CALIBER: CALIBRATED SYSTEM-ONE ROUTING FOR ADAPTIVE SYSTEM-TWO REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning models improve performance on difficult tasks but incur substantial latency and computational cost when invoked indiscriminately. We study whether a fast calibrated decision model can learn when deliberative reasoning is actually necessary. CALIBER introduces a System-One/System-Two architecture in which a lightweight decision model observes the question and one cheap System One pass and selects among a direct System One answer from a 4B model, a direct answer from a 27B model and extended reasoning at either scale, and can defer when no action is likely to succeed. The router returns a calibrated probability of success for every action rather than an unconstrained textual plan, enabling explicit risk-sensitive policies that trade expected success against cost. We train the router on the realized outcomes of executing every route on every question, and compare it with confidence cascades, a prompted language-model router, conventional classifiers, a binary win classifier, random routing and an outcome oracle. Experiments on MMLU-Pro and SuperGPQA measure task success, calibration, risk–coverage trade-offs, routing regret and GPU cost, both in distribution and under held-out-subject and cross-benchmark shift. The central hypothesis is that calibrated decision routing can preserve most of the capability of expensive reasoning systems while substantially reducing their invocation rate. The hypothesis holds only in part. At half the cost of always reasoning with the 27B model, CALIBER retains 83% of that route's accuracy gain over the System One answer, short of the pre-specified 90% target, while invoking System Two on 41% of questions. In the pre-specified comparison under the same budget cap it is 2.9 points more accurate than the best confidence cascade in distribution (81.5% versus 78.6%; 95% interval [1.3, 4.5] points), and about 2 points at matched realized cost. The gain comes from per-action success heads over a text encoder for the large direct and large reasoning actions, not from temperature scaling, System One confidence features or the small reasoning action, which lowers accuracy under this policy. Under held-out-subject and cross-benchmark shift, reasoning becomes longer and operating points transferred from the source calibration split miss their budgets; refitting temperatures and operating points on a small target calibration set restores budget adherence, with no significant difference from the cascade.

1 INTRODUCTION

Reasoning models that generate long chains of thought before answering solve problems that direct answering cannot, but they pay for it at every query. A reasoning trace can reach thousands of tokens, so a system that always reasons spends most of its compute on questions that a small model would have answered correctly in one forward pass. Test-time compute is most useful when it is allocated by difficulty (Snell et al., 2025; Damani et al., 2025), and a deployed system has to make that allocation before it knows the answer.

We frame the allocation as a dual-process decomposition. A fast *System One* layer reads the question once and returns a distribution over answers; a slow *System Two* layer reasons step by step before it answers. Between them sits a decision: answer now, answer with a larger model, reason, or decline to answer. We treat this decision itself as a System One task. From the question and one cheap System One pass, a router returns an estimated probability of success for every available action before any expensive action is executed, and a policy chooses by expected utility, trading predicted

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.

success against cost. Probabilities rather than a free-text plan make the policy explicit: the cost weight, the budget and the deferral threshold are parameters that can be set and audited.

The difficulty is that which action suffices varies from question to question. A confidence cascade escalates along a fixed order, from the cheapest model to the most expensive, whenever the current model's confidence falls below a threshold. Such cascades are strong baselines that learned deferral rules often fail to beat (Jitkrittum et al., 2023; Li et al., 2026). When actions differ in both scale and mode, they may still be ordered on average in cost and accuracy, yet their per-question outcomes are not nested: a small model that reasons can be right where a large model answering directly is wrong, and the reverse. A fixed escalation order cannot exploit this per-question non-monotonicity. Analysis of cascade deferral indicates that learned rules help mainly when downstream models behave like specialists (Jitkrittum et al., 2023), which is the situation such an action set creates. A router that estimates the success of each action separately can choose "reason with the small model" on the questions where that is enough and "answer with the large model" on others, without visiting the actions in a fixed order. For the per-action estimates to be traded against costs, and for a budget set on one distribution to hold on another, they should also be calibrated.

We study this question in a closed-book multiple-choice setting where every action can be executed and scored on every question. The actions are a direct answer from Qwen3-4B (the System One answer), a direct answer from Qwen3.8-27B, extended reasoning with either model; deferral is available as a selective-prediction option. We execute all four answering routes on 15,875 questions from MMLU-Pro (Wang et al., 2024) and SuperGPQA (M-A-P Team et al., 2025), record correctness and GPU cost, and train the router on these realized outcomes. Our contributions are:

- **Routing as per-action success estimation.** CALIBER predicts, before any expensive generation, a temperature-calibrated probability of success for each action in a scale$\times$mode action set, and chooses by expected utility with a cost weight selected on a calibration split for each target budget. Deferral is a selective-prediction threshold on the predicted success of the chosen action.
- **An execute-every-route evaluation.** Four routes are executed on every question, which gives an outcome oracle, per-question cost estimates and a matched-budget comparison with confidence cascades, a logistic classifier on System One confidence, a binary win classifier, a prompted self-assessment router and random routing. Two attribution controls, CALIBER without the small-reasoning action and a confidence cascade through it, separate calibrated per-action estimation from access to that action.
- **Pre-specified hypotheses and shift tests.** We state in advance a capability-retention target (H1), a confirmatory comparison with the better confidence cascade and its decision rule (H2), and a descriptive test of calibration transfer under held-out-subject and cross-benchmark shift with and without target recalibration (H3).

On MMLU-Pro, the System Two route (Qwen3.8-27B reasoning) is 45 points more accurate than the System One answer at about 600 times its GPU cost. At half the cost of always reasoning, CALIBER retains 83% of that gain, so H1 (90%) is not met. Under the same budget cap it is 2.9 points more accurate than the best confidence cascade (95% interval [1.3, 4.5]), which confirms H2 in distribution; at matched realized cost the descriptive difference is about 2 points. The attribution controls locate the gain in text-encoder success heads for the large direct and large reasoning actions: temperature scaling and the System One features leave accuracy unchanged, per-action logistic regression on System One confidence does not beat the cascade, and dropping the small reasoning action raises accuracy further. Calibration (ECE 0.03–0.06 after scaling) matters for the reported probabilities and for budget adherence rather than for in-distribution accuracy. Under shift, the in-distribution advantage does not survive: transferred operating points overspend because reasoning traces grow longer, and after target recalibration there is no significant difference between CALIBER and the cascade.

2 RELATED WORK

Routing and cascades. Cascades query models in order of cost and stop when an acceptance rule is met (Chen et al., 2024a); acceptance can come from self-verification (Aggarwal et al., 2024), answer consistency across samples (Yue et al., 2024) or learned deferral on token-level uncertainty (Gupta et al., 2024). Maximum-probability deferral is hard to beat; learned rules help mainly with

specialist downstream models, label noise and shift (Jitkrittum et al., 2023). Routers choose a model before generation. They have been trained as per-model correctness predictors on benchmark outcomes (Shnitzer et al., 2024), quality-gap encoders (Ding et al., 2024), preference-trained win classifiers (Ong et al., 2025), contrastive encoders (Chen et al., 2024b) and confidence-token models (Chuang et al., 2025), and extended to unseen model pools (Jitkrittum et al., 2026). Theory shows that optimal routing and cascading reduce to accurate quality estimates per model (Dekoninck et al., 2025), and that plug-in accuracy-and-cost estimators are rate optimal (Somerstep et al., 2025). Executing every model on every query to obtain an outcome matrix and an oracle is standard (Hu et al., 2024); under unified evaluation many routers do not beat simple baselines and a large oracle gap remains (Li et al., 2026). Abstention in cascades (Zellinger et al., 2025) and calibrating each model's own confidence within a cascade (Zhang et al., 2026b) have also been studied, as has the observation that escalation can break a correct small-model answer (Wang et al., 2026). CALIBER uses this plug-in design and this execute-everything protocol; it does not introduce either.

When to think. Several methods let a single model decide whether to reason, by reinforcement learning inside the model (Fang et al., 2025; Zhang et al., 2025; Tu et al., 2025; Lou et al., 2025), by an external switch trained on the realized success of each mode (Liang et al., 2025), by pre-generation router heads (Guo et al., 2026), by draft agreement without training (Lee et al., 2026), or with distributionally robust routing under shift (Zhang et al., 2026a). Route to Reason (Pan et al., 2026) already routes over a product space of models and reasoning strategies, predicting accuracy and token cost from executed outcomes; it is our nearest neighbour. Compute-optimal test-time scaling (Snell et al., 2025) and input-adaptive allocation (Damani et al., 2025) motivate trading model scale against reasoning.

Selective prediction and calibration. Deferral builds on selective classification (Geifman & El-Yaniv, 2017) and learning to defer (Mozannar & Sontag, 2020); we use a plug-in rule rather than a surrogate loss, so costs and thresholds can change without retraining. Temperature scaling (Guo et al., 2017) is our recalibration method. Multiple-choice letter probabilities are informative about correctness (Kadavath et al., 2022) but miscalibrated (Plaut et al., 2025), which is why a confidence cascade can rank questions well and still need calibration.

Position. Each ingredient of CALIBER has close prior work; CALIBER is an integration and measurement contribution. Relative to Route to Reason, the differences are calibration as a target and a metric, deferral, retention and oracle-gap reporting against a tuned confidence cascade, and recalibration under shift. Relative to calibration-aware cascades (Zhang et al., 2026b), CALIBER predicts success for actions that have not been executed rather than calibrating each model's own confidence after it has answered.

3 CALIBER

Setup. A question x has $K \leq 10$ displayed options and a gold option y. The answering actions are $\mathcal{A} = \{\texttt{s1}, \texttt{l1}, \texttt{r4}, \texttt{r27}\}$: a direct answer from Qwen3-4B (`s1`, the System One answer), a direct answer from Qwen3.8-27B (`l1`), and extended reasoning with Qwen3-4B (`r4`) or Qwen3.8-27B (`r27`, System Two). Each action returns a distribution $\pi_a(\cdot \mid x)$ over the K letters, predicts $\hat{y}_a(x) = \arg\max_k \pi_a(k \mid x)$ and succeeds with $S_a(x) = \mathbb{1}[\hat{y}_a(x) = y]$; any failure to produce a readout counts as $S_a(x) = 0$. Executing action a alone costs an estimated $g_a(x)$ GPU-seconds (Section 4). Every policy first pays for the System One pass, because the router reads its output, and for its own decision stage, so the realized cost of a policy π that chooses a is

$$c_a^\pi(x) = r_\pi(x) + g_{\texttt{s1}}(x) + \mathbb{1}[a \neq \texttt{s1}]\, g_a(x), \tag{1}$$

where $r_\pi(x)$ is the decision cost: the measured batched inference cost per question of a learned router (tokenization included), the prompt pass of a prompted router, and zero for rules that reuse probabilities already computed. `defer` returns no answer; it is not a fifth scored action but a selective-prediction threshold applied after the choice.

Per-action success estimation. The router input is the question text x together with a feature vector $\phi(x)$ of the System One distribution $\pi_{\texttt{s1}}(\cdot \mid x)$: its maximum probability, the margin between the top two probabilities, normalized entropy, the three largest probabilities, the number of options

and the prompt length. A text encoder h (ModernBERT-base, Warner et al., 2025) produces a mean-pooled representation, and a two-layer MLP maps $[h(x); \phi(x)]$ to one logit $f_a(x)$ per action. The heads are trained jointly, each with binary cross-entropy between $\sigma(f_a(x))$ and the executed outcome $S_a(x)$. The four targets are not mutually exclusive, so the heads are independent sigmoids rather than a softmax over actions.

Calibration. Each head receives its own temperature, fitted on a calibration split $\mathcal{D}_{\text{cal}}$ by minimizing negative log-likelihood,

$$\tilde{p}_a(x) = \sigma\big(f_a(x)/T_a\big), \qquad T_a = \arg\min_{T} \sum_{x \in \mathcal{D}_{\text{cal}}} \ell_{\text{BCE}}\big(\sigma(f_a(x)/T), S_a(x)\big). \tag{2}$$

Per-head temperatures matter because the policy below compares probabilities across heads: if one head is overconfident relative to another, the argmax favours its action for reasons unrelated to success.

Expected-utility policy and budgets. Given calibrated estimates, the policy maximizes predicted success minus a cost penalty,

$$a^*_\lambda(x) = \arg\max_{a \in \mathcal{A}} \; \tilde{p}_a(x) - \lambda\, \bar{c}_a, \tag{3}$$

where $\bar{c}_a$ is the mean realized cost of action a on $\mathcal{D}_{\text{cal}}$ and $\lambda \geq 0$ is the cost weight. The cost that is not yet known for a question is replaced by its calibration mean, so the policy needs no cost model; the router's own cost is paid for every action and does not affect the argmax. A budget is a fraction β of the mean cost of always reasoning with the large model on $\mathcal{D}_{\text{cal}}$, $B_\beta = \beta\, \bar{g}_{\texttt{r27}}$. For each β we pick, on $\mathcal{D}_{\text{cal}}$, the cost weight with the highest accuracy among those whose mean realized cost does not exceed B_β (ties go to the lower cost; if no weight is feasible, the cheapest one is used). The temperatures and λ_β are then applied unchanged to test data; all baselines select their operating points by the same rule.

Deferral. The policy answers only when the calibrated success probability of its chosen action clears a threshold, $\tilde{p}_{a^*}(x) \geq \tau$, and otherwise defers; sweeping τ traces a risk–coverage curve (Geifman & El-Yaniv, 2017). Deferral is not used in the budgeted comparisons.

Baselines. Let $m_a(x) = \max_k \pi_a(k \mid x)$ denote the maximum letter probability of action a. All thresholds and cost weights are selected on $\mathcal{D}_{\text{cal}}$ by the budget rule above, and every policy is charged its decision cost r_π.

- **Two-stage cascade (cascade2):** answer with `s1`; escalate to `r27` if $m_{\texttt{s1}}(x) < \tau$. Cost $g_{\texttt{s1}} + \mathbb{1}[m_{\texttt{s1}} < \tau]\, g_{\texttt{r27}}$.
- **Three-stage cascade (cascade3):** escalate from `s1` to `l1` if $m_{\texttt{s1}}(x) < \tau_1$, and from `l1` to `r27` if also $m_{\texttt{l1}}(x) < \tau_2$. Costs of every visited stage accumulate.
- **Logistic (conventional classifier):** one logistic regression per action on $\phi(x)$ alone, temperature-scaled, with the same expected-utility policy; it reuses the System One features on CPU and is charged $r_\pi = 0$.
- **Binary win classifier:** the CALIBER architecture with one head trained on $\mathbb{1}[S_{\texttt{r27}}(x) = 1 \wedge S_{\texttt{s1}}(x) = 0]$, in the style of RouteLLM (Ong et al., 2025); route to `r27` when the score exceeds τ, else answer with `s1`.
- **Self-assessment (prompted LLM router):** Qwen3-4B is asked whether it can answer directly or needs step-by-step reasoning (Appendix A); route to `r27` when the probability of "needs reasoning" exceeds τ. Its prompt pass is charged as its decision cost.
- **Random:** route to `r27` with probability $\rho = \text{clip}\big((B_\beta - \bar{g}_{\texttt{s1}})/\bar{g}_{\texttt{r27}}, 0, 1\big)$, else `s1`; accuracy and cost are computed in expectation.
- **Oracle:** for each question, the correct action with the lowest realized cost, or `s1` if none is correct; it ignores the budget.

Attribution controls. CALIBER gains over cascade2 and cascade3 could come from calibrated per-action estimation or simply from access to `r4`, which neither cascade uses. CALIBER *without `r4`*

restricts the argmax to $\{\texttt{s1}, \texttt{l1}, \texttt{r27}\}$, the action set of cascade3, with the same heads and temperatures. The *reasoning cascade (cascade_r4)* escalates from `s1` to `r4` if $m_{\texttt{s1}}(x) < \tau_1$ and from `r4` to `r27` if also $m_{\texttt{r4}}(x) < \tau_2$, with costs accumulating. *Ablations. Text only* removes $\phi(x)$. *Raw* uses $T_a = 1$: an uncalibrated per-action success estimator, comparable in kind to the per-pair accuracy predictors of Route to Reason (Pan et al., 2026).

4 Experimental setup

Tasks and roles. All roles were fixed with seed 17 before any route outcome was observed. From the MMLU-Pro test set (12,032 questions, 14 subjects, up to 10 options; Wang et al., 2024), four subjects are held out for task-level shift (economics, engineering, health, law), split 20/80 into shift_cal and shift_test. The remaining ten subjects are split per subject 60/20/20 into router_train, router_cal and id_test. For cross-benchmark shift we draw a seed-17 sample of 4,000 SuperGPQA questions (M-A-P Team et al., 2025) from those whose normalized stem does not occur in MMLU-Pro, split into xb_cal (1,000) and xb_test (3,000). When a normalized question stem occurs in more than one role, only its member with the smallest identifier is kept, which removes 157 questions. The final roles are router_train 4,947, router_cal 1,646, id_test 1,643, shift_cal 702, shift_test 2,937, xb_cal 1,000 and xb_test 3,000 (15,875 questions). Routers are trained on router_train only.

Routes and readout. Qwen3-4B and Qwen3.8-27B (Yang et al., 2025; Qwen Team, 2026) answer through their chat templates with thinking disabled (`s1`, `l1`) or enabled (`r4`, `r27`). Direct routes read out one forward pass: the probability of each displayed letter after the prefix "The answer is (", renormalized over the K letters. Reasoning routes sample a reasoning trace (temperature 0.6, top-p 0.95, top-k 20, with a sampling seed fixed by the question identity) up to a budget of 16,384 tokens, close the reasoning block, and apply the same letter readout. A trace that reaches the budget is closed, read out and flagged as truncated. The budget was fixed from a pilot on training questions so that truncation is rare; pilot outputs are not reported. All routes are executed on all 15,875 questions with vLLM in bf16, data parallel over four GPUs with questions assigned to shards round-robin. Prompts and settings are in Appendices A and B.

Costs. All costs are amortized GPU-time estimates under the batching we recorded. For direct routes, $g_a(x)$ is the wall time of a batch divided by the number of questions in it; for reasoning routes, the batch wall time is apportioned to questions in proportion to their generated tokens. Decision costs are estimated the same way (Section 3, averaged over router seeds). Policy costs are therefore accounting estimates, not a measured speedup of a deployed routed workload. Budgets are $\beta \in \{0.25, 0.5, 0.75\}$ of the mean `r27` cost on the calibration split, and retention curves sweep β from 0.05 to 1.0.

Router training. CALIBER encodes the subject, question and options (up to 1,024 tokens) with ModernBERT-base, mean-pools, concatenates $\phi(x)$ and applies a 256-unit GELU MLP with four sigmoid outputs, trained with AdamW (at most four epochs, early stopping on a 10% hold-out of router_train) for three seeds (17, 29, 43); Appendix C lists all hyperparameters.

Metrics. We report accuracy, realized mean cost and its ratio to the target budget, the action mix and the System Two invocation rate (the fraction routed to `r4` or `r27`). *Capability retention*, $\mathrm{Ret} = (\mathrm{Acc} - \mathrm{Acc}_{\texttt{s1}})/(\mathrm{Acc}_{\texttt{r27}} - \mathrm{Acc}_{\texttt{s1}})$ on the same test split, is the fraction of the gain of always-`r27` over `s1` that a policy keeps. *Routing regret* is the accuracy gap to the oracle. Calibration is measured per action by expected calibration error (15 equal-width bins) and Brier score, before and after temperature scaling. For deferral we report selective accuracy at coverages from 50% to 100% with confidence $\tilde{p}_{a^*}$ at the 50% budget, against `s1` answers ranked by $m_{\texttt{s1}}$. For reasoning routes we report readout faithfulness: the fraction of outputs with a parseable stated final answer and its agreement with the letter readout.

Hypotheses. The following were specified before test evaluation; deferral is not used in H1 or H2. **H1** (central): at the 50% operating point chosen on router_cal, CALIBER retains at least 90% of the `r27` gain over `s1` on id_test, and its realized id_test cost is at most half the id_test cost of always-`r27`. H1 is judged on point estimates (seed-mean retention ≥ 0.90 and the cost condition).

Table 1: Route outcomes. Accuracy (%) on the three test splits; all other columns are on id_test: mean generated tokens, truncation at the token budget (%), rate of parseable stated answers and their agreement with the letter readout (%), estimated GPU-seconds per question. Oracle: per-question cheapest correct route.

Route	id_test	shift_test	xb_test	Tokens	Trunc.	Parse	Agree	GPU-s
`s1`	44.0	44.3	27.8	–	–	–	–	0.005
`l1`	59.1	64.1	38.4	–	–	–	–	0.035
`r4`	78.1	59.0	43.0	3,550	1.4	46.6	99.1	2.27
`r27`	89.1	79.4	62.8	1,900	2.6	97.4	100.0	3.08
Oracle	93.1	87.4	75.7	–	–	–	–	0.71

H2 (confirmatory): at the 50% budget, let Δ be the id_test accuracy of CALIBER minus that of the better confidence cascade (cascade2 or cascade3, whichever is more accurate on router_cal). CALIBER correctness is averaged over router seeds per question, and a two-sided 95% percentile interval is computed from 10,000 question-level paired bootstrap resamples; it covers question sampling only, and per-seed differences are reported descriptively. If the lower bound exceeds zero, CALIBER is better; if the upper bound is below zero, the cascade is better; otherwise there is no significant difference. We make no equivalence claim. Every outcome is reported. **Secondary** (descriptive): realized id_test cost with 95% bootstrap intervals, and accuracy at a matched realized cost of half the id_test always-`r27` cost by interpolation on each router's test accuracy–cost frontier; this and the smallest budget reaching 90% retention are test-selected estimates, not deployable operating points. **Attribution** (descriptive): CALIBER without `r4` versus cascade3, and CALIBER versus cascade_r4. **H3** (descriptive): under held-out-subject (shift_test) and cross-benchmark (xb_test) shift, calibration and budget adherence degrade when temperatures and operating points are transferred from router_cal, and refitting only the temperatures and operating points on the target calibration split (shift_cal or xb_cal) restores them; router weights are never refit. We report the measured changes with paired bootstrap intervals.

5 RESULTS

5.1 ROUTE OUTCOMES

The routes form a steep accuracy–cost ladder (Table 1). On id_test, `r27` has 45 points of headroom over `s1` (89.1% versus 44.0%), and costs span nearly three orders of magnitude, from 0.005 to 3.08 GPU-s per question (about 600 times). The ladder is not nested per question: `r4` is correct and `l1` wrong on 28.4% of id_test questions, and the reverse holds on 9.3%. The oracle reaches 93.1% at 0.71 GPU-s, 23% of the always-`r27` cost, and uses `r4` on only 3.3% of questions. Reasoning is longer on the shifted sets: `r27` generates 4,479 and 4,808 tokens on shift_test and xb_test, against 1,900 on id_test, and costs 7.27 and 7.80 GPU-s. Over all 15,875 questions, 7.3% of `r27` and 3.1% of `r4` traces reach the token budget. `r27` states a parseable final answer in 92.5% of outputs, all of which agree with the letter readout; `r4` does so in 48.3%, with 98.7% agreement. The GPU cost per generated `r27` token was about 8% higher on one of the four shards; with round-robin assignment this affects all routers alike.

5.2 MATCHED-BUDGET COMPARISON

H1 is not met. At the 50% operating point CALIBER reaches 81.5% on id_test and retains 0.83 of the `r27` gain over `s1` (Table 2), below the 0.90 target. The cost condition holds: its realized cost of 1.43 GPU-s (95% interval [1.30, 1.58]) is below the H1 limit of 1.54 GPU-s, half the id_test always-`r27` cost. It invokes System Two on 41% of questions (35% `r4`, 6% `r27`) and answers almost all others with `l1`. On the id_test grid of budget fractions (Figure 1), 90% retention is first reached at a budget fraction of 0.60; this figure is test-derived and descriptive.

H2: CALIBER is better in distribution. The cascade selected on router_cal is cascade3 (`s1`→`l1`→`r27`). Under the same 50% budget cap (1.61 GPU-s, half the router_cal mean cost of `r27`), CALIBER is 2.9 points more accurate (81.5% versus 78.6%; paired bootstrap 95% interval

Table 2: Matched-budget comparison on id_test. Operating points are selected on router_cal under budget caps of 0.80, 1.61 and 2.41 GPU-s (25%, 50% and 75% of the router_cal mean `r27` cost) and applied unchanged. Acc. (%), retention of the `r27` gain over `s1`, and realized cost in GPU-s per question (decision stage included); seed means. Always-`r27` costs 3.08 GPU-s on id_test.

	25% budget			50% budget			75% budget		
Router	Acc.	Ret.	Cost	Acc.	Ret.	Cost	Acc.	Ret.	Cost
CALIBER	70.3	0.58	0.63	81.5	0.83	1.43	87.3	0.96	2.15
text only	71.0	0.60	0.68	81.4	0.83	1.44	88.0	0.97	2.31
raw ($T_a = 1$)	70.1	0.58	0.63	81.8	0.84	1.35	87.8	0.97	2.21
without `r4`	70.6	0.59	0.52	84.5	0.90	1.38	88.3	0.98	2.31
cascade2	54.9	0.24	0.71	67.0	0.51	1.35	79.9	0.80	2.10
cascade3	67.6	0.52	0.55	78.6	0.77	1.29	86.6	0.94	2.16
cascade_r4	55.4	0.25	0.75	69.1	0.56	1.53	78.5	0.77	2.37
Logistic	68.8	0.55	0.78	78.1	0.76	1.43	84.6	0.90	2.04
Binary win	58.9	0.33	0.60	73.3	0.65	1.39	82.6	0.85	2.33
Self-assessment	51.4	0.16	0.63	65.5	0.48	1.49	77.7	0.75	2.22
Random	55.2	0.25	0.77	66.5	0.50	1.54	77.8	0.75	2.31
Oracle (no budget)	accuracy 93.1, retention 1.09, cost 0.71 GPU-s								

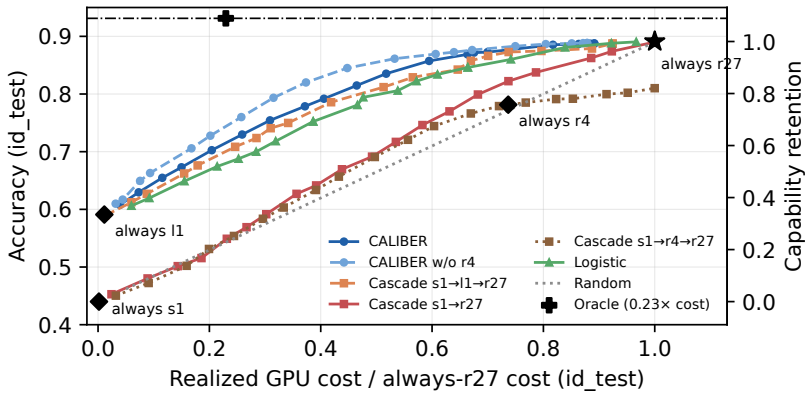

Figure 1: Accuracy (left axis) and capability retention (right axis) versus realized GPU cost on id_test, as a fraction of the id_test always-`r27` cost (3.08 GPU-s). Each point is an operating point selected on router_cal for one budget fraction (0.05 to 1.0 in steps of 0.05) and applied unchanged. Black markers: single routes; cross and dash-dotted line: the per-question oracle.

[1.3, 4.5] points), and each router seed gives +2.8 to +3.0 points. Realized costs differ: 1.43 [1.30, 1.58] GPU-s for CALIBER against 1.29 [1.09, 1.49] for the cascade. At a matched realized cost of 1.54 GPU-s, the descriptive, test-selected frontier estimate is 82.9% versus 80.9%, about 2.0 points. CALIBER is also ahead at the 25% budget, and the gap narrows to 0.6 points at 75% (descriptive). Routing regret at 50% is 11.6 points for CALIBER and 14.5 for cascade3.

Attribution and ablations. Access to `r4` is not the source of the gain. Restricted to the cascade3 actions {`s1`, `l1`, `r27`}, CALIBER reaches 84.5% at 1.38 GPU-s (retention 0.898), 5.9 points above cascade3 and 3.0 above full CALIBER, while the reasoning cascade cascade_r4 reaches only 69.1%. Full CALIBER sends 35% of questions to `r4`, which costs 74% as much as `r27` on id_test at 11 points lower accuracy; without `r4`, the router sends 44.5% of questions to `r27` instead. Temperature scaling and the System One features are not the source either: the uncalibrated (81.8%) and text-only (81.4%) variants match CALIBER. The advantage requires per-action success heads over the question and option text. Per-action logistic regression on the System One features alone, with the same policy, reaches 78.1%, no better than the cascade, and the other scalar-confidence routers trail further (binary win classifier 73.3%, cascade2 67.0%, self-assessment 65.5%, random 66.5%).

Table 3: Per-action calibration of CALIBER on id_test: ECE (15 bins) and Brier score, raw and temperature-scaled, seed-averaged.

	ECE		Brier	
Head	raw	scaled	raw	scaled
s1	0.068	0.058	0.216	0.215
l1	0.047	0.042	0.196	0.196
r4	0.070	0.049	0.154	0.152
r27	0.024	0.030	0.091	0.091

Table 4: Selective accuracy on id_test: CALIBER at the 50% budget (confidence $\tilde{p}_{a^*}$) versus s1 ranked by $m_{\texttt{s1}}$; accuracy (%).

Coverage	CALIBER	s1
80%	86.2	50.4
90%	84.1	47.0
100%	81.5	44.0

Table 5: Shift at the 50% budget. *Transfer*: temperatures and operating points from router_cal (budget 1.61 GPU-s). *Recal.*: refit on shift_cal or xb_cal (budgets 3.47 and 3.77 GPU-s); router weights unchanged. Acc. (%); Cost/B is realized cost over the budget in use; ECE is the mean over the four heads after scaling. The best cascade is cascade3 in all four settings.

		shift_test (held-out subjects)				xb_test (SuperGPQA)			
Router	Setting	Acc.	Ret.	Cost/B	ECE	Acc.	Ret.	Cost/B	ECE
CALIBER	Transfer	69.5	0.72	0.85	0.056	49.7	0.62	1.71	0.216
CALIBER	Recal.	75.1	0.88	0.96	0.042	53.2	0.73	1.06	0.097
Best cascade	Transfer	72.8	0.81	1.58	–	55.1	0.78	2.79	–
Best cascade	Recal.	75.1	0.88	1.04	–	53.0	0.72	1.05	–

5.3 CALIBRATION AND SELECTIVE PREDICTION

The per-action success probabilities are well calibrated in distribution (Table 3). Temperature scaling lowers ECE for three heads (s1 0.068 to 0.058, l1 0.047 to 0.042, r4 0.070 to 0.049) and raises it slightly for r27 (0.024 to 0.030), with Brier scores nearly unchanged; the fitted temperatures are close to one (seed means 1.19, 1.00, 0.85 and 1.04). Scaling helps calibration slightly but not routing accuracy (Section 5.2). Deferral trades coverage for accuracy (Table 4): answering only the 80% of questions with the highest $\tilde{p}_{a^*}$ raises CALIBER's accuracy from 81.5% to 86.2%, and 90% coverage gives 84.1%; System One answers ranked by their own confidence reach 50.4% and 47.0%.

5.4 DISTRIBUTION SHIFT

Budget adherence fails under transfer (Table 5). Operating points selected on router_cal overspend because reasoning is longer: cascade3 spends 1.58 and 2.79 times its budget on shift_test and xb_test, and CALIBER 0.85 and 1.71 times. The transferred comparisons are therefore not cost-matched. CALIBER is 3.3 points less accurate than cascade3 on shift_test (95% interval $[-4.4, -2.2]$) and 5.4 points on xb_test ($[-6.7, -4.1]$), but these differences mostly reflect budget miscalibration: at matched realized cost (half of each split's always-r27 cost) the descriptive frontier estimates are close, 75.3% versus 74.7% and 54.3% versus 52.5%. Calibration transfers across subjects but not across benchmarks: mean ECE is 0.056 on shift_test, near the 0.045 of id_test, but 0.216 on xb_test. Refitting temperatures and operating points on the target calibration split restores budget adherence (0.96–1.06 of the budget) and lowers mean ECE to 0.042 and 0.097. There is then no significant difference between CALIBER and cascade3 (-0.02 points, 95% interval $[-0.92, +0.84]$ on shift_test; $+0.21$ points, $[-1.01, +1.48]$ on xb_test); we make no equivalence claim. Recalibrated CALIBER almost never uses r4 (0.4% and 0% of questions) and coincides with CALIBER without r4. Recalibration does not uniformly restore per-action calibration: on xb_test the s1 ECE rises from 0.143 to 0.150, l1 remains high at 0.141, and the r4 temperature reaches the upper bound of its search range ($T = 20$) for every seed, which flattens that head rather than correcting it.

6 DISCUSSION

What the decomposition needs. The in-distribution result supports treating the routing decision as a System One task, provided that the fast layer predicts success per action rather than one scalar

confidence. The benefit comes from predicting, per question, whether the large direct answer and the large reasoning route will succeed, and that prediction needs a representation of the question: the same per-action policy on System One confidence features does no better than a confidence cascade. A scalar System One confidence ranks questions by how hard they are for the System One answer; it does not say which of several larger actions will succeed.

The small reasoning action. Per-question outcomes are not nested, since `r4` is right where `l1` is wrong on 28% of questions, yet adding `r4` lowers accuracy under the linear expected-utility policy. `r4` saves only 26% of the `r27` cost at 11 points lower accuracy, and the oracle uses it as the cheapest correct route on only 3.3% of questions. The policy prices each action at its mean cost, whereas `r27` generates fewer tokens than `r4` on average (1,900 versus 3,550 on id_test); under this cost structure the small reasoning action is not worth its place in the action set.

Calibration and shift. Calibration did not drive in-distribution accuracy; the raw heads were already nearly calibrated, and scaling matters for the reported probabilities and for budget adherence. The practical failure mode under shift is budget adherence: reasoning length changes with the input distribution, so operating points fixed on one distribution overspend on another. Target recalibration is therefore necessary, and 702 or 1,000 labelled target questions suffice for budget adherence, but it restores only parity with the cascade. A new cost weight or budget still needs no retraining, and deferral uses the same probabilities, but the accuracy advantage over a tuned cascade was not shown outside the training distribution.

7 LIMITATIONS

Several actions in the original motivation are not evaluated. Retrieval and tool use have no counterpart in closed-book multiple choice, we do not use agent execution traces, and we do not evaluate commercial System One models. All answering models belong to one model family at two scales, and one encoder architecture is used for the router. The tasks are multiple choice with a letter readout, which makes success exactly measurable but differs from open-ended generation. MMLU-Pro and SuperGPQA are public, so some questions may appear in pretraining data, which affects route accuracies for all routers alike. Costs are amortized, token-apportioned GPU-time estimates under the recorded batching, not measured execution times of a deployed routing policy; one of the four GPUs was shared with an unrelated process. Reasoning routes are scored by a letter readout after the reasoning block, which can differ from the answer stated in the text; for `r4`, whose post-reasoning text is often an explanation without a parseable final answer, agreement can be checked only on the parseable subset (Table 1). Temperature-only recalibration does not correct every head on SuperGPQA (Section 5.4). The confirmatory comparison covers one budget in distribution; the shift comparisons are descriptive.

8 FUTURE WORK

The next step is agent execution traces with outcome labels, where the router decides at intermediate states of a task and retrieval and tool calls join model scale and reasoning mode as actions. Heads that predict per-question cost, such as reasoning length, would let the policy exploit cost variation as well as success variation. The fast layer itself could be a distilled System One student rather than a general 4B model, lowering the cost of the pass that every policy pays. Batch-1 latency and the realized speedup of executing only the routed workload remain to be measured.

9 CONCLUSION

We cast the decision of when to reason as a System One task: from one cheap pass, predict the probability of success of every action and choose by expected utility under an explicit budget, with every route executed on every question. In distribution, CALIBER retains 83% of the System Two gain at half its cost, short of the 90% target, and beats the best confidence cascade by 2.9 points under the same budget cap through text-encoder per-action success prediction for the large actions. Under shift, transferred operating points miss their budgets; target recalibration restores budget adherence, with no significant difference from the cascade.

Reproducibility statement

The code for data splitting, route execution, feature extraction, router training and evaluation is provided in the supplementary material. Splits are generated deterministically with seed 17 from public datasets (MMLU-Pro test, SuperGPQA), and the answering and encoder models are public checkpoints at pinned revisions. Prompts, decoding settings, router hyperparameters, the cost-weight and threshold grids and the bootstrap procedure are given in Appendices A–D.

AI use statement

Generative AI tools were used to assist with retrieving and discovering related literature, code implementation, analysis, writing and polishing. The research ideas originated with the authors. The authors reviewed and verified AI-assisted content and take full responsibility for the paper's accuracy, originality and conclusions.

References

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. AutoMix: Automatically mixing language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2310.12963.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024a. URL https://openreview.net/forum?id=XUZ2S0JVJP. arXiv:2305.05176.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James T. Kwok, and Yu Zhang. RouterDC: Query-based router by dual contrastive learning for assembling large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b. arXiv:2409.19886.

Yu-Neng Chuang, Prathusha Kameswara Sarma, Parikshit Gopalan, John Boccio, Sara Bolouki, Xia Hu, and Helen Zhou. Learning to route LLMs with confidence tokens. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10859–10878, 2025. arXiv:2410.13284.

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. Learning how hard to think: Input-adaptive allocation of LM computation. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.04707.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A unified approach to routing and cascading for LLMs. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 12987–13010, 2025. arXiv:2410.10347.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2404.14618.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: LLM learns when to think. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2505.13379.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. arXiv:1705.08500.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, 2017. arXiv:1706.04599.

Huizhong Guo, Tianjun Wei, Dongxia Wang, Yingpeng Du, Ziyan Wang, Jie Zhang, and Zhu Sun. Think when needed: Model-aware reasoning routing for LLM-based ranking. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2026. arXiv:2601.18146.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2404.10136.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. RouterBench: A benchmark for multi-LLM routing system, 2024. ICML 2024 Agentic Markets Workshop.

Wittawat Jitkrittum, Neha Gupta, Aditya Krishna Menon, Harikrishna Narasimhan, Ankit Singh Rawat, and Sanjiv Kumar. When does confidence-based cascade deferral suffice? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2307.02764.

Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Congchao Wang, Zifeng Wang, Alec Go, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, Aditya Krishna Menon, and Sanjiv Kumar. Universal model routing for efficient LLM inference. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2502.08773.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know, 2022.

Jungseob Lee, Seongtae Hong, Seungjun Lee, Jaehyung Seo, Junyoung Son, Sugyeong Eo, Chanjun Park, Hyeongju Park, Hyeonseok Moon, and Heuiseok Lim. DART: Draft-agreement routing for training-free adaptive thinking budgets in hybrid reasoning models. In *Findings of the Association for Computational Linguistics: EMNLP 2026*, 2026. arXiv:2606.23181.

Hao Li, Yiqun Zhang, Zhaoyan Guo, Chenxu Wang, Shengji Tang, Qiaosheng Zhang, Yang Chen, Biqing Qi, Peng Ye, Lei Bai, Zhen Wang, and Shuyue Hu. LLMRouterBench: A massive benchmark and unified framework for LLM routing, 2026.

Guosheng Liang, Longguang Zhong, Ziyi Yang, and Xiaojun Quan. ThinkSwitcher: When to think hard, when to think fast. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 5185–5201, 2025. URL https://aclanthology.org/2025.findings-emnlp.278/.

Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. AdaCoT: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning, 2025.

M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, et al. SuperGPQA: Scaling LLM evaluation across 285 graduate disciplines. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. arXiv:2502.14739.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7076–7087, 2020. arXiv:2006.01862.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2406.18665.

Zhihong Pan, Kai Zhang, Yuze Zhao, and Yupeng Han. Route to reason: Adaptive routing for LLM and reasoning strategy selection. In *Proceedings of the ACM Web Conference 2026 (WWW '26)*, pp. 5568–5578, 2026. doi: 10.1145/3774904.3792556. arXiv:2505.19435.

Benjamin Plaut, Khanh Xuan Nguyen, and Tu Trinh. Probabilities of chat LLMs are miscalibrated but still predict correctness on multiple-choice Q&A. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=E6LOh5vz5x. arXiv:2402.13213.

Qwen Team. Qwen3.8-27B. Hugging Face model card, 2026. URL https://huggingface.co/Qwen/Qwen3.8-27B. Accessed 2026-09-25.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. In *Conference on Language Modeling (COLM)*, 2024. arXiv:2309.15789.

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 *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2408.03314.

Seamus Somerstep, Felipe Maia Polo, Allysson Flavio Melo de Oliveira, Prattyush Mangal, Mírian Silva, Onkar Bhardwaj, Mikhail Yurochkin, and Subha Maity. CARROT: A cost aware rate optimal router, 2025. ICLR 2025 Workshop on Foundation Models in the Wild.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. Learning when to think: Shaping adaptive reasoning in R1-style models via multi-stage RL. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2505.10832.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. arXiv:2406.01574.

Zheyuan Wang, Siyu Li, Peiqiao Song, Sijia Chen, Qianqian Song, and Qian Liu. Signed rescue routing: Harm-aware cascades for efficient LLM inference, 2026.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025. URL https://aclanthology.org/2025.acl-long.127/.

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

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thoughts representations for cost-efficient reasoning. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.03094.

Michael J. Zellinger, Rex Liu, and Matt Thomson. Cost-saving LLM cascades with early abstention, 2025.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. AdaptThink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3716–3730, 2025. URL https://aclanthology.org/2025.emnlp-main.184/.

Wenbo Zhang, Lijinghua Zhang, Liner Xiang, and Hengrui Cai. Reasoning is not free: Robust adaptive cost-efficient routing for LLM-as-a-judge. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026a. arXiv:2605.10805.

Yilin Zhang, Han Jiang, Cai Xu, Ying Liu, and Wei Zhao. Calibration-aware uncertainty cascades for efficient heterogeneous model collaboration, 2026b.

A PROMPTS

All prompts are sent as a single user message through the model's chat template, with thinking disabled for direct routes and the self-assessment router and enabled for reasoning routes. The answering prompt is

```
The following is a multiple-choice question about {subject}.

Question: {question}

Options:
A. {option 1}
B. {option 2}
...

Answer with the letter of the correct option.
```

The assistant turn is then continued with the prefix `The answer is (` and the next-token probabilities of the K displayed letters are read out and renormalized. For reasoning routes the prefix follows the generated reasoning and a closing `</think>` tag. The self-assessment router appends to the answering prompt

```
Do not answer the question. Instead decide: can you answer it correctly
right away without step-by-step reasoning?
A. Yes, I can answer it directly.
B. No, it needs careful step-by-step reasoning.
```

continues with `My choice is (` and reads the probabilities of A and B; the probability of B is its routing score.

B ROUTE SETTINGS

Table 6: Route settings.

Setting	Value
Models	Qwen3-4B and Qwen3.8-27B, public checkpoints at pinned revisions, bf16
Inference engine	vLLM, prefix caching, data parallel over four GPUs
Maximum prompt length	4,096 tokens (longer prompts count as failures)
Direct readout	one forward pass, greedy, letters restricted to the K displayed
Reasoning sampling	temperature 0.6, top-p 0.95, top-k 20, seed per question
Reasoning budget	16,384 generated tokens; unfinished traces closed and flagged
Success	argmax letter equals gold; any failure counts as incorrect
GPU cost	amortized batch wall time; reasoning routes apportioned by tokens
Shards	four, questions assigned round-robin

C ROUTER HYPERPARAMETERS

Table 7: Router hyperparameters (CALIBER, text-only ablation and binary win classifier share the architecture; the text-only variant omits $\phi(x)$ and the binary classifier has one head).

Setting	Value
Encoder	ModernBERT-base, mean pooling, max 1,024 tokens
Input text	subject, question and lettered options
System One features $\phi(x)$	max prob., top-2 margin, normalized entropy, top-3 probs., $K/10$, $\log(\text{prompt tokens})/8$, missing-output indicator
Head	Linear(768+9, 256), GELU, dropout 0.1, Linear(256, 4)
Loss	binary cross-entropy per head
Optimizer	AdamW; lr 3×10^{-5} (encoder), 10^{-3} (head)
Schedule	linear, 6% warm-up
Batch size, epochs	32; at most 4, early stopping with patience 1
Early-stopping split	10% of router_train (seeded shuffle)
Precision	bf16 autocast for the encoder, fp32 head
Seeds	17, 29, 43
Logistic baseline	scikit-learn, $C = 1$, per action, on $\phi(x)$ only

D EVALUATION DETAILS

Temperatures. Each head's temperature is found by bounded scalar minimization of binary cross-entropy over $\log T \in [\log 0.05, \log 20]$ on the calibration split.

Operating-point grids. The cost weight ranges over $\lambda = 0$ and 200 log-spaced values in $[10^{-4}, 10^{2}]$. Cascade2, binary and self-assessment thresholds use 201 quantiles of the calibration scores (plus $\pm\infty$); cascade3 and cascade_r4 use 41×41 grids of quantiles of $(m_{\texttt{s1}}, m_{\texttt{11}})$ and $(m_{\texttt{s1}}, m_{\texttt{r4}})$. For each budget the grid point with the highest calibration accuracy among those within budget is selected, ties broken by lower cost; if none is within budget, the cheapest is selected.

Budget reference. The budget reference is the mean `r27` GPU cost on the calibration split in use (router_cal for id_test and transfer; shift_cal or xb_cal for recalibration). The H1 cost limit and the matched-cost target instead use half of the id_test always-`r27` cost. Decision costs: learned routers add their seed-mean batched inference time per question (tokenization included), the self-assessment router adds its prompt pass, and the logistic baseline and the cascades add nothing.

Seeds and statistics. Metrics of seeded routers are averaged over seeds; retention is computed from the seed-mean accuracy. For H2, each question's CALIBER correctness is averaged over seeds and paired with the cascade's correctness; 10,000 bootstrap resamples of questions (seed 17) give the two-sided 95% percentile interval, which reflects question sampling only. Per-seed differences are reported descriptively. Realized-cost intervals and shift comparisons use 2,000 resamples. Accuracy at matched realized cost is read from the upper accuracy–cost frontier of each router's operating points evaluated on id_test, by linear interpolation at the target cost.

Selective prediction. Questions are ranked by confidence and selective accuracy is computed at coverages $0.50, 0.55, \ldots, 1.00$; the CALIBER curve is averaged over seeds.