852	0.5447	0.4424
BELT-B, four starts, s0	**0.6573**	**0.6395**	**0.6099**	**0.5503**	0.4421
BELT-B, four starts, s1	0.6157	0.5966	0.5671	0.5051	0.4413
BELT-B, four starts, s2	0.6354	0.6165	0.5891	0.5276	0.4416

Table 53: Individual training runs for the matched feature comparison on LiveCodeBench Post. Seed indices are not paired across methods.

Features / fitting / seed	Utility by λ ($\times 10^{-6}$)				
	2	5	10	20	50
OSP-1 features, main, s0	0.2745	0.2566	0.2267	0.1583	0.0675
OSP-1 features, main, s1	0.2890	0.2687	0.2392	0.1883	0.1296
OSP-1 features, main, s2	0.2527	0.2432	0.2274	0.2012	0.1282
OSP-1 features, four starts, s0	0.2749	0.2568	0.2369	0.1783	0.1000
OSP-1 features, four starts, s1	0.2893	0.2691	0.2358	0.1905	**0.1369**
OSP-1 features, four starts, s2	0.2527	0.2432	0.2274	0.2012	0.1122
BELT-H, main, s0	0.2744	0.2536	0.2332	**0.2038**	0.1309
BELT-H, main, s1	0.2836	0.2609	0.2259	0.1876	0.1134
BELT-H, main, s2	0.2772	0.2594	0.2393	0.1980	0.1146
BELT-H, four starts, s0	0.2728	0.2536	0.2332	0.2033	0.1309
BELT-H, four starts, s1	0.2836	0.2610	0.2258	0.1878	0.1091
BELT-H, four starts, s2	0.2833	0.2677	0.2419	0.1980	0.1234
BELT-B, main, s0	**0.3077**	**0.2831**	**0.2420**	0.1848	0.1086
BELT-B, main, s1	0.2753	0.2502	0.2050	0.1374	0.1077
BELT-B, main, s2	0.2896	0.2625	0.2174	0.1518	0.0971
BELT-B, four starts, s0	**0.3077**	**0.2831**	**0.2420**	0.1858	0.1153
BELT-B, four starts, s1	0.2767	0.2497	0.2015	0.1137	0.1083
BELT-B, four starts, s2	0.2879	0.2611	0.2193	0.1324	0.0974

C.3 BELT PROBE CAPACITY AND INPUT SUBSETS

BELT probe capacity. Replacing the BELT probe with the reasoner's top-two-layer branch raises utility from 0.6826 to 0.6849 under *Now*/*End* supervision and from 0.6854 to 0.6856 un-

der *Now*/*End*/*Later* supervision at $\lambda = 10^{-5}$ (Table 54). The compact BELT probe remains within 0.0023 utility of the larger branch in both comparisons, using $6.3\times$ fewer parameters (§ 6.4). These comparisons evaluate belief-input policies and jointly vary architecture, readout, and parameter count.

Table 54: Utility comparing BELT probe architectures with belief inputs. Reporting convention follows Table 47.

Architecture / prediction targets	Utility by λ ($\times 10^{-6}$)				
	2	5	10	20	50
BELT probe / *Now*, *End*	$0.7158_{\pm 0.0043}$	$0.7034_{\pm 0.0031}$	$0.6826_{\pm 0.0012}$	$0.6382_{\pm 0.0076}$	$0.5499_{\pm 0.0072}$
Top-two branch / *Now*, *End*	$\underline{0.7199}_{\pm 0.0036}$	$\underline{0.7068}_{\pm 0.0025}$	$0.6849_{\pm 0.0005}$	$0.6397_{\pm 0.0057}$	$0.5569_{\pm 0.0084}$
BELT probe / *Now*, *End*, *Later*	$0.7162_{\pm 0.0069}$	$\underline{0.7068}_{\pm 0.0053}$	$\underline{0.6854}_{\pm 0.0048}$	$\underline{0.6458}_{\pm 0.0044}$	$\mathbf{0.5720}_{\pm 0.0065}$
Top-two branch / *Now*, *End*, *Later*	$\mathbf{0.7212}_{\pm 0.0006}$	$\mathbf{0.7078}_{\pm 0.0011}$	$\mathbf{0.6856}_{\pm 0.0021}$	$\mathbf{0.6473}_{\pm 0.0041}$	$\underline{0.5665}_{\pm 0.0047}$

Combining beliefs with observed length. Table 55 keeps the four-target BELT probe fixed and refits the linear stopping policy. At $\lambda = 10^{-5}$, adding observed prefix length to *Now* raises mean utility from 0.6497 to 0.6645. *Now*/*Same* with length reaches 0.6807. The complete input set reaches 0.6904, as reported in Table 47. The *Now*-only comparison quantifies the benefit of prefix length for that input set.

Table 55: Utility comparing policy input subsets. Reporting convention follows Table 47.

Policy inputs	Utility by λ ($\times 10^{-6}$)				
	2	5	10	20	50
Now only	$0.6831_{\pm 0.0096}$	$0.6709_{\pm 0.0095}$	$0.6497_{\pm 0.0082}$	$0.6087_{\pm 0.0075}$	$0.4868_{\pm 0.0067}$
Now + length	$0.6923_{\pm 0.0233}$	$0.6804_{\pm 0.0239}$	$0.6645_{\pm 0.0190}$	$0.6331_{\pm 0.0085}$	$0.5320_{\pm 0.0126}$
End + length	$0.6740_{\pm 0.0008}$	$0.6595_{\pm 0.0006}$	$0.6354_{\pm 0.0005}$	$0.5903_{\pm 0.0049}$	$0.5121_{\pm 0.0160}$
Now, *Same* + length	$\mathbf{0.7102}_{\pm 0.0058}$	$\mathbf{0.6989}_{\pm 0.0057}$	$\mathbf{0.6807}_{\pm 0.0061}$	$\mathbf{0.6447}_{\pm 0.0042}$	$\mathbf{0.5597}_{\pm 0.0090}$
Same + length	$\underline{0.7090}_{\pm 0.0058}$	$\underline{0.6977}_{\pm 0.0053}$	$\underline{0.6789}_{\pm 0.0046}$	$\underline{0.6433}_{\pm 0.0013}$	$\underline{0.5506}_{\pm 0.0032}$
End, *Later* + length	$0.6739_{\pm 0.0006}$	$0.6595_{\pm 0.0005}$	$0.6355_{\pm 0.0002}$	$0.6015_{\pm 0.0082}$	$0.5155_{\pm 0.0135}$

C.4 Information provided to the controller by the four beliefs

Weights and stopping regions. Fig. 8a shows that BELT-B shifts relative weight from *Now* toward *Same* as the token price increases, while *End* favors continuation and *Later* becomes more negative. Across three independent BELT probe training runs, the mean threshold along the normalized policy direction falls from 1.33 to -0.33. Panels b and c show that, at low prices, stopping concentrates where *Now* is high and *End* is low. At high prices, continuation remains more likely in high-*End*, low-*Now* states (Fig. 8b,c). The direction and threshold change simultaneously.

Coefficient interpretation. The coefficient study uses the 160-question fitting and development splits and fixes $\kappa = 10^{-4}$ across independent probe training runs and token prices. Inputs follow Eq. (5): belief logits and prefix length are standardized on fitting boundaries, with the same transformation across prices within each training run. For belief j, $x_j = (\mathrm{logit}(\mathrm{clip}(p_j)) - \mu_j)/\sigma_j$. Increasing x_j by one changes stopping log-odds by w_j with other inputs fixed. Correlations are 0.78–0.95 for p^{now}, p^{same}, 0.94–0.95 for p^{end}, p^{later}, and -0.60 to -0.53 for p^{end} and prefix length. Coefficients therefore describe conditional roles rather than individual input contributions.

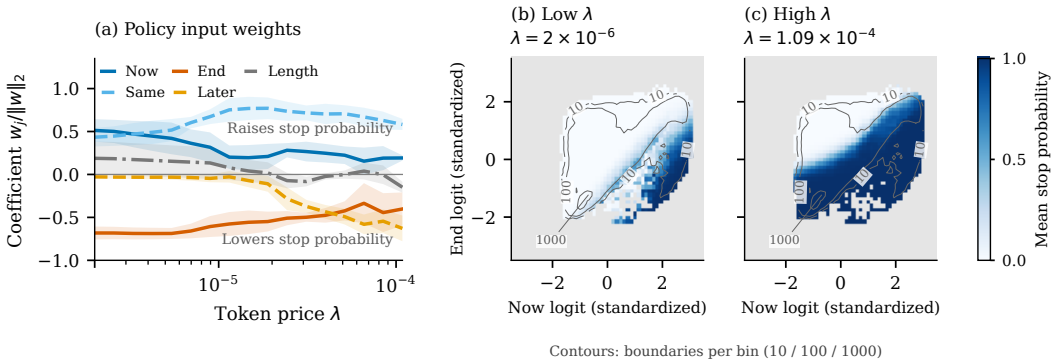

Figure 8: Policy adaptation to λ. (a) Normalized weights, shown as mean and sample SD over three independent BELT probe training runs. (b,c) Stop probabilities from one representative training run. Gray contours show boundary counts; gray cells are empty.

Table 56: Policy geometry at three token prices; λ is in 10^{-6} units. $\text{Mean}_{\pm\text{SD}}$ over three BELT probe seeds; probe size 16M.

λ	2.00	18.91	108.58
Length $\hat{w}_d$	$0.188_{\pm 0.195}$	$0.018_{\pm 0.124}$	$-0.149_{\pm 0.063}$
Threshold t_λ	$1.330_{\pm 0.224}$	$0.956_{\pm 0.028}$	$-0.332_{\pm 0.141}$
Scale a_λ	$6.741_{\pm 0.670}$	$5.906_{\pm 0.229}$	$8.052_{\pm 0.495}$

We separate the direction, threshold, and scale of each fitted policy by writing

$$\begin{aligned} \pi_i &= \sigma\big(a_\lambda[\hat{w}_\lambda^\top x_i - t_\lambda]\big)\,, \\ \hat{w}_\lambda &= w_\lambda/\|w_\lambda\|_2, \qquad t_\lambda = -b_\lambda/\|w_\lambda\|_2, \qquad a_\lambda = \|w_\lambda\|_2. \end{aligned} \tag{8}$$

Here $\hat{w}_\lambda$ is the unit direction, t_λ locates the 0.5 stop-probability boundary along that direction, and a_λ controls the sharpness of the transition. Normalization includes all five input weights and excludes the intercept.

Table 56 reports mean and sample SD after transforming each saved fit separately. These summaries do not define an averaged policy. The mean normalized length coefficient changes from $+0.188$ at the lowest price to -0.149 at the highest. Length is an observed covariate: cost already incurred is shared by stopping and continuing, while utility compares subsequent cost and answer outcomes. Its coefficient describes the conditional role of observed length in the fitted policy.

Heatmap construction. The heatmaps use an archived histogram from one representative training run, containing 212,101 forced-answer boundaries at $\lambda = 2 \times 10^{-6}$ and 1.085767×10^{-4}. They average full-policy stop probabilities over boundaries in a 60×60 *Now–End* grid over $[-4, 4)$ standardized logits. Each boundary retains its $p^{\text{later}}, p^{\text{same}}, d$ values; the displayed axes span $[-3.5, 3.5]$. Gray contours interpolate $\log_{10}$ counts at levels 1, 2, and 3 (10, 100, and 1,000 boundaries). Neither counts nor colors are weighted by question or policy survival. Counts show observed support rather than independent questions or policy visitation, and equal axis coordinates need not denote equal correctness probabilities. This archived illustration is not used to compute current test-performance results.

C.5 Prediction quality across questions and positions

Now and *Same* have within-question AUROCs of 0.853 and 0.928 on the source-validation bank. *End* and *Later* distinguish boundaries from different questions more strongly than boundaries within the same question.

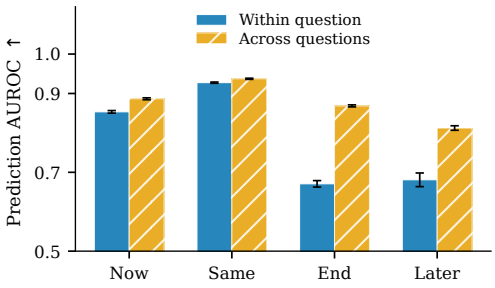

Figure 9: Within- and across-question belief AUROC, shown as mean and sample SD over three independent BELT probe training runs.

Prediction diagnostics use 1,065 source-validation questions (10,648 trajectories), distinct from the 320-question policy bank: 385,277 labelled positions for $p^{\text{now}}, p^{\text{end}}, p^{\text{later}}$ and 190,877 for p^{same}, which requires a parsed forced answer. Target-set comparisons share labelled boundaries within each belief. Within-question AUROC equally averages questions with both label classes; across-question AUROC compares positive/negative boundaries from different questions, not question means (Fig. 9). These diagnostics characterize outcome prediction; the policy-input ablations in § 6.2 quantify contributions to stopping utility.

C.6 Prediction quality under different supervision

Table 57: Belief average precision; mean and SD over three BELT probe seeds. – denotes an unsupervised output.

Training targets	BELT probe Params	*Now* AP ↑	*End* AP ↑	*Later* AP ↑	*Same* AP ↑
Now, *End*	16M	$0.7719_{\pm 0.0019}$	$0.7750_{\pm 0.0083}$	–	–
Now, *End*, *Later*	16M	$0.7587_{\pm 0.0024}$	$0.7694_{\pm 0.0027}$	$0.8235_{\pm 0.0034}$	–
Now, *End*, *Same*	16M	$0.7782_{\pm 0.0009}$	$0.7769_{\pm 0.0070}$	–	$0.9389_{\pm 0.0022}$
All four	16M	$0.7767_{\pm 0.0027}$	$0.7787_{\pm 0.0068}$	$0.8294_{\pm 0.0060}$	$0.9380_{\pm 0.0004}$

Table 57 reports the prediction quality of each supervision configuration on the bank described in § C.5. Adding p^{same} to $p^{\text{now}}, p^{\text{end}}, p^{\text{later}}$ improves p^{now} AP by 0.018, while adding p^{later} to $p^{\text{now}}, p^{\text{end}}$ changes it by -0.013. Auxiliary targets can therefore improve prediction quality for one belief while reducing it for another. Label base rates are 0.258, 0.365, 0.538 and 0.481 for *Now*, *End*, *Later* and *Same*. Targets and labelled supports differ, so AP should be compared across supervision configurations within a target, not used to rank the beliefs' control importance.

C.7 Reading multiple reasoner layers

Table 58 compares four layer-reading schemes. All variants retain four belief targets, two causal Transformer layers, a 768-dimensional representation, and approximately 16.0M parameters. The reference uses the same training recipe and normalization and reads layer 16. Alternatives combine layers 16, 20, 24, and 28 by scalar mixture, compressed concatenation, or dynamic attention. Compressed concatenation projects each 2,048-dimensional layer to 384 dimensions before joining them, matching the reference's input-projection weight count.

The 512-question screen uses two trajectories per question, eight epochs, and one independent probe training run. The full-bank comparison uses up to eight trajectories per question and three independent probe training runs. Both select checkpoints by validation belief loss and fit linear stopping policies on either belief logits plus prefix length or the representation. The 160-question fitting/development sets contain 320/320 trajectories in the screen and 1,278/1,276 in the full-bank study. At each reported price, development utility selects from 17 policies. Thus the panels differ in policy data as well as belief training data.

This auxiliary evaluation uses 8,400 trajectories, numeric-equality grading, and total generated tokens rather than the current strict OOD protocol. In the screen, concatenation improves the belief-input policy at all five prices but the representation-input policy at only the two lowest. Full-bank mean gains are 0.0026–0.0051 and 0.0030–0.0082, respectively, with sample SD over three independent probe training runs. At most prices, concatenation has higher utility in two of the three runs.

Table 58: Utility comparing one- and four-layer inputs to the BELT probe on the historical evaluation bank.

Reasoner layers / policy input	Utility by λ ($\times 10^{-6}$)				
	2	5	10	20	50
(a) Small-data screen: one BELT probe seed					
Layer 16 / beliefs	0.7210	0.7106	0.6952	0.6629	0.5895
Layer 16 / representation	0.7151	0.7047	0.6873	0.6635	**0.5956**
Four-layer mixture / beliefs	0.7197	0.7093	0.6919	0.6573	0.5777
Four-layer mixture / representation	0.7172	0.7059	0.6896	0.6613	0.5858
Compressed concatenation / beliefs	**0.7298**	**0.7189**	**0.7008**	**0.6645**	0.5900
Compressed concatenation / representation	0.7202	0.7096	0.6843	0.6583	0.5821
Four-layer attention / beliefs	0.7216	0.7103	0.6915	0.6545	0.5757
Four-layer attention / representation	0.7020	0.7079	0.6834	0.6602	0.5906
(b) Full-bank training: three BELT probe seeds					
Layer 16 / beliefs	$0.7361_{\pm 0.0039}$	$0.7258_{\pm 0.0039}$	$0.7083_{\pm 0.0049}$	$0.6723_{\pm 0.0072}$	$0.5976_{\pm 0.0074}$
Layer 16 / representation	$0.7247_{\pm 0.0056}$	$0.7150_{\pm 0.0045}$	$0.6991_{\pm 0.0027}$	$0.6711_{\pm 0.0044}$	$0.6022_{\pm 0.0044}$
Compressed concatenation / beliefs	$\mathbf{0.7407}_{\pm 0.0030}$	$\mathbf{0.7284}_{\pm 0.0021}$	$\mathbf{0.7126}_{\pm 0.0025}$	$\mathbf{0.6774}_{\pm 0.0040}$	$0.6025_{\pm 0.0015}$
Compressed concatenation / representation	$0.7329_{\pm 0.0019}$	$0.7218_{\pm 0.0021}$	$0.7034_{\pm 0.0022}$	$0.6766_{\pm 0.0016}$	$\mathbf{0.6051}_{\pm 0.0004}$

D Controller Parameters and Computational Cost

We report controller parameters, inference arithmetic, and estimated training resources. FLOP estimates count a multiply–accumulate as two operations and omit normalization, activations, and embedding lookups.

D.1 Inference arithmetic

Reasoner and controller inference. Let L be the number of reasoner blocks, d_q the total query-head width, d_{kv} the total key/value-head width, and P_{blk} the parameters per block. Grouped-query attention reduces key/value storage while retaining attention for every query head. For Qwen3, $L = 28$, $d_q = 2{,}048$, $d_{\text{kv}} = 1{,}024$, and $P_{\text{blk}} \approx 50.3$M. With $P_{\text{LLM}} = LP_{\text{blk}}$ and an output

embedding of approximately 0.31B parameters, the approximate cost of a decode step at context length t is

$$\mathcal{F}_{\mathrm{LLM}}(t) \simeq 2(P_{\mathrm{LLM}} + P_{\mathrm{emb}}) + 4Ld_q t \simeq 3.44 \text{ GFLOPs} + 2.29 \times 10^5 t \text{ FLOPs}. \tag{9}$$

The key/value cache has $2Ld_{\mathrm{kv}} = 57{,}344$ entries per token; bytes depend on precision.

The BELT probe uses cached layer-16 states, width $d_{\mathrm{p}} = 768$, and $L_{\mathrm{p}} = 2$ causal Transformer layers. Of its $P_{\mathrm{p}} = 15{,}957{,}765$ parameters, $P_{\mathrm{mat}} = 15{,}335{,}424$ participate in token-wise matrix multiplications; boundary readouts and the unused inherited classifier are excluded. With incremental attention over cached history, the leading per-token cost is

$$\mathcal{F}_{\mathrm{BELT}}(t) \simeq 2P_{\mathrm{mat}} + 4L_{\mathrm{p}}d_{\mathrm{p}}t \simeq 3.07 \times 10^7 + 6.14 \times 10^3 t \text{ FLOPs}. \tag{10}$$

The weight term is about 0.89% of the reasoner's and the attention term about 2.68%. The BELT probe stores $2L_{\mathrm{p}}d_{\mathrm{p}} = 3{,}072$ cache entries per token, or 5.36% of the reasoner's entry count. Four belief readouts add about $6{,}144$ FLOPs per boundary. The BELT-B/H policy adds a 5/768-dimensional dot product and an intercept at each boundary.

OSP adds two reasoner blocks and a stopping head; the original blocks still serve generation. Its approximate per-token cost is $2 \cdot 2P_{\mathrm{blk}} + 4 \cdot 2d_q t \simeq 2.01 \times 10^8 + 1.64 \times 10^4 t$ FLOPs. Both its attention term and its $4d_{\mathrm{kv}} = 4{,}096$ cache entries are $2/28 = 7.14\%$ of the corresponding reasoner quantities. Each request executes only its selected checkpoint.

The dense-attention counter gives 85.5M FLOPs per banked token for the BELT probe and 347.6M for OSP, a ratio of approximately 4.1. This counts full-sequence attention, whereas Eqs. (9) and (10) approximate causal token-by-token arithmetic.

D.2 Estimated training resources

Under the recorded schedules on one RTX 4090, controller training is estimated at 10.0 GPU-hours for three BELT epochs and 50.9 GPU-hours for one seven-epoch OSP run. Five OSP runs, one per token price, give an estimated 254.5 GPU-hours (Table 5). The corresponding ratios are approximately 5.1 and 25.5. The BELT estimate extrapolates its first epoch to three epochs; the five-checkpoint OSP estimate assumes each run costs as much as the recorded $\lambda = 10^{-5}$ run. Both include validation and exclude shared trajectory generation, feature export, and CPU policy fitting.

The estimates cover 31,248 training and 2,554 validation trajectories, with batch sizes 8 for BELT and 3 for OSP. Frozen reasoner states are recomputed each epoch, with reuse only within a bounded store. Permanently caching these states would change the training-cost comparison by removing repeated frozen-prefix computation after initial extraction.

D.3 Preference adaptation and parameter storage

BELT uses one belief training run, one feature export, and one CPU policy-fitting job per token price. For K deployed policies, its parameter storage is $P_{\mathrm{p}} + KP_{\mathrm{policy}}$, where $P_{\mathrm{policy}} = 6$ or 769 for BELT-B/H, compared with KP_{OSP} for the full OSP family. These inventories exclude the shared reasoner and standardization statistics. Each request uses only its selected policy or checkpoint; OSP-1 uses one checkpoint with varied thresholds.

Table 59: Trainable controller parameters, excluding the frozen reasoner. OSP counts one two-block branch and its stopping head.

Backbone	BELT	OSP
Qwen3	15,957,765	100,676,098
OLMo3	19,103,494	404,791,298

E Additional Comparisons and Protocol Checks

This appendix checks the reading protocol, baseline implementations and evaluation sets used in §§ 5 and 6.

Table 60: Split-half sensitivity of test-frontier readings. n: members per seed; med.: median shrinkage; max$|\cdot|$: largest absolute shrinkage; SD: median cross-reading SD, pooled over splits and seeds for LiveCodeBench.

Method	Qwen3 ID/OOD (2,719)					OLMo3 OOD (1,025)					LiveCodeBench (549 + 416)				
	Probe Params	n	med.	max$\lvert\cdot\rvert$	SD	Probe Params	n	med.	max$\lvert\cdot\rvert$	SD	Probe Params	n	med.	max$\lvert\cdot\rvert$	SD
Natural	–	1	+0.000	0.00	0.01	–	–	–	–	–	–	–	–	–	–
Fixed cap	0	5	+0.000	1.04	1.04	0	8	+0.002	1.20	0.66	–	–	–	–	–
CoDE-Stop	0	225	+0.001	1.43	1.09	0	75	+0.001	0.79	0.65	–	–	–	–	–
DEER	0	54	+0.001	1.30	1.06	0	18	+0.000	0.99	0.61	–	–	–	–	–
TC	514	300	+0.003	0.31	0.99	514	150	+0.003	1.11	1.13	–	–	–	–	–
ROM	5M	21	+0.002	1.40	1.03	8M	21	+0.002	1.68	1.27	–	–	–	–	–
OSP	101M $\times 5$	5	+0.001	0.76	1.03	–	–	–	–	–	101M $\times 5$	5	+0.036	11.52	2.42
OSP-1	101M	13	+0.000	0.57	1.04	405M	13	+0.002	5.11	1.02	101M	13	+0.006	1.41	2.45
BELT-B-1	–	–	–	–	–	19M	13	+0.001	0.17	0.95	–	–	–	–	–
BELT-H-1	–	–	–	–	–	19M	13	+0.001	0.20	0.99	–	–	–	–	–
BELT-B	16M	17	+0.001	0.45	1.03	19M	221	+0.002	0.99	0.96	16M	17	+0.007	0.88	2.10
BELT-H	16M	17	+0.000	0.53	1.02	19M	221	+0.003	0.94	0.97	16M	17	+0.021	6.67	2.26

Table 61: Token saving, accuracy and utility comparing predictive warm-up and direct utility training on the OOD test frontier. Differences pair the same training seed; BELT references report mean and SD.

Seed	Recipe	Near-never-stop checkpoints	Saving (%) ↑ at 1 pp	Acc. (%) ↑ at 0.5 cost	Acc. (%) ↑ at 0.3 cost	Utility ↑
0	Direct	4/4	1.09	33.32	24.65	0.557
0	Belief warm-up → direct	0/4	67.27	69.37	63.44	0.645
1	Direct	0/2	68.35	68.46	64.32	0.638
1	Belief warm-up → direct	0/2	68.24	69.97	63.11	0.651
2	Direct	1/1	8.27	57.53	46.15	0.557
2	Belief warm-up → direct	1/1	0.94	24.83	20.31	0.557
0	Warm-up − direct	–	+66.18	+36.05	+38.79	**+0.089**
1	Warm-up − direct	–	-0.11	+1.51	-1.21	<u>+0.013</u>
2	Warm-up − direct	–	-7.34	-32.69	-25.85	0.000
–	Natural completion	–	0.00	–	–	0.557
–	BELT-B(three probe seeds)	–	67.56 ± 0.41	69.61 ± 0.24	63.15 ± 1.31	<u>0.654</u> ± 0.002
–	BELT-H(three probe seeds)	–	70.53 ± 0.29	70.05 ± 0.28	64.95 ± 0.18	**0.656** ± 0.001

E.1 Stability of test-frontier readings

Test-frontier operating points select frontier members and mixture weights on the test questions (§ A.7). To measure the resulting selection optimism, we draw 200 stratified half splits of each test population. For every (method, training run), we select each operating point on one half and evaluate it on the other, then reverse the halves. Shrinkage is the whole-population reading minus the mean of these cross-readings. Its units are percentage points for tolerance and budget readings, and utility for price readings. Table 60 summarizes all reported scopes and levels. For Qwen3 tolerance readings, median shrinkage is 0.04 (OSP), 0.06 (BELT-B), 0.07 (BELT-H) and 0.09 pp (OSP-1). Price readings shrink by less than 0.001 utility. The largest absolute shrinkage among Qwen3 readings is 0.53 pp for BELT-H and 0.76 pp for OSP. On OLMo3 the largest shrinkage stays below 1.0 pp for BELT-B/H against 5.11 pp for OSP-1. LiveCodeBench readings are less stable for every family. OSP has the largest shrinkage (11.52 pp); with five members, its tolerance readings can depend strongly on a single member. For the Qwen3 OSP and BELT families, the reported accuracy and saving readings shrink by less than one percentage point.

E.2 Direct utility training of the BELT probe

To separate belief training from architecture, we keep the reasoner frozen and train the BELT probe with a linear stopping head directly on the utility in Eq. (2), following OS-Pruner. We use three

training seeds, anneal the token price to $\lambda = 2 \times 10^{-5}$ by epoch 4, and stop at epoch 7. A second variant first trains the probe on *Now* and *End* labels, then applies the same direct training. Table 61 reads every retained checkpoint, with its native stochastic policy and 13 source-validation logit-quantile thresholds, on the OOD roster under the test-frontier protocol of § A.7 (micro average; natural completion 65.73% accuracy at 5,022 tokens). A checkpoint is near-never-stop when its native policy saves less than 0.5% of natural tokens.

Direct training reaches BELT-level operating points in one seed and collapses to near-never-stop in two of three seeds. Seed 1 saves 68.35% at the 1 pp tolerance with test-family-max utility 0.638, below the three-seed BELT-B/H means of 0.654/0.656 (micro 67.56/70.53% saving). Every retained checkpoint of seeds 0 and 2 is near-never-stop, with test-family-max utility equal to natural completion (0.557). The *Now*/*End* warm-up avoids collapse for seed 0 (utility 0.645) but not seed 2. A warm start therefore changes some outcomes without eliminating the failure mode. Freezing the belief-trained probe and fitting only the linear policy yields dev-selected utility above natural completion for every probe seed and every reported price (Table 37). These runs document the outcome of one recipe with three seeds; they share the evaluation rule with BELT but not the candidate families.

Gradient attenuation in direct training. Consider one trajectory's unregularized return $R_\lambda = \sum_{i=1}^{T} \rho_i u_i$ with $u_i = y_i - \lambda C_i$, a stop logit z_k with $\pi_k = \sigma(z_k)$ at a candidate boundary $k < T$, and W_{k+1} the expected utility conditional on continuing past k. Holding the other logits fixed, only $S_k\{\pi_k u_k + (1-\pi_k)W_{k+1}\}$ depends on z_k; S_k and W_{k+1} are fixed. Thus

$$\frac{\partial R_\lambda}{\partial z_k} = S_k\, \pi_k(1-\pi_k)\,(u_k - W_{k+1}). \tag{11}$$

The factor $\pi_k(1-\pi_k)$ vanishes when a logit saturates in either direction. When the policy never stops, every π_k is near zero; when it stops at the first boundary, π_1 is near one and S_k is near zero for every later boundary. At either extreme, bounded utility gaps imply small logit gradients, which also attenuate the signal propagated to probe parameters. A high initial price favors stopping at the first boundary; the near-never-stop endpoints in Table 61 illustrate the opposite extreme. Belief training avoids this dependence on stopping probabilities. Its cross-entropy gradient at a labelled boundary does not depend on the policy's reach probability S_k, so every labelled position contributes in Eq. (4). We then freeze the probe and fit a linear policy from a bounded start (Table 9). This gradient analysis identifies attenuation near saturated policies as a possible mechanism for the observed training difficulty.

E.3 Code transfer

Both controllers read the same frozen Qwen3 trajectories and therefore share any overlap between LiveCodeBench Pre and the reasoner's pretraining data. We complement this shared-backbone comparison with Post, which postdates the stated cutoff (§ B.3). Threshold expansion gives OSP 13 inference-time thresholds per checkpoint, the same number BELT receives per policy. Expanded OSP reaches 40.11% token saving at the 1 pp tolerance on Pre, below expanded BELT-H's 42.55% and expanded BELT-B's 46.14% (§ B.6). Both BELT interfaces retain their lead after expansion; § B.6 specifies the candidate counts for each family. On Post, OSP leads at the 1 and 2 pp tolerances and at the 70% and 60% budgets, but BELT remains competitive: both interfaces lead at the 40% budget, and BELT-B has higher dev-selected utility at the lowest price. These results use one 16M BELT probe, compared with five 101M OSP checkpoints, preserving BELT's parameter and controller-training advantages (§ D). The large spread of OSP's Pre readings at the 0 pp tolerance (12.5 ± 21.6, Table 3) matches its split-half shrinkage in Table 60: with five members, whether the costliest checkpoint clears the tolerance line decides the reading for that training run.

E.4 OS-Pruner reproduction

With no public code available at the time of reproduction, we independently implemented OS-Pruner, strictly following its published architecture and objective, with adaptations to our shared trajectory, grading and total-token protocol (§ A). Its strong performance supports the quality of this reproduction: OSP is the strongest baseline in every Qwen3 table, ahead of TC, ROM and the training-free rules by 15 pp or more of token saving at the 1 pp tolerance (Table 1). Full OSP is also more accurate

than OSP-1 at every shared budget where both are feasible (Table 2), consistent with the benefit of per-price training. OSP checkpoints maximize validation utility after price annealing, and OSP also receives the same 13-threshold expansion as BELT (§ B.6).

E.5 Comparator on OLMo3

Full OSP on OLMo3 would require five 405M-parameter checkpoints per training run (Table 59), fifteen for three runs. We therefore report OSP-1, the strongest single-checkpoint utility-trained controller on Qwen3: it is second only to full OSP at the 1 and 2 pp OOD tolerances (Table 1) and the strongest method after the OSP and BELT families at every shared budget (Table 2). OLMo3 BELT-B omits prefix length (§ A.9); Table 3 reports both interfaces.

E.6 Evaluation sets

OOD denotes benchmarks that are not among the three training sources, not a harder distribution: OOD natural accuracy (60.6%) exceeds ID (48.9%, Table 2) because the ID sets draw from competition-level subsets of NuminaMath, DAPO-Math and DeepScaleR. Overlap between training and evaluation questions was screened by normalized-text deduplication and by exact and 13-word overlap against recorded external references; the screen removed nine AIME repetitions, and the retained roster is fixed by a versioned question registry (§ A.1). To complement lexical filtering with semantic screening, we embedded all 4,241 training-bank questions and all 1,025 OOD questions with a sentence encoder (`bge-base-en-v1.5`, cross-checked with `all-MiniLM-L6-v2`) and inspected the 30 highest-cosine training neighbours by hand. Two OOD questions repeat a DAPO-Math training problem with different LaTeX formatting (one Gaokao and one AMC 2023 item; cosine 0.97 and 0.96). One MATH-500 question shares a template with a DeepScaleR question but has a different polynomial and answer. Every other pair down to cosine 0.88 contains different problems sharing a topic or definition. No OOD question reaches cosine 0.98, and no AIME question has a training neighbour above 0.90. The two identified overlaps are retained in the fixed roster used by every method. All reported comparisons use this common integer-answer evaluation population.

E.7 *Later* and *Same* as training targets

Later and *Same* are partially substitutable: *End–Later* correlations are 0.94–0.95 and *Now–Same* 0.78–0.95 (§ C.4). Adding *Later* to *Now*/*End* raises mean utility at all five reported prices (up to +0.022 at $\lambda = 5 \times 10^{-5}$); adding *Same* raises it at all five (up to +0.021). The four-target recipe has the highest mean utility at four of five prices (Table 47). The small loss from removing *Later* as a policy input (§ 6.2) is consistent with redundancy when *Same* is present. *Later* improves mean control utility across all five prices despite lower *Now* AP (Table 57).

F Future Work

The reusable-probe framework supports three directions for further study. First, extending the banked-trajectory evaluation to a serving stack, with one forced-answer generation after stopping, would quantify end-to-end latency and live decoding behavior. Second, the same training approach could incorporate a belief about remaining computation cost, as in ZIP-RC (Manvi et al., 2026), alongside the four answer-outcome beliefs. Third, symbolic-answer and open-ended tasks would extend the evaluation beyond integer-answer mathematics and LiveCodeBench.