AMBR: AGREEMENT-BASED ROUTING FOR EFFICIENT REASONING WITH LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models demonstrate strong reasoning capabilities, but deploying them at scale involves a major cost-accuracy trade-off between inexpensive but weaker models and highly capable yet costly ones. We introduce Agreement-Based Routing (AmBR), a lightweight yet effective routing strategy that uses inter-model agreement as a proxy for answer correctness. AmBR first queries two inexpensive models and returns their shared prediction when they agree, escalating only disagreements to a larger (more expensive) fallback model, requiring no training, additional sampling, or model modifications. We evaluate AmBR across multiple reasoning benchmarks using a diverse set of open and closed models. Our results show that agreement between different models is a highly reliable and low-cost correctness signal, enabling substantial reductions in inference cost compared with strong baselines and prior works, while maintaining competitive or even better accuracy, demonstrating a practical and model-agnostic pathway to scalable reasoning.

1 INTRODUCTION

Large language models (LLMs) have demonstrated impressive capabilities across a broad range of reasoning tasks, including mathematical problem-solving and symbolic inference (Comanici et al., 2025; Agarwal et al., 2025). Despite these advances, deploying state-of-the-art LLMs at scale remains constrained by their high inference costs. To address this, smaller, more affordable models were developed that often perform well on relatively simple benchmarks such as GSM8K (Cobbe et al., 2021), yet their accuracy deteriorates sharply on more challenging datasets like MATH (Hendrycks et al., 2021). Conversely, frontier models achieve strong performance across difficult reasoning tasks but incur substantially higher inference costs. This tension poses a practical barrier for cost-sensitive or large-scale applications that demand reliable reasoning performance.

To address this trade-off between inference cost and reasoning quality, we propose **Agreement-Based Routing** (AmBR), a simple and fully zero-shot inference framework that reallocates computation across models rather than increasing it. The key insight behind AmBR is that agreement between independent models can serve as a high-precision signal of correctness. At its core, AmBR relies on the assumption that independent models are not correlated on their errors (although this can be violated in practice): it queries two small, cost-efficient LLMs one time each; if their answers agree, the shared response is returned directly; if they disagree, the query is escalated to a larger fallback model. Thus, unlike self-consistency (Wang et al., 2023b), which repeatedly samples from a single model and can reproduce the same systematic failures, AmBR exploits diversity across independently developed models. Although distinct models may still share errors, agreement becomes a high-precision correctness signal when their wrong answers are less correlated compared with repeated samples from the same model.

AmBR is training-free, requires no external reward models or labeled routing data, and operates entirely in a zero-shot setting (see Figure 1). By making only a single call to each small model and invoking the expensive model selectively, AmBR remains computationally lightweight while systematically reducing dependence on high-cost inference. This distinguishes AmBR from prior routing approaches, which typically rely on expensive learned difficulty predictors or quality estimators to decide whether a query should be handled by a small or large model. Such methods require additional supervision, or fine-tuning, to estimate query difficulty or response quality. In contrast, AmBR replaces learned difficulty estimation with a simple, training-free signal based on agreement between independent models.

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.

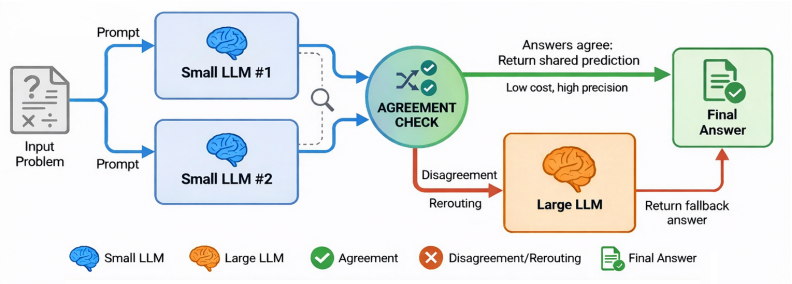


Figure 1: AmBR leverages low-cost small-LLM agreement as a correctness signal. Only disagreements incur expensive fallback calls, enabling scalable, accurate LLM reasoning at reduced cost.

We evaluate AmBR on a diverse suite of reasoning benchmarks spanning arithmetic, symbolic manipulation, and commonsense inference, including GSM8K (Cobbe et al., 2021), GSM Symbolic (Mirzadeh et al., 2025), MATH (Hendrycks et al., 2021), AQuA (Ling et al., 2017), CommonSenseQA (Talmor et al., 2019), StrategyQA (Geva et al., 2021) (with and without evidence), and SciNLI (Sadat & Caragea, 2022). Across these datasets, our best-performing setup AmBR Pro—consisting of Llama 3.3 70B, GPT OSS 120B, and Gemini 2.5 Pro—consistently maintains, and in several cases improves, accuracy relative to the Gemini 2.5 Pro baseline, while achieving an average reduction in inference cost by up to 60% (Figure 3(a)). Our contributions are summarized as follows:

- We introduce Agreement-Based Routing (AmBR), a lightweight, fully zero-shot framework that routes queries based on agreement between two independent LLMs, escalating to a larger model only when disagreement occurs. AmBR requires no router training, auxiliary reward models, or additional sampling, and is compatible with heterogeneous model families.
- Across seven reasoning benchmarks, we show that, under the same zero-shot constraints, AmBR consistently matches or nearly matches the accuracy of the strongest model in each setup at a substantially lower inference cost.
- We provide a systematic analysis of model-derived correctness signals and demonstrate that agreement between independent (or near independent) models yields consistently higher conditional accuracy than within-model sampling, particularly on harder benchmarks.

2 Related Work

Efficient LLM Inference. The choice of inference technique significantly impacts both the performance and computational cost of large language models (LLMs) on downstream tasks. Methods range from greedy decoding and beam search (Freitag & Al-Onaizan, 2017) to sampling-based techniques such as self-consistency (Wang et al., 2023b) and adaptive consistency (Aggarwal et al., 2023), as well as interactive protocols including debate (Du et al., 2023; Liang et al., 2024). While these approaches often improve performance on reasoning benchmarks, they typically require multiple forward passes or multi-agent interaction, causing inference costs to scale linearly or multiplicatively with the number of samples or agents. Because per-token inference cost grows sharply with model size (Pope et al., 2023), such strategies can become prohibitively expensive with large LLMs.

LLM Routing and Cascading. LLM routing has emerged as a complementary approach for balancing inference cost and response quality by dynamically selecting among models of differing capability. Early work primarily studied binary routing setups, assigning queries between one small-but-cheap and one large-but-expensive model. Hybrid LLM inference methods (Ding et al., 2024; Ong et al., 2025) demonstrate that routing easier queries to smaller models can reduce cost while preserving overall performance by escalating harder queries to larger models. Subsequent work generalizes this paradigm to routing among larger model pools. For example, Srivatsa et al. (2024) perform input-feature-based selection over many candidate models, while Šakota et al. (2024) employ meta-models to predict cost-accuracy trade-offs. ZOOTER (Lu et al., 2024) leverages reward models to route queries to suitable LLMs, achieving improved efficiency.

More recent approaches combine routing with adaptive test-time compute. BEST-Route (Ding et al., 2025) introduces a multi-head router that jointly selects a model and the number of samples to draw, using best-of-n sampling and a proxy reward model to improve the performance of cheaper models.

This enables fine-grained cost-quality trade-offs, achieving substantial cost reductions with minimal accuracy loss. FrugalGPT (Chen et al., 2024) proposes a budget-aware cascade that sequentially invokes multiple LLM APIs and relies on a learned post-hoc quality estimator to decide whether to accept or escalate an answer. By exploiting model diversity and adaptive stopping thresholds, FrugalGPT demonstrates that routing across heterogeneous APIs can substantially reduce inference cost while maintaining accuracy comparable to a single large model. More broadly, sequential model cascades (Leviathan et al., 2023; Schuster et al., 2022; Kolawole et al., 2025) share this cost-accuracy trade-off structure, passing queries down a chain of increasingly capable models until a confidence threshold is met. Yue et al. (2024) instead route on the answer consistency of multiple weak-model samples, whose cost grows linearly with the number of samples and can exceed that of the strong model when the cost gap is small.

Closer to our work is Agreement-Based Cascading (ABC) (Kolawole et al., 2025), which cascades ensembles of models across multiple stages and escalates whenever the fraction of agreeing models falls below a threshold θ. Its accuracy guarantee (Kolawole et al., 2025, Prop. 4.1) is conditional on the existence of a safe deferral rule, that is, a θ under which the probability of accepting an incorrect ensemble prediction is at most ϵ, and such a θ must be estimated on a held-out labeled validation set.

AmBR instead routes on agreement between two small models, deferring to a strong, larger model only on disagreement, and requires no tuned thresholds, learnable components, or labeled validation set. Rather than presupposing a reliable deferral rule, we derive the precision of agreement from the small models' accuracies and the probability that they coincide on the same wrong answer (Proposition 3.1), which explains why small heterogeneous pairs yield a high-precision correctness signal. Practically, AmBR performs a single agreement check before escalating directly to the strong model, so each query incurs at most two small-model calls and one strong-model call, and it removes ABC's dependence on θ, or labeled validation set (which are often expensive to obtain).

ABC further restricts its evaluation to tasks with a fixed or normalizable answer space, noting that its voting rule does not directly apply to open-ended generation (Kolawole et al., 2025); AmBR extends to free-form outputs by replacing exact match with bidirectional entailment under an off-the-shelf NLI classifier (Kuhn et al., 2023), leaving the routing rule itself unchanged (Appendix D.4).

3 Proposed Approach: AmBR

We now introduce Agreement-Based Routing (AmBR), a zero-shot inference framework that combines the complementary strengths of small and large LLMs to achieve both efficiency and high reasoning quality. AmBR leverages a routing-based paradigm, where the core objective is to dynamically allocate queries to the appropriate model based on each query's difficulty, but instead of learning a router to predict which queries are easy, our AmBR is entirely training-free and makes routing decisions directly at inference time by leveraging only the outputs of the models themselves. Specifically, it queries two independently trained small models and treats their *agreement* as a high-precision, zero-shot signal of correctness to determine when to use the small models' prediction and when to invoke the large model.

Formally, let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ the set of all possible output responses. Given a query $x \in \mathcal{X}$, the goal is to produce an accurate response $y \in \mathcal{Y}$ while minimizing inference cost. We assume access to three models of differing capability and cost: two small but cost-efficient models $S_1, S_2 : \mathcal{X} \to \mathcal{Y}$, and one large but expensive fallback model $L : \mathcal{X} \to \mathcal{Y}$. We first prompt the two small (inexpensive) independent LLMs to obtain each model's prediction.

$$y_{S_1} = S_1(x) \text{ and } y_{S_2} = S_2(x) \tag{1}$$

A lightweight agreement function checks whether the two responses encode the same answer:

$$\mathbb{A}(x) = \mathbf{1}[y_{S_1} \equiv y_{S_2}], \tag{2}$$

$\mathbb{A}(x)$ takes the value 1 when both models produce an equivalent answer and 0 otherwise. When $\mathbb{A}(x) = 1$ (the two small models agree), we treat the shared prediction as reliable and return it as the final response; when $\mathbb{A}(x) = 0$ (the two models disagree), we prompt the large model and return its response.

$$y_L = L(x) \tag{3}$$

This yields the routing function

$$y(x) = \begin{cases} y_{S_1} & \text{if } \mathbb{A}(x) = 1 \quad \text{(agreement — return shared prediction of } S_1 \text{ and } S_2\text{)}, \\ y_L & \text{if } \mathbb{A}(x) = 0 \quad \text{(disagreement — invoke the large model } L\text{)}. \end{cases} \tag{4}$$

The entire procedure requires no learned parameters, no auxiliary scorer, and no access to model internals: the only inputs are the two model outputs and a lightweight answer-matching function. For tasks with structured outputs, such as the arithmetic benchmarks considered in this work, answer matching is performed via exact string comparison after normalization. The simplicity of this mechanism is deliberate — it ensures that AmBR can be deployed immediately on top of any pair of off-the-shelf models, with no additional infrastructure or setup cost.

Answer equivalence. The agreement function A in Eq. (2) requires a notion of equivalence ($\equiv$) between two answers, which depends on the answer type. For numerical answers and option labels, we use exact string match after normalization. For MATH and MATH500, where one expression can be written in many forms, we use symbolic equivalence with the grader of Lightman et al. (2024). Free-form generation is the most demanding case, since two responses can express the same meaning through entirely different surface forms and no canonical target exists against which to normalize. Following Kuhn et al. (2023), we treat two free-form responses as equivalent when each entails the other under a natural language inference (NLI) classifier. Across all answer types, only the equivalence relation changes, and the routing mechanism in Eqs. (1)–(4) remains unchanged. We provide further details in Appendix D.4.

Agreement as a Correctness Signal. The intuition behind this design relies on model diversity. Because S_1 and S_2 belong to different model families and are trained independently (each with different inductive biases), their errors tend to be idiosyncratic rather than correlated: each model will fail on different subsets of samples, for different reasons. As a consequence, their error distributions are largely independent. If both models independently produce the same answer, the probability that this shared answer is incorrect is substantially lower than the individual error rate of either model alone. Assuming conditional independence of errors,

$$p(y_1 = y_2 = y \neq y^*) \ll p(S_i(x) \neq y^*),\ i = 1, 2, \tag{5}$$

where y^* denotes the ground-truth answer. Agreement therefore acts as a high-precision correctness filter: it is conservative, flagging only the cases where both models are confidently aligned, and escalating all others. This naturally separates queries that the small models can resolve reliably from those that genuinely require the large model. This design stands in contrast to self-consistency (Wang et al., 2023b), which samples multiple responses from a single model: because all samples share the same training distribution, they share the same systematic biases, and majority voting cannot correct errors that are correlated across samples. By exploiting diversity across models rather than across samples, AmBR turns inter-model agreement into a high-precision correctness proxy without any additional supervision.

We now formalize the intuition behind AmBR: when two small independent models agree, the accepted prediction is more likely to be correct than the prediction of either small model alone.

Proposition 3.1 (Agreement boosts precision under independent errors)**.** *Let us fix an input x, and define the events $C_i := \{S_i(x) = y^*(x)\}, i \in \{1, 2\}$, that is, S_i produces the correct answer y^* for input x, and the agreement event $A := \{S_1(x) = S_2(x)\}$. Assume that: 1. each small model has correctness probability $\Pr(C_1) = p_1, \quad \Pr(C_2) = p_2$; and 2. when both models are incorrect, the probability that they produce the same wrong answer is at most ρ, i.e., $\Pr(A \mid \neg C_1, \neg C_2) \leq \rho$, where ρ is typically small when the models make diverse errors. Then $\Pr(C_1 \cap C_2 \mid A) \geq \frac{p_1 p_2}{p_1 p_2 + \rho(1-p_1)(1-p_2)}$. Equivalently, the precision of AmBR on the agreement set is lower bounded by*

$$\Pr\big(S_1(x) = y^*(x) \mid A\big) \ \geq \ \frac{p_1 p_2}{p_1 p_2 + \rho(1 - p_1)(1 - p_2)}. \tag{6}$$

Proof. We show the proof of this proposition in Appendix I. □

The bound shows that agreement becomes highly reliable when: (i) the small models are each reasonably accurate (p_1, p_2 are not too small), and (ii) they are unlikely to make the same mistake (ρ

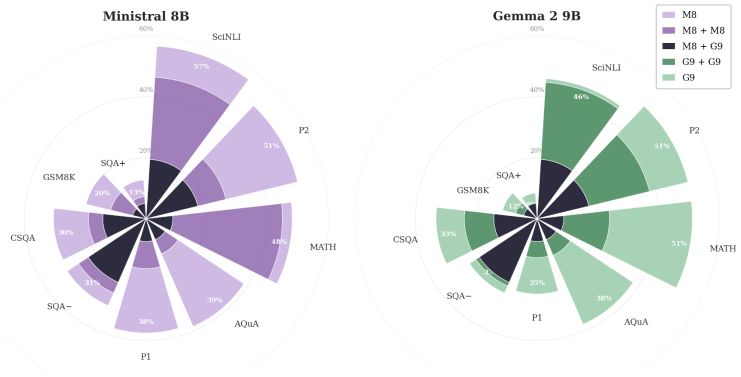


Figure 2: Error-rate decomposition for Ministral 8B (M8) and Gemma 2 9B (G9) across benchmarks. Radial bars show individual-model error, within-model agreement error, and the shared cross-model agreement error for M8+G9. The smaller dark inner region indicates that errors under cross-model agreement are substantially lower than the error rates of either individual model or within-model agreement, supporting agreement as a high-precision correctness signal.

is small). In particular, as $\rho \to 0$,

$$\Pr\big(S_1(x) = y^*(x) \mid A\big) \to 1, \tag{7}$$

which formalizes the central intuition of AmBR: agreement between diverse small models is a high-precision correctness signal.

Proposition 3.1 shows that the precision of agreement depends not only on the standalone accuracies of the small models, but also on their error overlap. When the models are diverse and rarely collapse onto the same wrong answer, agreement sharply increases the likelihood of correctness. This theoretical perspective explains why AmBR benefits from pairing heterogeneous small models: diversity reduces ρ, thereby improving the reliability of agreement and reducing unnecessary escalations.

Correlated errors. In practice, the conditional independence assumption is violated when the two models share training data, architectural inductive biases, or common failure modes. To capture this, let $E_i = \mathbf{1}[\hat{Y}_i = Y]$ be the indicator that model i is correct, and let $\rho_c = \mathrm{Corr}(E_1, E_2)$ measure the degree of error correlation. The joint correctness probability then becomes

$$\Pr(E_1 = 1,\, E_2 = 1) = p_1 p_2 + \rho_c \sqrt{p_1(1-p_1)\, p_2(1-p_2)}. \tag{8}$$

When $\rho_c > 0$, both joint correctness and joint error become more likely, so agreement provides a weaker guarantee than Proposition 3.1 suggests: correlated errors inflate the shared wrong-answer probability ρ_c, degrading the signal.

Figure 2 decomposes the error rates of Ministral 8B and Gemma 2 9B across benchmarks into individual-model, within-model agreement, and cross-model agreement components. While the two models exhibit correlated errors consistent with $\rho_c > 0$, the structure of the answer space modulates agreement quality in ways consistent with our theoretical analysis (see Appendix I): $SQA+$, a binary true/false task with supporting evidence, yields low error rates overall, whereas $SQA-$, the same binary task without evidence, exhibits some of the highest error rates across both within- and cross-model agreement, precisely because the restricted answer space forces erring models to coincide on the single wrong label. Conversely, open-ended benchmarks (GSM8K, MATH, P1, P2) show the largest reductions in error rate when moving from within-model to cross-model agreement, consistent with the multiclass result that agreement becomes an increasingly precise correctness signal as the number of options for target label grows and independent errors are less likely to collide. Crucially, the cross-model agreement region (dark inner wedge) is substantially smaller than either individual or within-model error region across nearly all benchmarks, demonstrating that even positively correlated models can yield a meaningfully lower error rate under agreement than either model achieves alone.

We verify this empirically in Section 5, where within-model pairs, which maximize ρ_c, consistently achieve lower precision than cross-model pairs at similar coverage levels. This motivates a concrete design recommendation: routing pairs should be drawn from distinct model families to keep ρ_c small, reducing shared wrong-answer overlap and preserving the reliability of the agreement signal.

Table 1: Main results across seven benchmarks in the open-source setting. Each cell reports accuracy | inference cost, where higher accuracy is better and lower cost is better. P1 is moved to Appendix C due to space constraints.

Model / Specification	GSM8K Acc | Cost	P2 Acc | Cost	MATH Acc | Cost	AQuA Acc | Cost	CSQA Acc | Cost	SQA- Acc | Cost	SQA+ Acc | Cost	SciNLI Acc | Cost	Average Acc | Cost
Phi-3 Mini 4B (P4)	79.7 | 32	48.7 | 47	41.4 | 67	44.5 | 26	71.6 | 19	63.3 | 23	85.4 | 28	44.4 | 22	59.9 | 33
Ministral 8B (M8)	79.8 | 31	48.8 | 48	52.0 | 63	61.4 | 49	69.6 | 29	68.9 | 25	87.3 | 29	43.3 | 22	63.9 | 37
Llama 3.1 8B (L8)	81.9 | 32	56.7 | 45	50.3 | 101	64.2 | 42	75.6 | 26	70.0 | 28	91.0 | 24	49.8 | 25	67.4 | 40
Gemma 2 9B (G9)	88.3 | 59	49.0 | 90	49.0 | 103	62.2 | 73	76.0 | 60	73.4 | 44	91.6 | 54	54.0 | 68	67.9 | 69
Small models (avg)	82.4 | 38	50.8 | 58	48.2 | 84	58.1 | 48	73.2 | 34	68.9 | 30	88.8 | 34	47.9 | 34	64.8 | 45
Llama 3.3 70B (L70)	95.5 | 266	87.9 | 418	74.2 | 637	81.1 | 358	83.9 | 364	81.7 | 329	96.2 | 291	58.5 | 412	82.4 | 384
$\mathbf{AmBR_1}$: L8 + G9, L70	94.3 | 141	84.7 | 387	73.5 | 621	80.7 | 277	82.6 | 205	79.2 | 154	95.9 | 114	58.0 | 283	81.1 | 273
$\mathbf{AmBR_2}$: L8 + M8, L70	93.8 | 130	87.2 | 362	73.8 | 545	79.9 | 251	82.1 | 159	77.9 | 136	95.5 | 95	59.1 | 296	81.2 | 247
$\mathbf{AmBR_3}$: M8 + G9, L70	93.5 | 144	84.4 | 404	73.3 | 546	82.7 | 301	82.8 | 227	78.2 | 147	94.5 | 122	59.2 | 405	81.1 | 287
$\mathbf{AmBR_4}$: L8 + P4, L70	95.1 | 134	86.9 | 371	72.8 | 584	80.3 | 291	82.1 | 154	75.3 | 144	95.1 | 98	56.8 | 262	80.5 | 255
AmBR 1–4 avg (L70)	94.2 | 137	85.8 | 381	73.4 | 574	80.9 | 280	82.4 | 186	77.7 | 145	95.2 | 107	58.3 | 312	81.0 | 266
GPT OSS 120B (G120)	89.6 | 249	83.6 | 472	82.1 | 694	90.2 | 267	85.6 | 116	78.7 | 284	95.9 | 190	62.1 | 409	83.5 | 335
$\mathbf{AmBR_5}$: L8 + G9, G120	93.3 | 138	83.4 | 426	83.8 | 616	88.6 | 238	82.6 | 115	77.7 | 144	95.9 | 102	59.5 | 337	83.1 | 270
$\mathbf{AmBR_6}$: L8 + M8, G120	91.3 | 126	85.5 | 397	83.3 | 558	87.4 | 218	82.1 | 89	76.1 | 127	95.1 | 81	61.0 | 321	82.7 | 247
$\mathbf{AmBR_7}$: M8 + G9, G120	92.0 | 140	82.9 | 439	83.2 | 566	89.0 | 257	82.8 | 121	77.3 | 139	94.6 | 109	59.3 | 323	82.6 | 267
$\mathbf{AmBR_8}$: L8 + P4, G120	91.6 | 139	84.7 | 408	81.9 | 608	88.2 | 236	82.1 | 81	73.8 | 125	95.1 | 82	60.8 | 341	82.3 | 255
AmBR 5–8 avg (G120)	92.1 | 136	84.1 | 418	83.1 | 587	88.3 | 237	82.4 | 101	76.2 | 134	95.2 | 94	60.1 | 331	82.7 | 259

4 Experiments

We evaluate AmBR on four sets of experiments. First, we study AmBR in an open-source setting, using sub-10B models as the small-model pair and two large open-source models as the fallback (§4.2). Second, we evaluate AmBR in a commercial setting to test whether the same agreement-based routing principle transfers to stronger fallback models (§4.3). Third, we compare AmBR against random routing, learned routing, and adaptive sampling baselines (§4.4), isolating whether the gains come from cross-model agreement rather than from model mixing or repeated sampling. Fourth, we evaluate AmBR on free-form generation against Hybrid LLM on MixInstruct (§4.5). We report statistical analysis over three seeds, including significance tests against the fallback models in Appendix H.

4.1 Experimental Setup

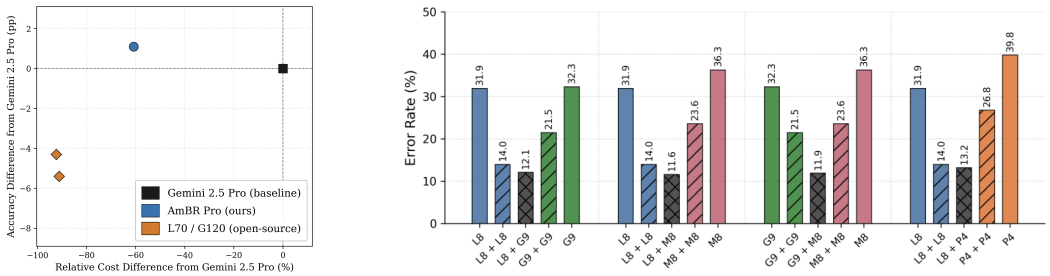
Datasets. We evaluate AmBR on reasoning benchmarks spanning domain and answer format. The benchmarks include *free-response* tasks, where models generate a numerical or symbolic answer, and *multiple-choice* tasks, where models choose from predefined options or labels. The free-response benchmarks are GSM8K (Cobbe et al., 2021), GSM Symbolic P2 (Mirzadeh et al., 2025), and MATH (Hendrycks et al., 2021); AQuA (Ling et al., 2017) provides a multiple-choice mathematical benchmark. For broader natural language reasoning, we use CommonsenseQA (Talmor et al., 2019), StrategyQA without and with evidence (SQA− / SQA+) (Geva et al., 2021), and SciNLI (Sadat & Caragea, 2022). All experiments are zero-shot with chain-of-thought prompting (Wei et al., 2022).

Models. In the open-source setting, we consider two large baselines, Llama 3.3 70B (L70) (Grattafiori et al., 2024) and GPT OSS 120B (G120) (Agarwal et al., 2025), as fallback models. The small models are smaller LLMs: Ministral-8B-Instruct-2410 (M8), Llama 3.1 8B (L8) (Grattafiori et al., 2024), Gemma 2 9B (G9) (Team et al., 2024), and Phi-3 Mini 4B (P4) (Abdin et al., 2024). We construct eight AmBR variants by pairing these small models: AmBR_1–AmBR_4 use L70 as fallback, while AmBR_5–AmBR_8 use G120. In the commercial setting, we use Gemini 2.5 Pro (Comanici et al., 2025) as the fallback model and evaluate *AmBR Pro*, where LLaMA 3.3 70B and GPT OSS 120B serve as the routing "small" models. This setting tests whether agreement-based routing remains effective when the routing models themselves are highly capable.

Evaluation Metrics. We report Accuracy (Acc) and Inference Cost. Accuracy is measured using the standard exact-match criterion, with the exception of MATH/MATH500, which use a custom grader developed by Lightman et al. (2024). Inference Cost follows Ding et al. (2025) and is computed as the average token-based monetary cost per question, scaled by 10^6 for readability. For open-source models, we estimate cost by treating them as if accessed through commercial APIs.

4.2 Open-Source Results

Here, we compare AmBR against two primary baselines: *all-at-large*, where every query is answered by the fallback model, and *all-at-small*, where every query is answered by one of the small models used in the AmBR setup. Table 1 reports results across seven benchmarks. Each cell reports accuracy



(a) Commercial setting.

(b) Agreement error rates.

Figure 3: Left: accuracy–cost trade-off in the commercial setting relative to the Gemini 2.5 Pro baseline. Right: average error rates for individual models, within-model agreement, and cross-model agreement. Lower error rates indicate more reliable predictions on the agreed subset.

and inference cost. For AmBR variants, we use Δ_{Acc}, the absolute accuracy difference from the corresponding fallback baseline, and Δ_{Cost}, the relative inference cost difference. Positive Δ_{Acc} indicates improvement over the fallback; negative Δ_{Cost} indicates cost savings.

AmBR is much stronger than standalone small models, though not as cheap. Small models alone are cheapest but fall far below the fallbacks in accuracy; AmBR uses agreement to recover most of the fallback accuracy while still reducing cost.

AmBR remains close to the all-at-large baselines in accuracy. Under the L70 fallback, AmBR_1–AmBR_4 yield Δ_{Acc} between -1.9 and -1.2 percentage points, with AmBR_2 giving the smallest gap. Under G120, AmBR_5–AmBR_8 achieve Δ_{Acc} between -1.2 and -0.4 percentage points, with AmBR_5 closest to matching G120's average (across all datasets).

AmBR is substantially cheaper than all-at-large inference. The L70-backed variants reduce average inference cost by 25%–36% (Δ_{Cost} between -36% and -25%), with AmBR_2 providing the largest reduction. In the G120 regime, AmBR_5–AmBR_8 reduce average cost by 19%–26%, with AmBR_6 achieving the largest savings while maintaining near-parity accuracy.

4.3 Commercial Results

Figure 3a (full results in Table 8, Appendix C) summarizes the average accuracy–cost trade-off in the commercial setting, with all methods plotted relative to the Gemini 2.5 Pro all-at-large baseline at the origin. While replacing the commercial model with open-source alternatives (L70, G120) achieves substantial cost reductions, it comes at the cost of noticeably lower accuracy. AmBR Pro avoids this trade-off: by using agreement to selectively route queries to the large model only when needed, it occupies the favorable upper-left region of the cost–accuracy plot, exceeding Gemini's accuracy by 1.1 percentage points while reducing inference cost by over 60% averaged across benchmarks.

4.4 Ablation & Comparison to Prior Work

Table 2 compares AmBR with four alternative strategies for reducing L70 inference cost: random routing, Hybrid LLM (Ding et al., 2024), Adaptive Consistency (Aggarwal et al., 2023), and Agreement Based Cascading (ABC) (Kolawole et al., 2025). We also include L8 as an all-at-small lower bound. The comparison uses the same benchmark suite as the main evaluation, with MATH500 and SciNLI500 replacing the full MATH and SciNLI test sets. Figure 4 in Appendix C.1 places all methods on a single cost–accuracy Pareto frontier.

AmBR provides the strongest high-accuracy cost reduction. L8 reaches 67.2 average accuracy against 82.4 for L70. AmBR closes most of this gap: AmBR_1, AmBR_2, and AmBR_4 achieve average accuracy of 81.4, 81.2, and 80.5 respectively, while reducing average cost by 29–36%.

The gains come from agreement, not simply mixing models. Random routing between L8 and L70 is cheaper than AmBR but drops to 75.3 average accuracy — more than seven points below L70 and about six points below the average AmBR setup. AmBR obtains a substantially better trade-off because agreement acts as an informative signal for when routing-model predictions are reliable, rather than mixing models indiscriminately.

Table 2: Ablation and comparison to prior work across seven benchmarks reporting accuracy and inference cost, where higher accuracy is better and lower cost is better. Representative samples are used for MATH and SciNLI for ablation due to the expensive nature of Adaptive Consistency.

Model / Specification	GSM8K Acc | Cost	P2 Acc | Cost	MATH500 Acc | Cost	AQuA Acc | Cost	CSQA Acc | Cost	SQA- Acc | Cost	SQA+ Acc | Cost	SciNLI500 Acc | Cost	Average Acc | Cost
Llama 3.1 8B (L8) ∞	81.9 | 32	56.7 | 45	47.2 | 111	64.2 | 42	75.6 | 26	70.0 | 28	91.0 | 24	51.4 | 26	67.2 | 42
Llama 3.3 70B (L70) ∞	95.5 | 266	87.9 | 418	74.0 | 638	81.1 | 358	83.9 | 364	81.7 | 329	96.2 | 291	59.0 | 406	82.4 | 384
$\mathbf{AmBR_1}$: L8 + G9, L70	94.3 | 141	84.7 | 387	73.4 | 610	80.7 | 277	82.6 | 205	79.2 | 154	95.9 | 114	60.2 | 279	81.4 | 271
$\mathbf{AmBR_2}$: L8 + M8, L70	93.8 | 130	87.2 | 362	72.4 | 542	79.9 | 251	82.1 | 159	77.9 | 136	95.5 | 95	60.6 | 288	81.2 | 245
$\mathbf{AmBR_4}$: L8 + P4, L70	95.1 | 134	86.9 | 371	72.4 | 587	80.3 | 291	82.1 | 154	75.3 | 144	95.1 | 98	56.8 | 267	80.5 | 255
AmBR (1,2,4) avg (L70)	94.4 | 135	86.3 | 373	72.7 | 580	80.3 | 273	82.3 | 173	77.5 | 145	95.5 | 102	59.2 | 278	81.0 | 257
Random: L8 + L70	89.2 | 151	72.2 | 239	62.2 | 362	73.2 | 209	79.5 | 200	76.9 | 175	93.2 | 162	55.6 | 204	75.3 | 213
Hybrid LLM: L8 + L70, t=0.75	84.5 | 57	65.8 | 144	71.8 | 587	64.2 | 42	75.6 | 26	70.0 | 28	91.0 | 24	51.4 | 26	71.8 | 117
Hybrid LLM: L8 + L70, t=0.95	88.2 | 95	77.9 | 265	74.0 | 638	81.1 | 358	83.9 | 364	81.7 | 329	96.2 | 291	59.0 | 406	80.2 | 343
ABC ($\theta = 0.25$)	91.2 | 171	75.5 | 339	71.6 | 631	79.9 | 277	81.4 | 168	75.0 | 154	95.2 | 157	56.8 | 278	78.3 | 272
ABC ($\theta = 0.5$)	94.5 | 196	84.7 | 469	74.4 | 767	81.1 | 357	82.7 | 204	75.3 | 163	95.2 | 159	58.0 | 386	80.7 | 338
ABC ($\theta = 0.75$)	95.4 | 247	87.8 | 569	75.6 | 902	81.5 | 457	84.1 | 302	80.6 | 264	96.4 | 212	58.8 | 550	82.5 | 438
Adapt. Consistency L8, n=40	91.5 | 239	78.8 | 797	57.6 | 1,530	74.8 | 749	77.7 | 179	72.5 | 274	95.3 | 142	55.8 | 289	75.5 | 525

AmBR outperforms learned routing and cascading without requiring router training. Hybrid LLM (Ding et al., 2024) uses a learned DeBERTa-v3-Large router (He et al., 2023) to choose between L8 and L70. Its highest threshold (t=0.95) reaches 80.2 average accuracy but reduces cost by only 11% relative to L70. We run ABC (Kolawole et al., 2025) on the same pool, with all four small models as its first stage and L70 as fallback; $\theta \in \{0.25, 0.5, 0.75\}$ covers every non-trivial rule for four voters. AmBR_1 and AmBR_2 are more accurate and cheaper than both Hybrid LLM (t=0.95) and ABC ($\theta \leq 0.5$), with no labeled data, threshold tuning, or router training. At θ=0.75, ABC exceeds L70 by only 0.1 points yet costs 14% more than L70 itself.

Sampling alone has limited upside without a larger fallback. Adaptive Consistency (Aggarwal et al., 2023) improves over single-sample L8 but reaches only 75.5 average accuracy while increasing average cost beyond the L70 baseline. AmBR performs better by combining a reliability signal from cross-model agreement with access to a stronger model when agreement fails. We show results on L8 Self-Consistency (Wang et al., 2023b) (n=10) in Table 7 in Appendix C.

4.5 Free-form Generation Results

We evaluate our free-form extension of AmBR on MixInstruct (Jiang et al., 2023). AmBR uses agreement between Llama 2 7B and Llama 2 13B (Touvron et al., 2023), escalating disagreements to GPT-3.5-Turbo, the same fallback as Hybrid LLM (Ding et al., 2024), which routes between Llama 2 13B and GPT-3.5-Turbo. Following Hybrid LLM, performance drop measures the percentage decrease in BARTScore (Yuan et al., 2021) relative to always using GPT-3.5-Turbo; cost advantage (CA) is the percentage of queries resolved without escalation. AmBR's drop (3.46%) falls within Hybrid LLM's range (2.9%–3.5%), while its CA is 17 percentage points higher (57% vs. 40%), without training or validation data (Table 3).

Table 3: MixInstruct results (%). DT denotes data transformation.

Method	Drop ↓	CA ↑
Hybrid (Det.)	3.5	40
Hybrid (Prob.)	3.4	40
Hybrid (Prob.+DT)	2.9	40
AmBR	3.46	**57**

5 Analysis

The results in Section 4.4 show that agreement is a more informative routing signal than random mixing or repeated sampling. We now characterize *why*: cross-model agreement is more reliable than within-model agreement, and the agreement rate itself directly governs the cost reductions reported in §4.2. Figure 3b reports average error rates for individual models, within-model agreement, and cross-model agreement across all benchmarks including P1, while Table 4 reports agreement rate and precision conditioned on agreement across seven benchmarks.

Cross-model agreement is more reliable than within-model agreement. Diverse model pairs consistently achieve higher precision conditioned on agreement than uniform pairs, despite agreeing on fewer examples. Averaged across benchmarks, diverse pairs achieve precision between 85.7%

Table 4: Agreement behavior of uniform and diverse model pairs across benchmarks. Each cell reports the agreement rate (Agr.), i.e., the fraction of examples for which the two model outputs match and can be resolved without invoking the fallback, and the precision, or accuracy conditioned on agreement, (Prec.). Higher agreement rates indicate greater coverage, while higher precision indicates more reliable agreement.

Ensemble	GSM8K		P2		MATH		AQuA		CSQA		SQA-		SQA+		SciNLI		Average	
	Agr.	Prec.	Agr.	Prec.	Agr.	Prec.	Agr.	Prec.	Agr.	Prec.	Agr.	Prec.	Agr.	Prec.	Agr.	Prec.	Agr.	Prec.
L8 + L8	74.5	96.1	38.4	87.0	42.0	92.8	56.3	86.7	77.1	83.7	78.2	76.5	89.2	95.6	57.7	60.5	64.2	84.9
G9 + G9	85.8	92.8	60.3	62.1	54.8	76.2	60.2	86.9	65.4	76.1	83.1	76.5	93.4	94.5	93.6	55.4	74.6	77.6
M8 + M8	83.5	88.1	44.0	73.0	93.4	55.2	59.4	87.4	71.7	81.1	76.6	73.4	88.8	92.8	69.2	53.5	73.3	75.6
P4 + P4	80.3	88.4	54.1	66.9	34.4	84.1	26.4	55.2	82.1	78.2	80.1	70.2	86.8	94.0	51.1	46.1	61.9	72.9
Uniform avg	81.0	91.3	49.2	72.2	56.1	77.1	50.6	79.0	74.1	79.8	79.5	74.2	89.6	94.2	67.9	53.9	68.5	77.8
L8 + G9	80.4	97.3	34.5	83.3	40.7	93.9	51.6	93.1	66.6	87.3	74.7	78.2	89.2	96.2	54.4	64.4	61.5	86.7
L8 + M8	73.9	96.2	35.6	94.6	42.1	93.8	53.1	89.6	68.7	86.5	74.7	76.8	87.2	95.0	40.1	64.2	59.4	87.1
M8 + G9	79.8	95.5	36.3	82.5	42.9	91.2	52.4	92.5	63.3	85.8	73.4	77.2	87.5	95.0	23.7	80.4	57.4	87.5
L8 + P4	72.9	97.5	33.4	89.4	34.6	92.5	39.0	86.9	70.9	87.1	73.1	75.9	85.6	97.1	48.2	59.2	57.2	85.7
Diverse avg	76.8	96.6	34.9	87.4	40.1	92.8	49.0	90.5	67.4	86.7	74.0	77.0	87.4	95.8	41.6	67.1	58.9	86.8

and 87.5%, compared to 72.9%–84.9% for uniform pairs — an average gap of 9 percentage points. This higher precision comes at the cost of lower agreement rate: diverse pairs agree on 57–62% of examples on average, versus 62–75% for uniform pairs. The trade-off is favorable: what AmBR loses in coverage it more than recovers in reliability, since escalating more queries to the fallback is precisely the correct behavior when the routing pair is uncertain.

Within-model agreement is vulnerable to shared failure modes. For each model, moving from individual prediction to within-model agreement roughly halves the error rate, but cross-model agreement reduces it further still. For M8 — with a 36.3% individual error rate — within-model agreement (M8+M8) reduces error to 23.6%, while cross-model pairs involving M8 reduce it to 11.6%–11.9%, roughly half again. The same pattern holds for P4: within-model agreement reduces error from 39.8% to 26.8%, while L8+P4 reduces it further to 13.2%. When a model has a systematic failure mode, sampling from it twice surfaces the same error twice, inflating apparent confidence without improving reliability.

Cross-model agreement rate drives cost reduction. The agreement rate determines what fraction of queries are resolved without invoking the fallback, and therefore directly governs cost. Uniform pairs achieve higher agreement rates on average but at lower precision, meaning a larger fraction of their agreed predictions are incorrect — errors that go uncorrected since the fallback is never called. Diverse pairs agree less often but more reliably, producing cost reductions that reflect genuine resolution rather than overconfident routing. For instance, L8+G9 agrees on 80.4% of GSM8K queries at 97.3% precision, but only 34.5% of P2 queries at 83.3% precision — appropriately routing far more of the harder benchmark to the fallback. This behavior, where agreement rate naturally falls on harder benchmarks without any explicit difficulty signal, is a key practical advantage of AmBR over threshold-based routing methods that require difficulty to be estimated in advance.

Additional results and analyses. We provide supplementary analysis across several appendices. Dataset and prompt details are given in Appendices A and B; model pricing details and a qualitative latency analysis are in Appendices G.1 and G.2. Full per-benchmark commercial results, including AmBR Pro, are in Table 8 (Appendix C), alongside P1 results in Table 7. Appendix H reports variance across repeated runs. Appendix D.2 analyzes how precision and agreement rate vary across pairings of differing capability. Appendix E.1 shows that verbalized confidence is a near-useless routing signal in practice. Appendix G.3 provides HybridLLM implementation details. Finally, Appendix D.3 extends the agreement analysis to fine-grained MATH difficulty levels.

6 Conclusion

We introduced AmBR, a training-free routing framework that uses agreement between independent models as a lightweight correctness signal, preserving fallback model accuracy while substantially reducing inference cost without router training, auxiliary reward models, or additional sampling. Our ablations show these gains stem from cross-model agreement rather than random mixing, and that agreement naturally becomes more selective on harder examples — suggesting model diversity can decide when expensive inference is necessary. By reducing inference costs without sacrificing reliability, AmBR lowers the barrier to deploying capable reasoning systems in resource-constrained and cost-sensitive settings.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data and to assist with implementing methods. For the robustness experiments in Appendix F.1 and F.2, we used GPT-5.6 Luna to construct noisy variants of the benchmark inputs through back-translation and synonym replacement, in order to evaluate how our method performs under perturbed input. For implementation, generative AI tools were occasionally used to draft portions of experiment code; all such code was manually reviewed, tested, and refined by the authors, and the majority of the codebase was written without AI assistance.

We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide ingredients for or write proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, clean or reformat datasets, or interpret results. Qualitative or thematic data analysis is not applicable to this work.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects or personal data. All experiments use publicly available benchmarks and models, accessed in accordance with their respective licenses and terms of use. We are not aware of any ethical concerns or conflicts of interest arising from this work.

REPRODUCIBILITY STATEMENT

Datasets and prompts are described in Appendices A and B. Inference costs are computed from the token-level provider pricing listed in Appendix G.1 (Table 18), which reflects a snapshot taken at specific dates. All methods are costed with the same prices, so comparisons within the paper are consistent, but absolute costs, and the relative savings that depend on price ratios between models, may differ as provider pricing changes.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12375–12396, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.761. URL https://aclanthology.org/2023.emnlp-main.761/.

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*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=cSimKw5p6R. Featured Certification.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=02f3mUtqnM.

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Del Carmen Hipolito Garcia, Menglin Xia, Laks V. S. Lakshmanan, Qingyun Wu, and Victor Rühle. BEST-route: Adaptive LLM routing with test-time optimal compute. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=tFBIbCVXkG.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:258841118.

Markus Freitag and Yaser Al-Onaizan. Beam search strategies for neural machine translation. In Thang Luong, Alexandra Birch, Graham Neubig, and Andrew Finch (eds.), *Proceedings of the First Workshop on Neural Machine Translation*, pp. 56–60, Vancouver, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3207. URL https://aclanthology.org/W17-3207/.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. doi: 10.1162/tacl_a_00370. URL https://aclanthology.org/2021.tacl-1.21/.

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

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sE7-XhLxHA.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.792. URL https://aclanthology.org/2023.acl-long.792/.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Steven Kolawole, Don Dennis, Ameet Talwalkar, and Virginia Smith. Agreement-based cascading for efficient inference. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=jn9B7LMlzk.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International conference on machine learning*, pp. 19274–19286. PMLR, 2023.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17889–17904, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.992. URL https://aclanthology.org/2024.emnlp-main.992/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1015. URL https://aclanthology.org/P17-1015/.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1964–1974, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.109. URL https://aclanthology.org/2024.naacl-long.109/.

Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=AjXkRZIvjB.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

Mobashir Sadat and Cornelia Caragea. SciNLI: A corpus for natural language inference on scientific text. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7399–7409, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.511. URL https://aclanthology.org/2022.acl-long.511/.

Marija Šakota, Maxime Peyrard, and Robert West. Fly-swat or cannon? cost-effective language model choice via meta-modeling. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 606–615, 2024.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=uLYc4L3C81A.

Kumar Shridhar, Koustuv Sinha, Andrew Cohen, Tianlu Wang, Ping Yu, Ramakanth Pasunuru, Mrinmaya Sachan, Jason Weston, and Asli Celikyilmaz. The ART of LLM refinement: Ask, refine, and trust. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5872–5883, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.327. URL https://aclanthology.org/2024.naacl-long.327/.

Kv Aditya Srivatsa, Kaushal Kumar Maurya, and Ekaterina Kochmar. Harnessing the power of multiple minds: Lessons learned from LLM routing. In Shabnam Tafreshi, Arjun Akula, João Sedoc, Aleksandr Drozd, Anna Rogers, and Anna Rumshisky (eds.), *Proceedings of the Fifth Workshop on Insights from Negative Results in NLP*, pp. 124–134, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.insights-1.15. URL https://aclanthology.org/2024.insights-1.15/.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2609–2634, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.147. URL https://aclanthology.org/2023.acl-long.147/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore: Evaluating generated text as text generation. *Advances in neural information processing systems*, 34:27263–27277, 2021.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=6okaSfANzh.

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, Tongshuang Wu, and Jianshu Chen. Fact-and-reflection (FaR) improves confidence calibration of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8702–8718, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.515. URL https://aclanthology.org/2024.findings-acl.515/.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WZH7099tgfM.

A DATASETS EVALUATED

Our experiments focus primarily on reasoning benchmarks. Five of these evaluate mathematical problem-solving skills, spanning difficulty levels from grade-school arithmetic to advanced high-school competition problems. The remaining three benchmarks assess multiple-choice reasoning, including natural language inference, true-or-false judgment, and commonsense reasoning. We selected these datasets both for their prominence in related research and to capture a range of difficulty levels that highlight the strengths and limitations of our method.

All of the techniques explored are in a zero-shot setting, with no examples used. Where available, we have tested on the test set of these benchmarks. In the case of CommonsenseQA we used the validation set.

Dataset 1: GSM8K The GSM8K (Cobbe et al., 2021) is a well-known dataset. It consists of 8.5K carefully curated grade-school-level math word problems requiring multi-step arithmetic reasoning, making it a standard benchmark for evaluating the reasoning abilities of large language models. This is the least challenging mathematical dataset that we explore.

Dataset 2: GSM Symbolic Mirzadeh et al. (2025) introduce a variation of the GSM8K dataset called GSM Symbolic, with named entities changed from the original dataset. Furthermore, they introduce two relevant variations of the GSM Symbolic dataset, GSM Symbolic P1 (results in the Appendix) and GSM Symbolic P2. These variations include one and two extra clauses respectively, making the problems slightly more complex. Both versions consist of 5,000 questions generated from 100 templates with various instantiations of named entities and numbers used in the problems. Our experiments do not require training or examples, so we test on the full P1 dataset and 2,500 questions of the P2 dataset, as the remaining questions for P2 were not made public by the authors.

P1 & P2 represent a more difficult version of the GSM8K dataset, and allow us to compare our approach on an interim difficulty, before moving to the MATH dataset.

Dataset 3: MATH The most difficult dataset that we use is the MATH dataset (Hendrycks et al., 2021), which consists of a total of 12,500 competition mathematics problems. We perform our experiments on the publicly available test set of 5,000 problems. Each question in the dataset includes extra information such as a difficulty level (1-5, 5 being the hardest), which allows us to break down the performance by level.

The solutions to the problems in MATH are LaTeX expressions that can go beyond a float or an integer. This can make automatic extraction of the solutions for evaluation purposes a challenge. In our evaluations, we used three approaches: our own string comparison function, which is conservative in its evaluations, LLM-guided evaluation with GPT-4o, and the automatic grading system developed first by Hendrycks et al. (2021) and improved by Lightman et al. (2024). Unless said otherwise, all of our MATH & MATH500 results are evaluated by the automatic grader.

Dataset 4: MATH500 MATH500 (Lightman et al., 2024), is a sample of the full MATH dataset, consisting of 500 problems of various levels and topics. We include this subset of MATH as it is a commonly used benchmark that allows for comparison to more baselines, and because the lower problem set size reduces experiment costs on our end.

Dataset 5: AQuA AQuA (Ling et al., 2017) is a multiple-choice algebraic word-problem dataset designed to evaluate mathematical reasoning beyond short arithmetic answers. Each question is paired with five answer options, requiring models to solve the problem and select the correct choice.

Dataset 6: CommonSenseQA CommonsenseQA (CSQA) (Talmor et al., 2019) is a multiple-choice dataset for commonsense question answering. Each question has five options and only one correct answer. Our experiments for this dataset are performed on the validation set for evaluation, as the test set does not have target labels publicly released.

Dataset 7: StrategyQA StrategyQA (SQA) (Geva et al., 2021) is a Boolean yes/no QA dataset targeting implicit multi-step reasoning. Each instance consists of a short "strategy question," a manual decomposition into sub-questions (i.e., the reasoning steps), and corresponding evidence paragraphs drawn from Wikipedia. The full dataset includes 2,780 examples, with the test dataset consisting of 687 questions. Because the reasoning steps are not explicit in the input question, StrategyQA is

designed to challenge a model's ability to infer a reasoning strategy and chain multiple inferential steps for open-domain questions.

The StrategyQA dataset includes the related evidence, but whether this is provided to models in experiments varies. Some works include experiments with LLMs without providing the extra evidence (Shridhar et al., 2024; Zhao et al., 2024), while others (Kojima et al., 2022; Wang et al., 2023a) do provide the extra information to the models. We report our results on both versions in Table 1 and Table 8 in Appendix C.

Dataset 8: SciNLI SciNLI (Sadat & Caragea, 2022) is a scientific-domain NLI dataset with 107k sentence pairs drawn from NLP research papers, capturing four relation types: entailment, contrasting, reasoning, and neutral. Using linking phrases for distant supervision plus 6,000 manually annotated examples, it reflects the longer, more complex structure of scientific writing. Models perform substantially worse on SciNLI than on general NLI datasets, showing it is a challenging benchmark for scientific text understanding.

Dataset 9: SciNLI500 SciNLI500 is the evaluation subset we construct from the SciNLI test set. It consists of 500 sentence pairs sampled from the test split, with stratification over the target labels to preserve the label distribution while reducing evaluation cost.

Dataset 10: MixInstruct MixInstruct (Jiang et al., 2023) is an instruction-following benchmark built from a mixture of existing instruction datasets, comprising 110K instructions split into 100K training, 5K validation, and 5K test examples. Each instruction is paired with a reference response and eleven candidate responses generated by open instruction-tuned LLMs (including Alpaca, Vicuna, Dolly V2, and Open Assistant), together with automatic reference-based scores (e.g., BLEU, ROUGE, BERTScore, BARTScore) computed against the reference for each candidate. For 4,771 of the test examples, candidate pairs are additionally annotated with an oracle pairwise preference obtained by prompting ChatGPT.

We use MixInstruct to evaluate the free-form generation extension of AmBR described in Sections 3 & D.4, reporting results on the same test split used by Hybrid LLM (Ding et al., 2024) in Section 4.5.

B PROMPTS

Table 5: Prompt templates used for mathematical reasoning benchmarks. Placeholders such as `<question>` and `<options>` are replaced with the corresponding dataset fields at inference time.

Benchmark	Prompt Template

GSM8K / P1 / P2	As an expert problem solver, solve step by step the following mathematical question.
	
	`Q: <question>`
	`A: Let's think step by step.`
	
	Use the following template for structuring your answer:
	
	`Step 1:`
	`Step 2:`
	`...`
	`Last Step:`
	`Final numerical answer: ...`

MATH / MATH500	You are a helpful assistant that interacts entirely in LaTeX code. All responses should be formatted in LaTeX, including explanations, equations, and text.
	
	As an expert problem solver, solve step by step the following mathematical question.
	
	`Q: <question>`
	`A: Let's think step by step.`
	
	Use the following template for structuring your answer:
	
	`Step 1:`
	`Step 2:`
	`...`
	`Last Step:`
	`Final answer: \boxed{...}`

AQuA	Solve the following multiple-choice math problem.
	
	`Question: <question with options>`
	
	Think step by step, then give only the final answer line in this exact format:
	
	`Final Answer: LETTER`

Table 6: Prompt templates used for natural language reasoning benchmarks. Placeholders such as `<question>`, `<evidence>`, `<premise>`, and `<hypothesis>` are replaced with the corresponding dataset fields at inference time.

Benchmark	Prompt Template

CSQA	As an expert problem solver, solve step by step the following question and choose the right answer from the presented options.
	
	`Q: <question>`
	`A: Let's think step by step.`
	
	`The options are: <options>`
	
	Indicate the letter and the word of the option you are choosing as the final answer in the following format:
	
	`Final Answer: 'Letter'. 'Word'`

SQA−	As an expert problem solver, solve step by step the following question and choose the right answer from the presented options.
	
	`Q: <question>`
	`A: Let's think step by step.`
	
	Indicate your final decision as True or False on the final line of your output as follows:
	
	`Final Answer: ...`
	
	If you are unsure, take a guess, but always return either True or False at the end.

SQA+	Here is evidence relevant to the target question:
	
	`<evidence>`
	
	As an expert problem solver, solve step by step the following question and choose the right answer from the presented options.
	
	`Q: <question>`
	`A: Let's think step by step.`
	
	Indicate your final decision as True or False on the final line of your output as follows:
	
	`Final Answer: ...`
	
	If you are unsure, take a guess, but always return either True or False at the end.

SciNLI / SciNLI500	Consider the following two sentences:
	
	`Sentence 1 (premise): <premise>`
	`Sentence 2 (hypothesis): <hypothesis>`
	
	Based only on the information available in these two sentences, which of the following options is true?
	
	`a. Sentence 1 generalizes, specifies, or has an equivalent meaning with Sentence 2.`
	`b. Sentence 1 presents the reason, cause, or condition for the result or conclusion made in Sentence 2.`
	`c. Sentence 2 mentions a comparison, criticism, juxtaposition, or limitation of something said in Sentence 1.`
	`d. Sentence 1 and Sentence 2 are independent.`
	
	Task: Determine the relation between Sentence 1 and Sentence 2.
	
	`Options: entailment, neutral, contrasting, reasoning.`
	
	Denote your chosen option at the end of your response in the following format:
	
	`Final Answer: ...`

C Further Results

Table 7 also includes L8 Self-Consistency at $n = 10$ as an additional baseline. Despite sampling ten times from the same model, Self-Consistency falls substantially short of the L70 baseline across all benchmarks — most strikingly on MATH, where it reaches only 58.5% accuracy while actually *exceeding* L70's inference cost (678 vs. 637). All AmBR variants backed by L70 outperform Self-Consistency in accuracy and, except $AmBR_3$ on P2, on cost as well, reinforcing the finding from Section 4.4 that repeated sampling from a single model has limited upside when that model cannot reliably solve harder instances.

Table 7: Extension of Main results, including GSM Symbolic P1 and Llama 3.1 8B Self Consistency, n = 10.

Model / Specification	GSM8K Acc | Cost	P1 Acc | Cost	P2 Acc | Cost	MATH Acc | Cost
Phi-3 Mini 4B (P4)	79.7 | 32	63.1 | 40	48.7 | 47	41.4 | 67
Ministral 8B (M8)	79.8 | 31	62.5 | 38	48.8 | 48	52.0 | 63
Llama 3.1 8B (L8)	81.9 | 32	73.2 | 37	56.7 | 45	50.3 | 101
Gemma 2 9B (G9)	88.3 | 59	75.3 | 73	49.0 | 90	49.0 | 103
L8 Self-Consistency (n=10)	91.5 | 244	86.5 | 315	74.8 | 392	58.5 | 678
Llama 3.3 70B (L70)	95.5 | 266	92.1 | 328	87.9 | 418	74.2 | 637
$\mathbf{AmBR_1}$: L8 + G9, L70	94.3 | 141	92.3 | 228	84.7 | 387	73.5 | 621
$\mathbf{AmBR_2}$: L8 + M8, L70	93.8 | 130	92.0 | 228	87.2 | 362	73.8 | 545
$\mathbf{AmBR_3}$: M8 + G9, L70	93.5 | 144	90.1 | 243	84.4 | 404	73.3 | 546
$\mathbf{AmBR_4}$: L8 + P4, L70	95.1 | 134	91.8 | 229	86.9 | 371	72.8 | 584
GPT OSS 120B (G120)	89.6 | 249	84.7 | 369	83.6 | 472	82.1 | 694
$\mathbf{AmBR_5}$: L8 + G9, G120	93.3 | 138	89.8 | 239	83.4 | 426	83.8 | 616
$\mathbf{AmBR_6}$: L8 + M8, G120	91.3 | 126	87.7 | 247	85.5 | 397	83.3 | 558
$\mathbf{AmBR_7}$: M8 + G9, G120	92.0 | 140	86.2 | 260	82.9 | 439	83.2 | 566
$\mathbf{AmBR_8}$: L8 + P4, G120	91.6 | 139	87.3 | 248	84.7 | 408	81.9 | 608

Table 8: Main results across seven benchmarks in the commercial setting, with Gemini 2.5 Pro as the all-at-large baseline. Each cell reports accuracy | inference cost, where higher accuracy is better and lower cost is better.

Model / Specification	GSM8K Acc | Cost	P2 Acc | Cost	MATH Acc | Cost	AQuA Acc | Cost	CSQA Acc | Cost	SQA- Acc | Cost	SQA+ Acc | Cost	SciNLI Acc | Cost	Average Acc | Cost
Gemini 2.5 Pro	95.7 | 3088	96.6 | 5046	95.5 | 9939	82.3 | 2544	85.0 | 4479	86.9 | 4078	96.4 | 2101	63.7 | 3564	87.8 | 4355
Llama 3.3 70B (L70)	95.5 | 266	87.9 | 418	74.2 | 637	81.1 | 358	83.9 | 364	81.7 | 329	96.2 | 291	58.5 | 412	82.4 | 384
GPT OSS 120B (G120)	89.6 | 249	83.6 | 472	82.1 | 694	90.2 | 267	85.6 | 116	78.7 | 284	95.9 | 190	62.1 | 409	83.5 | 335
$\mathbf{AmBR_{Pro}}$: L70 + G120, Gemini	96.4 | 829	97.2 | 2071	96.0 | 4768	87.8 | 1005	86.9 | 1078	85.3 | 1361	97.1 | 554	64.4 | 1989	88.9 | 1708

C.1 Cost–Accuracy Frontier of the Ablation Methods

Figure 4 plots average accuracy against average inference cost across the eight benchmarks. Both AmBR_1 and AmBR_2 lie on the Pareto front, occupying the region between the low-cost baselines and the large model. AmBR_2 attains 81.2% accuracy at a cost of 245, recovering 98.5% of L70's accuracy (82.4%) while reducing cost by 36%; AmBR_1 reaches 81.4% at 29% lower cost than L70. At comparable budgets, AmBR clearly outperforms the alternatives: random routing reaches only 75.3% at a cost of 213, and ABC with $\theta = 0.25$ achieves 78.3% at a cost (272) nearly identical to AmBR_1, a gap of 3.1 points. Hybrid LLM ($t = 0.95$) and ABC ($\theta = 0.5$) require 343 and 338 units of cost respectively and still fall below both AmBR_1 and AmBR_2 in accuracy. The only configuration exceeding L70 is ABC with $\theta = 0.75$ (82.5%), which does so marginally at a 14% higher cost, while Adaptive Consistency with L8 incurs the highest cost (525) and is dominated by every AmBR and ABC configuration. Hybrid LLM ($t = 0.75$) and random routing also lie on the frontier, but only in the low-accuracy regime below 76%. AmBR_4 (80.5% at 255) is narrowly dominated by AmBR_2, indicating that the choice of auxiliary model has a modest effect on the trade-off, although all three variants remain close to the frontier.

D Further Analysis of the Agreement Signal

D.1 Sampling Depth vs. Model Diversity

Table 9 compares two strategies for obtaining agreement signals on three mathematical reasoning benchmarks: repeated sampling from L8 at $n \in \{5, 10\}$ under both unanimous and majority-vote aggregation, and unanimous or majority-vote agreement across the diverse L8+G9+M8 triplet, all using L70 as the fallback.

Unanimous model diversity matches baseline accuracy at a fraction of the cost of unanimous same-model sampling. With $n = 10$ unanimous samples, L8 matches the L70 baseline ($\Delta_{\mathrm{Acc}} =$

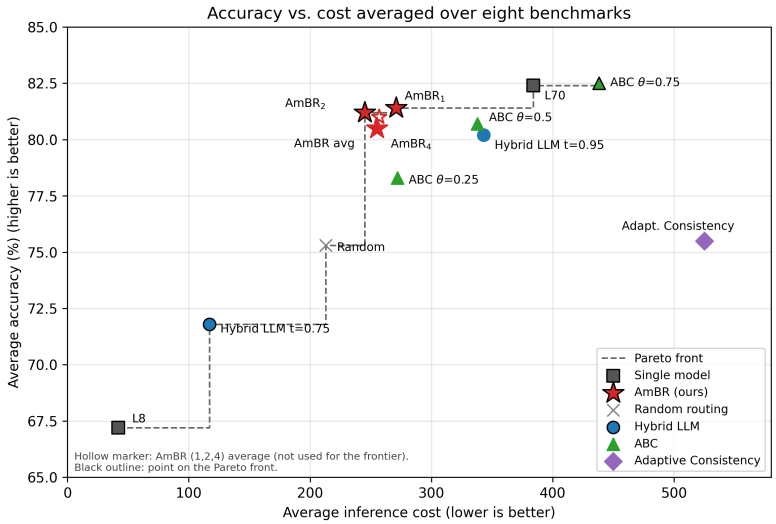


Figure 4: Average accuracy against average inference cost for all methods in Table 2. The dashed line marks the Pareto frontier.

Table 9: Sampling depth vs. model diversity on selected mathematical reasoning benchmarks. Same-model rows use repeated calls to L8, while diverse-model rows use one call each to L8, G9, and M8. Each benchmark cell reports accuracy, API cost, and routing coverage (Acc. | Cost | Cov.), where coverage is the fraction of examples resolved without invoking the fallback under either unanimous agreement or majority vote. The average column reports average accuracy difference (AAD), relative inference cost difference (ΔC) to the L70 all-at-large baseline, and average coverage. All routing methods use Llama 3.3 70B as the fallback model.

Model / Specification	**GSM8K**			**P2**			**MATH**			**Average**		
	Acc.	Cost	Cov	Acc.	Cost	Cov	Acc.	Cost	Cov	AAD	ΔC	Cov.
L8 (n=5, unanimous)	95.1	232	60.1	87.7	568	14.5	73.9	878	25.2	-0.3	+27.0	33.3
L8 (n=5, majority vote)	94.3	156	89.2	83.6	387	57.8	69.7	643	61.0	-3.3	-10.2	69.3
L8 (n=10, unanimous)	95.5	384	49.9	87.9	784	7.3	74.2	1,225	16.7	+0.0	+81.2	24.6
L8 (n=10, majority vote)	95.3	286	86.8	86.1	627	44.9	71.6	981	53.9	-1.5	+43.4	61.9
L8 + G9 + M8 (unanimous)	95.1	199	70.3	87.8	517	21.3	74.3	705	34.2	-0.1	+7.6	41.9
L8 + G9 + M8 (majority vote)	92.7	138	93.0	80.1	355	60.0	71.4	537	59.9	-4.5	-22.0	71.0
Llama 3.3 70B	95.5	266	–	87.9	418	–	74.2	637	–	+0.0	+0.0	0.0

0.0) but increases cost by 81.2% and resolves only 24.6% of examples — the opposite of the cost-saving objective. The unanimous L8+G9+M8 triplet achieves near-identical accuracy ($\Delta_{\text{Acc}} = -0.1$) while increasing cost by only 7.6% and resolving 41.9% of examples. The gap is most pronounced on harder benchmarks: unanimous L8 sampling resolves only 7.3% of P2 examples at $n = 10$, whereas the diverse triplet resolves 21.3%, because independently trained models are less likely to share the systematic failure modes that cause a single model to disagree with itself on difficult inputs.

Majority vote improves coverage but erodes accuracy, regardless of whether sampling is same-model or diverse. Majority vote across the diverse triplet reduces cost by 22.0% and resolves 71.0% of examples, but drops 4.5 accuracy points below the L70 baseline. Same-model majority vote at $n = 5$ similarly falls 3.3 points below baseline despite lower coverage. In both cases, relaxing the agreement criterion from unanimity to majority vote gains coverage at the cost of admitting lower-confidence predictions that unanimity would have escalated. Unanimous agreement — whether same-model or diverse — is therefore the safer criterion when preserving accuracy is the priority, and unanimous diverse agreement dominates unanimous same-model sampling on cost and coverage without sacrificing accuracy.

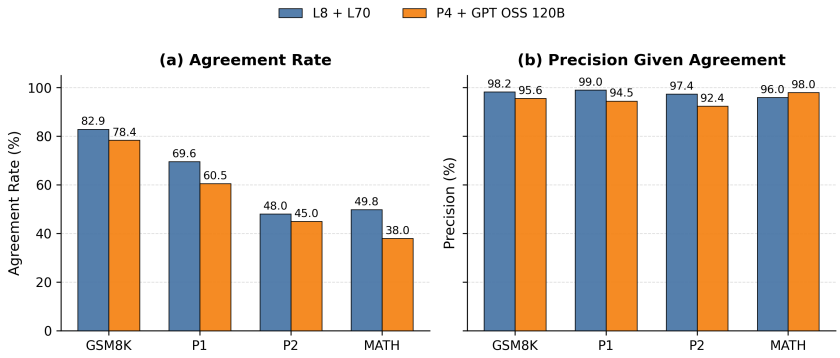


Figure 5: Agreement rate and precision conditioned on agreement for mixed small–large model pairs across mathematical reasoning benchmarks.

Table 10: Additional results with Qwen 8B on mathematical reasoning benchmarks. Each cell reports accuracy | inference cost, where higher accuracy is better and lower cost is better.

Model / Specification	GSM8K Acc | Cost	P1 Acc | Cost	P2 Acc | Cost	MATH Acc | Cost
Llama 3.1 8B (L8)	81.9 | 32	73.2 | 37	56.7 | 45	50.3 | 101
Qwen 8B (Q8)	91.7 | 296	79.8 | 424	64.9 | 551	76.1 | 862
Llama 3.3 70B (L70)	95.5 | 266	92.1 | 328	87.9 | 418	74.2 | 637
Gemini 2.5 Pro (Gemini)	95.7 | 3088	96.7 | 4046	96.6 | 5046	95.5 | 9939
AmBR: L8 + Q8, L70	95.6 | 378	92.7 | 574	88.5 | 832	75.9 | 1307
AmBR: Q8 + L70, Gemini	96.9 | 811	96.4 | 1588	96.5 | 2844	96.2 | 4761

D.2 The Role of Model Pairing

The effectiveness of an agreement-based routing depends not only on model accuracy, but also on the relative cost and capability of the routing pair with respect to the fallback. We examine this through two complementary analyses: Figure 5 compares agreement behavior across model pairs of varying capability mismatch, while Table 10 illustrates how the same model can be either an ineffective or highly effective router depending on where it sits in the setup.

Precision conditioned on agreement is stable across model pairings; coverage is not. Regardless of how mismatched the two models are in scale, exact agreement remains a high-precision correctness signal. L8+L70 achieves 96.0–99.0% precision across the math benchmarks, and P4+GPT OSS 120B remains between 92.4–98.0%, suggesting that when two models of very different capability agree, their shared answer is nearly always correct. The bottleneck is coverage: agreement rate is constrained by the weaker model in the pair, since both models must produce the same answer for a query to be resolved without the fallback. P4+GPT OSS 120B illustrates this clearly — despite high precision, its agreement rate drops to 38.0% on MATH, meaning over 60% of queries are still escalated. The same pattern appears in Table 4: pairs involving P4 consistently achieve lower agreement rates than pairings among stronger routing models, regardless of benchmark.

Similarly capable routing models maximise the coverage of AmBR. Since precision is relatively stable across pairings but agreement rate varies substantially, routing model capability drives cost efficiency more than it drives reliability. A pair that agrees frequently handles more queries cheaply without sacrificing precision on the agreed subset; a mismatched pair achieves comparable precision but escalates more often, eroding the cost advantage of routing. This makes similarly capable routing models the natural choice for AmBR: they are more likely to agree on the queries within their competence while still correctly escalating the ones that genuinely require the fallback.

Expensive routing models can erase AmBR's cost advantage. A model should not be treated as a "small" routing model solely because it has relatively few parameters — what matters is whether its cost is low enough, relative to the fallback, that resolving agreed examples without escalation

produces meaningful savings. Table 10 makes this concrete. Q8 is substantially more accurate than L8 as a standalone model across all four mathematical benchmarks, improving from 81.9% to 91.7% on GSM8K and from 50.3% to 76.1% on MATH. However, this accuracy comes at much higher inference cost: Q8 is more expensive than L70 on every benchmark. As a result, using Q8 as a low-level router in L8+Q8→L70 is not cost-effective — though it slightly improves accuracy, it exceeds L70 cost in all cases, reaching more than twice the L70 cost on MATH.

Effective AmBR pairings are defined by relative cost and capability. The same model can become highly useful when paired with more appropriate models. Rather than treating Q8 as a cheap router for L70, pairing Q8 with L70 as an intermediate ensemble for the much more expensive Gemini 2.5 Pro yields a favorable trade-off: on GSM8K, accuracy improves from 95.7% to 96.9% while cost falls from 3088 to 811; on MATH, accuracy improves from 95.5% to 96.2% while cost falls from 9939 to 4761. The key design principle is that AmBR should pair models according to the cost differential between the routing pair and fallback, not parameter count alone. A model too expensive to serve as a low-level router may still be valuable as part of a higher-tier agreement pair when the fallback is substantially more costly.

D.3 Agreement Rate and Benchmark Difficulty

Table 11: MATH performance by difficulty level. Large-model rows report accuracy (%). Small-model pair rows report agreement rate and precision conditioned on agreement, shown as Agr. | Prec. Higher accuracy, agreement rate, and precision are better.

Model / Specification	Level 1		Level 2		Level 3		Level 4		Level 5		Overall	
	Agr.	Prec.	Agr.	Prec.	Agr.	Prec.	Agr.	Prec.	Agr.	Prec.	Agr.	Prec.
Llama 3.3 70B (L70) (Acc)	95.2		88.9		83.0		72.0		51.9		74.2	
GPT OSS 120B (Acc)	86.5		86.9		85.1		81.0		76.0		82.1	
L8 + G9	79.4	97.4	58.2	96.3	48.5	95.1	32.9	92.5	16.5	82.2	40.7	93.9
L8 + M8	78.0	97.4	59.8	96.6	50.6	95.8	33.9	91.5	18.4	81.6	42.1	93.8
M8 + G9	81.7	97.5	61.6	94.9	51.5	92.8	33.6	88.7	18.7	74.1	42.9	91.2
L8 + P4	66.6	95.9	52.0	95.9	40.7	93.5	26.9	92.0	14.2	77.7	34.6	92.5
Count of small-agreement-correct questions that L70 would have missed												
L8 + G9	2/437		7/894		16/1131		10/1214		12/1324		47/5000	
L8 + M8	1/437		4/894		20/1131		6/1214		12/1324		43/5000	
G9 + M8	5/437		9/894		22/1131		8/1214		10/1324		54/5000	
L8 + P4	0/437		4/894		13/1131		10/1214		11/1324		38/5000	
Count of small-agreement-correct questions that G120 would have missed												
L8 + G9	31/437		56/894		48/1131		37/1214		20/1324		192/5000	
L8 + M8	30/437		56/894		52/1131		47/1214		22/1324		207/5000	
G9 + M8	36/437		55/894		50/1131		40/1214		21/1324		202/5000	
L8 + P4	21/437		42/894		41/1131		27/1214		22/1324		153/5000	

Table 11 breaks down AmBR's agreement behavior on the MATH dataset across five difficulty levels, extending the analysis of agreement level per benchmark observed in Section 5 to a finer-grained difficulty axis. Two complementary patterns emerge that together suggest AmBR functions as an implicit difficulty detector — routing more aggressively to the fallback precisely when small models are least reliable.

Agreement rate tracks problem difficulty. Agreement rate falls sharply and consistently with difficulty across all routing pairs. For L8+G9, agreement drops from 79.4% at Level 1 to just 16.5% at Level 5, meaning that on the hardest problems, over 80% of queries are escalated to the fallback. L8+P4 have the lowest agreement rate at level 5, falling from 66.6% to 14.2%, while M8+G9 is the most resilient, maintaining 18.7% agreement at Level 5. Crucially, this escalation behavior mirrors L70's own accuracy curve, which falls from 95.2% at Level 1 to 51.9% at Level 5: AmBR routes more queries to the fallback on precisely the problems that are hardest even for the fallback model itself, without ever observing difficulty labels. This mirrors the benchmark-level adjustment noted in Section 5, where agreement rate naturally fell on harder benchmarks such as P2 and MATH relative to GSM8K — here we see the same effect operating within a single benchmark across difficulty strata.

Precision conditioned on agreement remains high across all levels. Despite declining coverage, agreed predictions remain reliable throughout. L8+G9 achieves 97.4% precision at Level 1 and 82.2% at Level 5 — well above any individual small model's accuracy at that level. M8+G9 shows the sharpest precision drop, falling to 74.1% at Level 5, reflecting greater shared failure modes between those two models on the hardest problems. Overall, precision across all pairs and levels remains above 74%, confirming that agreement continues to function as a high-precision correctness signal even when coverage is low. Together with the agreement rate results above, this suggests that AmBR's routing naturally concentrates fallback calls where they matter most: on hard examples where small models are unreliable and the fallback's superior capability is most needed.

Agreed predictions are largely complementary to fallback coverage. The lower panels of Table 11 show the proportion of small-model agreement-correct questions that L70 or G120 would have answered incorrectly — cases where AmBR's routing recovers correct answers that the fallback itself misses. Against L70, this is rare: across all pairs and levels, fewer than 55 out of 5,000 questions fall into this category, confirming that L70 is a reliable fallback and that little accuracy is lost by escalating to it. Against G120, the picture is more striking: up to 207 out of 5,000 agreed-correct questions would have been answered incorrectly by G120 alone, concentrated at Levels 2 and 3. This suggests that in the G120-backed setting, the routing pair's agreed answers are not merely redundant with the fallback — they occasionally recover correct answers that G120 would have missed, providing an additional source of accuracy gain beyond simple cost reduction.

D.4 Semantic Agreement for Free-Form Generation

When answers cannot be compared as strings or expressions, we replace the exact-match check with bidirectional entailment (Kuhn et al., 2023). Let $\mathrm{NLI}(s, s')$ denote the label that an NLI classifier assigns to the ordered pair (s, s'). Then

$$A(x) = \mathbf{1}\big[\mathrm{NLI}(y_{S_1}, y_{S_2}) = \text{entailment} \ \wedge \ \mathrm{NLI}(y_{S_2}, y_{S_1}) = \text{entailment}\big]. \tag{9}$$

We use DeBERTa-large fine-tuned on MNLI (He et al., 2021) and prepend the query x to each response before classification. When $A(x) = 1$, the two responses are treated as interchangeable and one is returned, chosen uniformly at random. When $A(x) = 0$, the query is escalated to the fallback model as before. Only the agreement criterion changes; AmBR still requires no additional sampling, training, or model modification. We evaluate this extension in Section 4.5.

E Agreement on Open-Answer versus Multiple-Choice Tasks

The benchmarks in our main evaluation fall into two groups. Open-answer tasks (GSM8K, P2, MATH) have answers that are numbers or mathematical expressions. Multiple-choice tasks (AQuA, CSQA, SQA−, SQA+, SciNLI) have answers drawn from a small set of labels. Both groups use the exact or symbolic equivalence described in Section 3, so the analysis below isolates the effect of answer format on agreement quality.

Precision gap. Agreement is more precise on open-answer tasks than on multiple-choice tasks (Table 4). When each diverse pair's precision is averaged over the benchmarks in a group, the four pairs range from 89.7% to 94.9% on open-answer tasks and from 81.2% to 86.2% on multiple-choice tasks.

Role of answer-space size. The size of the answer space accounts for most of this gap. An open-answer response is a number or an expression, so two models that err independently rarely produce the same wrong answer. With only a few labels, two erring models can select the same incorrect option even when their reasoning differs, as the K-class analysis in Appendix I predicts. The drop is sharpest on SciNLI, where three of the four diverse pairs range from 59.2% to 64.4%, and on SQA−, where all four range from 75.9% to 78.2%. AQuA and SQA+ remain at the open-answer level (diverse-pair averages of 90.5% and 95.8%), plausibly because the mathematical structure of AQuA and the supporting evidence in SQA+ constrain the answer more tightly.

Effect on coverage. The small answer space also raises coverage. Diverse pairs agree on 63.9% of examples on average across the multiple-choice benchmarks, compared with 50.6% across the open-answer benchmarks, so fewer multiple-choice queries reach the fallback.

Informativeness of multiple-choice agreement. Agreement nonetheless remains informative on multiple-choice tasks. Diverse pairs are more precise than uniform pairs on all five benchmarks, by margins ranging from 1.6 points (SQA+) to 13.2 points (SciNLI), although they agree on fewer examples on every benchmark (Figure 6 and Table 4).

E.1 Verbalized Confidence as a Routing Signal

Table 12: Verbalized confidence coverage and accuracy on GSM Symbolic P1 (5,000 examples) and P2 (2,500 examples). Each cell reports the fraction of responses assigned a given confidence label (Cov.) and the accuracy on those responses (Acc.), where higher accuracy and more balanced coverage are better. Models express "high" confidence on nearly all queries regardless of benchmark difficulty, rendering the signal unable to discriminate easy from hard examples.

Model	GSM Symbolic P1			GSM Symbolic P2		
	High	Medium	Low	High	Medium	Low
	Cov. | Acc.	Cov. | Acc.	Cov. | Acc.	Cov. | Acc.	Cov. | Acc.	Cov. | Acc.
Llama 3.1 8B (L8)	95.4% | 76.3%	0.2% | 55.6%	0.1% | 0.0%	92.2% | 56.6%	0.5% | 33.3%	0.2% | 33.3%
Llama 3.3 70B (L70)	98.6% | 91.8%	0.5% | 66.7%	0.4% | 45.0%	94.8% | 89.7%	2.0% | 52.0%	0.5% | 61.5%
Gemma 2 9B (G9)	97.6% | 69.6%	0.9% | 15.9%	1.2% | 1.6%	93.7% | 44.5%	3.2% | 3.7%	2.6% | 0.0%
Phi-3 Mini 4B (P4)	99.8% | 62.7%	0.1% | 0.0%	0.1% | 0.0%	99.5% | 47.2%	0.2% | 0.0%	0.1% | 0.0%

A natural alternative to agreement-based routing is to use a model's verbalized confidence as a routing signal, prompting it to report whether it is confident in its answer. We investigated this on GSM Symbolic P1 (5,000 examples) and P2 (2,500 examples) across four models (L8, L70, G9, P4). The results reveal two fundamental limitations that together render verbalized confidence impractical as a routing signal.

Confidence labels are almost uniformly high, regardless of correctness. Across both benchmarks and all models, between 92% and 99% of responses are labelled "high" confidence, with "medium" and "low" labels appearing on roughly fewer than 5% of responses combined. This near-total concentration on a single label means verbalized confidence provides almost no discrimination between queries a model can and cannot reliably answer, making it effectively useless as a signal for selective escalation.

Accuracy on "high" confidence responses varies widely and tracks task difficulty, not expressed confidence. On P1, accuracy on "high" confidence responses ranges from 62.7% (Phi) to 91.8% (L70). On the harder P2, these figures drop to 44.5% (G9) and 89.7% (L70) respectively — a substantial decline that mirrors the benchmark's increased difficulty. Crucially, this decline is not reflected in the confidence labels: models express "high" confidence at nearly identical rates on P1 and P2 despite the large drop in actual accuracy. A routing signal that does not respond to task difficulty cannot selectively escalate harder queries, which is precisely the property required for cost-efficient routing.

Together, these results suggest that verbalized confidence fails as a routing signal not because models are systematically overconfident, but because the label distribution is too degenerate to carry useful information. In contrast, AmBR's agreement signal naturally becomes more selective on harder benchmarks — as shown in Section 5 and Appendix D.3 — without any explicit difficulty signal or self-reporting from the model.

F Robustness of the Agreement Signal

F.1 Robustness to Input Noise

To directly test whether AmBR's routing pairs are vulnerable to correlated failure under noisy inputs, we construct a noised variant of the same seven benchmarks used in Table 1 (GSM8K,

Table 13: Results evaluated on round-trip-translation-noised English test data (EN→FR→EN via GPT-5.6 Luna) under 0-shot Chain-of-Thought prompting. Each cell reports accuracy | inference cost, where higher accuracy is better and lower cost is better.

Model / Specification	GSM8K Acc | Cost	P2 Acc | Cost	MATH Acc | Cost	AQuA Acc | Cost	CSQA Acc | Cost	SQA- Acc | Cost	SQA+ Acc | Cost	SciNLI Acc | Cost	Average Acc | Cost
Phi-3 Mini 4B (P4)	80.1 | 41	41.0 | 70	40.4 | 68	48.4 | 44	70.3 | 37	62.6 | 37	83.1 | 38	43.4 | 51	58.7 | 48
Ministral 8B (M8)	81.3 | 40	52.2 | 63	50.4 | 64	62.6 | 59	69.5 | 31	68.3 | 31	89.2 | 33	49.1 | 50	65.3 | 46
Llama 3.1 8B (L8)	83.6 | 36	48.2 | 83	48.1 | 84	62.2 | 69	72.9 | 31	69.3 | 35	87.8 | 31	40.4 | 39	64.1 | 51
Gemma 2 9B (G9)	86.9 | 73	49.2 | 111	49.4 | 110	58.7 | 88	74.8 | 66	64.0 | 69	84.7 | 70	50.2 | 99	64.7 | 86
Small models (avg)	83.0 | 48	47.6 | 82	47.1 | 81	58.0 | 65	71.9 | 41	66.0 | 43	86.2 | 43	45.8 | 60	63.2 | 58
Llama 3.3 70B (L70)	94.6 | 318	71.0 | 660	73.4 | 643	79.1 | 528	81.7 | 353	79.0 | 341	94.5 | 316	58.1 | 507	78.9 | 458
$\mathbf{AmBR_1}$: L8 + G9, L70	94.1 | 173	71.8 | 598	73.4 | 582	79.5 | 410	81.9 | 225	78.3 | 211	95.1 | 162	55.5 | 417	78.7 | 347
$\mathbf{AmBR_2}$: L8 + M8, L70	93.7 | 152	71.2 | 542	73.4 | 531	81.9 | 379	80.4 | 170	75.4 | 161	93.7 | 110	57.7 | 396	78.4 | 305
$\mathbf{AmBR_3}$: M8 + G9, L70	93.6 | 180	71.0 | 550	72.8 | 544	81.1 | 417	80.5 | 235	76.6 | 207	94.5 | 159	56.4 | 397	78.3 | 336
$\mathbf{AmBR_4}$: L8 + P4, L70	94.2 | 160	71.4 | 597	73.2 | 584	78.7 | 426	79.9 | 194	75.0 | 170	92.6 | 124	57.9 | 423	77.9 | 335
AmBR 1–4 avg (L70)	93.9 | 166	71.4 | 572	73.2 | 560	80.3 | 408	80.7 | 206	76.3 | 187	94.0 | 139	56.9 | 408	78.3 | 331
GPT OSS 120B (G120)	88.8 | 269	77.0 | 678	80.3 | 645	89.4 | 342	80.7 | 275	77.4 | 354	93.4 | 243	59.4 | 400	80.8 | 401
$\mathbf{AmBR_5}$: L8 + G9, G120	92.9 | 163	80.6 | 609	82.5 | 583	88.2 | 321	80.6 | 197	77.1 | 215	94.5 | 148	56.7 | 358	81.6 | 324
$\mathbf{AmBR_6}$: L8 + M8, G120	91.5 | 140	79.2 | 553	81.9 | 533	87.8 | 291	79.8 | 146	75.5 | 165	93.6 | 99	58.0 | 331	80.9 | 282
$\mathbf{AmBR_7}$: M8 + G9, G120	91.8 | 169	79.6 | 560	81.3 | 545	88.2 | 322	79.4 | 204	75.7 | 211	93.4 | 146	58.4 | 345	81.0 | 313
$\mathbf{AmBR_8}$: L8 + P4, G120	91.6 | 147	78.4 | 609	81.6 | 586	86.2 | 316	79.4 | 166	72.9 | 174	92.4 | 111	58.7 | 353	80.2 | 308
AmBR 5–8 avg (G120)	91.9 | 155	79.5 | 583	81.8 | 562	87.6 | 313	79.8 | 178	75.3 | 191	93.5 | 126	57.9 | 347	80.9 | 307

GSM Symbolic P2, MATH, AQuA, CSQA, StrategyQA without and with evidence, and SciNLI) via round-trip translation: each test example is translated from English to French and back to English (EN→FR→EN) using GPT-5.6 Luna, introducing minor paraphrastic noise while preserving the underlying question. We re-evaluate all four small models, both large fallbacks, and all eight AmBR configurations under identical routing pairs and prompting (0-shot CoT) as in Table 1. Table 13 reports the results.

Noise degrades almost every method, but degrades the fallbacks more than AmBR. Average accuracy falls for almost every method relative to Table 1: the small models drop by 1.6 points (64.8→63.2), L70 drops by 3.5 points (82.4→78.9), and G120 drops by 2.7 points (83.5→80.8). AmBR's degradation is smaller than its corresponding fallback's in both settings: the L70-backed variants fall by 2.7 points on average (81.0→78.3) and the G120-backed variants fall by only 1.8 points (82.7→80.9).

Under noise, AmBR moves closer to L70 and edges past G120. Under L70, Δ_{Acc}, as defined in section 4.2, shrinks from -1.4 points on clean inputs to -0.6 points under noise. Under G120, Δ_{Acc} is already near zero on clean inputs (-0.8) and turns slightly positive under noise ($+0.1$), i.e., the AmBR 5–8 average matches G120 itself. This is the opposite of what correlated-failure degradation would predict: if noise caused the two small models to agree more often on the same wrong answer, the precision of agreement — and with it AmBR's accuracy relative to its fallback — would fall rather than hold steady or improve.

Cost savings are largely preserved. Relative to the corresponding fallback, average inference cost changes only modestly between clean and noised settings: -30.7% vs. -27.7% under L70, and -22.7% vs. -23.4% under G120.

Benchmark-level exceptions. Two benchmarks deviate from the overall pattern. On SciNLI, the accuracy gap to L70 widens under noise (from -0.2 to -1.2 points), but this does not recur under G120, where the gap narrows (from -2.0 to -1.5 points). Because the agreement decision is independent of the fallback, we attribute this primarily to L70's unusual robustness on SciNLI (58.5→58.1) rather than to a failure of the agreement signal. On SQA−, relative cost savings shrink more than on any other benchmark under both fallbacks ($-55.9\% \rightarrow -45.2\%$ under L70, $-52.8\% \rightarrow -46.0\%$ under G120). This reflects more frequent escalation to the fallback, which means noise causes the small models to disagree more often, not to converge on shared incorrect answers.

Table 14: Average change in accuracy, agreement rate, and precision (accuracy conditioned on agreement) under synonym-replacement noise, relative to the unperturbed test sets, averaged across seven benchmarks (GSM8K, AQuA, MATH, SciNLI, SQA−, SQA+, CSQA). All values are percentage points.

Model	ΔAcc
L8	−1.50
M8	−1.47
G9	−0.66
P4	+0.19

Pair	ΔAgr	ΔPrec
L8+M8	−1.57	−1.09
L8+G9	−0.20	−1.07
L8+P4	+0.51	−1.09
M8+G9	−0.54	−0.83

These results show that under this form of minor input noise, AmBR's cost–accuracy trade-off holds up as well as, or better than, its all-at-large baselines, and we find no evidence that back-translation noise induces the kind of correlated agreement failure that would erode precision.

F.2 Robustness to Synonym Replacement

As a second, independent noise setting, complementing the round-trip-translation perturbation above, we replace k randomly chosen words per test example with a synonym, where $k \in \{1, 2, 3\}$ is drawn uniformly per example, using GPT-5.6 Luna to select the replacements. Unlike the round-trip-translation setting, which evaluates the full AmBR pipeline (accuracy and cost after routing to the fallback), here we isolate the agreement signal itself: we evaluate the four small models and four diverse routing pairs (L8+M8, L8+G9, L8+P4, M8+G9) on seven benchmarks (GSM8K, AQuA, MATH, SciNLI, SQA−, SQA+, CSQA), reporting the change in individual-model accuracy and in pairwise agreement rate and precision conditioned on agreement, relative to the unperturbed test sets. Table 14 reports the average change across all seven benchmarks.

Precision conditioned on agreement is stable under synonym replacement. Averaged across benchmarks, precision falls by no more than 1.09 percentage points for any of the four pairs, a smaller and more tightly clustered shift than either agreement rate or individual-model accuracy, both of which move in either direction depending on the pair and benchmark. This mirrors the finding in Appendix F.1 under a different noise type: even when a perturbation shifts which model is more accurate or how often two models agree, the correctness of the agreed prediction itself is largely unaffected.

AQuA stresses the underlying signal hardest, and precision absorbs most of it. AQuA shows the sharpest declines in the entire experiment: individual accuracy falls by up to 9.45 points (M8), and the agreement rate for L8+M8 falls by 15.75 points, the single largest movement observed for any pair on any benchmark. Despite this, precision conditioned on agreement moves by at most 4.22 points (L8+P4, itself the largest precision shift in the whole experiment), and by 1.67 points or less for the other three pairs. Every other benchmark shows accuracy and agreement-rate shifts within roughly five points in either direction, well below AQuA's swings, consistent with the near-zero seven-benchmark average. AQuA's multiple-choice format, where synonym substitution can flip which answer option a model selects without changing its underlying reasoning, is a plausible source of this added sensitivity.

These results extend the robustness finding in Appendix F.1 to a second, word-level noise type: even on the one benchmark where the underlying signal is disrupted the most, precision conditioned on agreement, the property AmBR's routing decision depends on, is disrupted the least.

F.3 Robustness to Alternative Prompting Strategies

The main results in Table 1 use zero-shot Chain-of-Thought (CoT) prompting throughout, for both the small models and the fallbacks. To test whether AmBR's cost–accuracy profile is an artifact of this particular prompt, we repeat the full evaluation under two single-pass alternatives, Plan-and-Solve (PnS) (Wang et al., 2023a) and Least-to-Most (LtM) (Zhou et al., 2023) prompting, using the same eight benchmarks as Table 1. All routing pairs, fallbacks, and AmBR configurations are unchanged from Table 1; only the prompt template differs. Table 15 reports results under PnS and Table 16 under LtM. As in Table 1, the Average column is computed over the same eight benchmarks used.

Table 15: Results across the same eight benchmarks as Table 1, under the Plan-and-Solve (PnS) prompting strategy. Each cell reports accuracy | inference cost, where higher accuracy is better and lower cost is better.

Model / Specification	GSM8K		P2		MATH		AQuA		CSQA		SQA-		SQA+		SciNLI		Average	
	Acc	Cost	Acc	Cost	Acc	Cost	Acc	Cost	Acc	Cost	Acc	Cost	Acc	Cost	Acc	Cost	Acc	Cost
Phi-3 Mini 4B (P4)	70.8	48	39.8	71	30.7	68	51.2	56	72.9	43	66.4	42	85.7	44	45.1	62	57.8	54
Ministral 8B (M8)	74.2	42	43.3	64	39.8	61	56.3	56	71.6	34	63.3	31	86.5	34	50.0	56	60.6	47
Llama 3.1 8B (L8)	84.3	39	50.6	59	48.5	82	59.4	82	74.1	35	67.5	39	90.0	35	44.5	44	64.9	52
Gemma 2 9B (G9)	84.1	84	48.2	121	47.0	124	54.7	95	78.3	71	67.1	73	90.5	72	49.3	123	64.9	95
Small models (avg)	78.4	53	45.5	79	41.5	84	55.4	72	74.2	46	66.1	46	88.2	46	47.2	71	62.1	62
Llama 3.3 70B (L70)	96.0	362	87.6	528	72.9	672	83.9	512	84.8	382	82.8	374	96.5	342	60.4	540	83.1	464
$\mathbf{AmBR_1}$: L8 + G9, L70	95.3	201	87.2	539	72.6	619	83.9	466	83.1	196	77.4	222	95.2	149	57.1	412	81.5	350
$\mathbf{AmBR_2}$: L8 + M8, L70	95.1	197	86.7	498	72.6	600	83.1	422	82.3	182	77.1	198	95.2	123	58.9	402	81.4	328
$\mathbf{AmBR_3}$: M8 + G9, L70	92.3	221	81.1	527	71.7	635	81.5	451	83.1	200	76.6	223	94.8	158	58.5	448	79.9	358
$\mathbf{AmBR_4}$: L8 + P4, L70	95.6	210	87.0	515	72.5	649	83.1	449	82.4	184	75.4	195	94.8	133	58.2	426	81.1	345
AmBR 1–4 avg (L70)	94.6	207	85.5	520	72.4	626	82.9	447	82.7	190	76.6	210	95.0	141	58.2	422	81.0	345
GPT OSS 120B (G120)	91.0	284	83.4	540	82.0	674	90.2	520	85.5	267	79.9	397	94.9	252	61.3	412	83.5	418
$\mathbf{AmBR_5}$: L8 + G9, G120	93.3	184	84.0	547	83.3	620	88.6	470	83.4	169	76.3	229	94.8	138	57.8	354	82.7	339
$\mathbf{AmBR_6}$: L8 + M8, G120	92.9	172	84.0	506	82.4	602	88.6	427	82.2	148	75.7	205	94.9	109	59.8	331	82.6	312
$\mathbf{AmBR_7}$: M8 + G9, G120	91.0	200	79.6	535	81.8	636	87.8	455	83.9	171	75.3	231	94.6	144	59.2	384	81.6	345
$\mathbf{AmBR_8}$: L8 + P4, G120	93.0	184	84.8	524	82.2	651	88.2	453	83.1	152	74.7	202	93.7	119	59.4	350	82.4	329
AmBR 5–8 avg (G120)	92.6	185	83.1	528	82.4	627	88.3	451	83.1	160	75.5	217	94.5	128	59.0	355	82.3	331

Table 16: Results across the same eight benchmarks as Table 1, under the Least-to-Most (LtM) prompting strategy. Each cell reports accuracy | inference cost, where higher accuracy is better and lower cost is better.

Model / Specification	GSM8K		P2		MATH		AQuA		CSQA		SQA-		SQA+		SciNLI		Average	
	Acc	Cost	Acc	Cost	Acc	Cost	Acc	Cost	Acc	Cost	Acc	Cost	Acc	Cost	Acc	Cost	Acc	Cost
Phi-3 Mini 4B (P4)	73.1	57	34.1	80	29.1	81	53.1	65	70.4	42	64.5	48	85.0	48	46.3	71	57.0	61
Ministral 8B (M8)	76.6	56	41.3	80	46.5	75	53.5	71	68.4	44	67.4	45	84.6	46	48.0	67	60.8	61
Llama 3.1 8B (L8)	79.9	42	45.8	65	43.0	78	50.0	81	73.5	29	68.1	34	81.2	33	41.1	50	60.3	51
Gemma 2 9B (G9)	85.1	97	47.9	141	44.1	131	56.7	112	77.9	76	71.0	82	88.9	87	44.9	143	64.6	109
Small models (avg)	78.7	63	42.3	91	40.7	91	53.3	82	72.5	48	67.8	52	84.9	54	45.1	83	60.7	71
Llama 3.3 70B (L70)	95.2	410	88.0	616	69.3	722	81.5	625	84.8	351	81.2	385	96.4	337	59.7	567	82.0	502
$\mathbf{AmBR_1}$: L8 + G9, L70	94.8	240	85.4	624	69.2	687	81.1	562	83.8	193	78.9	235	94.9	191	58.2	577	80.8	414
$\mathbf{AmBR_2}$: L8 + M8, L70	94.9	226	86.0	591	68.9	625	80.7	526	83.2	192	77.1	203	93.0	155	58.6	460	80.3	372
$\mathbf{AmBR_3}$: M8 + G9, L70	94.6	268	84.5	636	68.5	662	79.9	528	83.9	225	78.5	241	94.3	193	57.1	509	80.2	408
$\mathbf{AmBR_4}$: L8 + P4, L70	94.8	240	86.3	613	69.0	718	79.5	530	83.4	181	76.4	217	94.3	161	58.0	478	80.2	392
AmBR 1–4 avg (L70)	94.8	243	85.6	616	68.9	673	80.3	537	83.6	198	77.7	224	94.1	175	58.0	506	80.4	396
GPT OSS 120B (G120)	87.3	361	79.9	602	83.8	803	87.8	565	85.3	335	78.3	470	93.9	324	62.2	594	82.3	507
$\mathbf{AmBR_5}$: L8 + G9, G120	92.6	228	80.9	615	84.3	741	87.4	526	84.8	189	77.7	261	93.9	189	61.0	595	82.8	418
$\mathbf{AmBR_6}$: L8 + M8, G120	90.8	210	80.4	581	84.0	678	85.8	490	83.3	186	75.7	231	92.0	152	60.0	476	81.5	376
$\mathbf{AmBR_7}$: M8 + G9, G120	90.5	254	79.1	627	83.5	713	85.8	495	83.7	220	76.3	267	93.6	191	59.6	523	81.5	411
$\mathbf{AmBR_8}$: L8 + P4, G120	90.8	223	81.2	602	83.4	781	85.8	493	83.5	176	75.0	247	93.3	158	60.9	495	81.7	397
AmBR 5–8 avg (G120)	91.2	229	80.4	606	83.8	728	86.2	501	83.8	193	76.2	251	93.2	172	60.4	522	81.9	400

Both alternative prompts increase cost without a matching accuracy gain, for small models and fallbacks alike. Switching from CoT to PnS raises the small models' average cost from 45 to 62 ($+37.8\%$) while accuracy falls by 2.7 points (64.8→62.1); switching to LtM raises cost further to 71 ($+57.8\%$) with accuracy falling by 4.1 points (64.8→60.7). The same pattern holds for L70: cost rises from 384 (CoT) to 464 under PnS ($+20.8\%$) and 502 under LtM ($+30.7\%$), for accuracy changes of only $+0.7$ and -0.4 points respectively. Longer prompts do not translate into more accurate small-model or fallback reasoning here; they largely just cost more.

AmBR preserves its cost advantage over the fallback under both alternative prompts. Under L70, AmBR's accuracy shortfall relative to the fallback is -1.4 points under CoT, -2.1 under PnS, and -1.6 under LtM, while its cost saving relative to L70 is -30.7%, -25.6%, and -21.1% respectively. Under G120, the shortfall is -0.8, -1.2, and -0.4 points, with cost savings of -22.7%, -20.8%, and -21.1%. In every case, AmBR stays within roughly two points of its fallback while saving at least a fifth of its inference cost, regardless of which prompt is used for the underlying models.

Longer input prompts are a contributing factor to the added cost. Table 17 reports mean input length in tokens for each benchmark under all three prompts. Relative to CoT, PnS increases input

Table 17: Mean input length (tokens) per benchmark under each prompting strategy, and the ratio to CoT length.

Benchmark	CoT	PnS	LtM	PnS/CoT	LtM/CoT
GSM8K	157.7	183.7	245.7	1.165	1.558
P2	195.2	221.2	280.2	1.133	1.436
MATH	190.5	216.5	275.5	1.136	1.446
AQuA	165.3	199.3	258.3	1.206	1.563
CSQA	149.4	165.4	224.4	1.107	1.502
SQA−	135.3	139.3	198.3	1.030	1.466
SQA+	183.9	187.9	246.9	1.022	1.343
SciNLI	254.8	303.8	365.8	1.192	1.436
Average	*179.0*	*202.1*	*261.9*	*1.129*	*1.463*

length by 2–21% across benchmarks ($1.129\times$ overall), and LtM by 34–56% ($1.463\times$ overall). AQuA shows the largest relative increase under both alternative prompts, and SQA+ the smallest. This indicates that longer input prompts contribute to the cost increases observed in Tables 15 and 16.

G Implementation and Cost Details

G.1 Model Costs

Table 18: Token pricing (in $ per 1M tokens) for model inputs and outputs. Gemini 2.5 Pro price points are based on prompts shorter than 200k tokens, as is the case across our tested benchmarks. The prices for Gemini 2.5 Pro are sourced from *https://ai.google.dev/gemini-api/docs/pricing*, up to date as of November 18th 2025. Prices for other models are as shown on *https://groq.com/pricing* up to date as of the same date. Gemma 2 9B price is also sourced from Groq, prior to its deprecation from the provider in October 2025. Phi-3-Mini is priced based on the provider Fireworks AI *https://fireworks.ai/pricing*, up to date as of January 18th 2026. Ministral-8B-2410 is priced based on *https://pricepertoken.com/pricing-page/model/mistral-ai-ministral-8b*, up to date as of August 11th 2026. Qwen 3 8B pricing is taken from *https://openrouter.ai/qwen/qwen3-8b* and is up to date to May 4th 2026.

Model	Input Cost ($/1M tokens)	Output Cost ($/1M tokens)
Phi-3-Mini 4B	0.10	0.10
Llama 3.1 8B	0.05	0.08
Ministral-8B-Instruct-2410	0.10	0.10
Gemma 2 9B	0.20	0.20
Qwen 3 8B	0.05	0.40
Llama 3.3 70B	0.59	0.79
GPT OSS 120B	0.15	0.60
Gemini 2.5 Pro	1.25	10.00

Table 18 reports the token-level pricing used to compute inference costs throughout our experiments. We acknowledge that API prices are subject to change over time: providers regularly revise their pricing in response to competition, model updates, and infrastructure improvements, and several models used in this work have already been deprecated or repriced by their respective providers since our experiments were conducted. As a result, the absolute cost figures reported in this paper should be interpreted as estimates grounded in a specific pricing snapshot rather than stable constants. The *relative* cost reductions achieved by AmBR—the primary quantity of interest—are more robust to price fluctuations, since they depend on the *ratio* of routing-pair cost to fallback cost rather than any single price point. Nevertheless, we recommend that practitioners re-estimate inference costs at deployment time using current provider pricing, as shifts in the cost structure between routing models and the fallback can affect the efficiency gains AmBR provides in practice.

G.2 AmBR's Impact on Latency

Measuring wall-clock inference latency in cloud-based settings is inherently noisy: observed throughput varies with server load, hardware allocation, request batching, and network conditions, making precise latency comparisons difficult to reproduce. With this caveat in mind, we use provider-reported throughput figures as a basis for qualitative analysis. Groq reports output throughputs of approximately 560, 560, 280, and 500 tokens per second for Llama 3.1 8B, Gemma 2 9B, LLaMA 3.3 70B, and GPT OSS 120B, respectively.[1] Throughput figures for Ministral-8B and Phi-3-Mini-4B are not publicly reported by Groq.

G.2.1 Parallel Inference Setup

AmBR's latency profile relative to the all-at-large baseline depends on whether the small-model pair reaches agreement. In the agreement case—where the fallback is never invoked—AmBR resolves queries using only the two small models, which run in parallel and are substantially faster than the large fallback. For the L70-backed setting, small models operate at roughly $2\times$ the throughput of L70 (560 vs. 280 tokens per second), meaning AmBR is approximately twice as fast as the all-at-large baseline when agreement is reached. For G120-backed variants, small-model throughput (560 tokens per second) is comparable to G120 (500 tokens per second), yielding similar latency on agreement.

[1] `https://console.groq.com/docs/models`

In the disagreement case—where the fallback is invoked—queries must pass through the small models before escalating to the large model, so AmBR incurs the latency of both stages sequentially. For the L70-backed setting, this results in approximately $1.5\times$ the latency of calling L70 directly, since the small models complete in roughly half the time of L70. For the G120-backed setting, where small and large model throughputs are more similar, fallback escalation results in approximately $2\times$ the latency of querying G120 alone. In both cases, the latency overhead is incurred only on the fraction of queries that escalate, which AmBR minimises through its agreement-based routing.

In the commercial setting, the latency picture is particularly favourable, with AmBR Pro delivering an overall expected speedup of approximately $1.4\times$ relative to Gemini 2.5 Pro. Gemini 2.5 Pro generates approximately 120 tokens per second as measured by independent benchmarking,[2] substantially slower than the AmBR Pro routing models, whose parallel execution is bottlenecked by the slower of the two at 280 tokens per second. In the agreement case—which accounts for over 70% of queries across benchmarks—AmBR Pro resolves queries at approximately $2.3\times$ the speed of the Gemini 2.5 Pro baseline. In the escalation case, the routing stage adds a sequential overhead: for a response of N output tokens, total latency is $N/280 + N/120 \approx N/84$ seconds, compared to $N/120$ for Gemini alone, resulting in approximately $1.4\times$ the latency of a direct Gemini call. Weighting by the empirical agreement rate (70% agreement, 30% escalation), the expected throughput of AmBR Pro across all queries is approximately 165 tokens per second, resulting in an average $1.4\times$ speedup. AmBR's latency impact is thus most favourable when the routing models are substantially faster than the fallback—precisely the regime in which AmBR Pro operates—and is bounded in the worst case by a modest $1.4\times$ overhead on the minority of queries that escalate.

G.2.2 Sequential Inference Setup

Under sequential small-model execution, AmBR's latency profile shifts unfavourably relative to the all-at-large baseline, and the degree of degradation depends on whether agreement is reached. In the agreement case, AmBR resolves queries by running the two small models one after the other, incurring twice the per-model latency compared to the parallel regime. For the L70-backed setting, where each small model operates at 560 tokens per second, sequential execution yields an effective throughput of 280 tokens per second—identical to L70 itself—meaning AmBR offers no latency advantage over the all-at-large baseline on agreement. For G120-backed variants, sequential small-model execution (280 tokens per second effective) is substantially slower than G120 (500 tokens per second), resulting in approximately $1.8\times$ the latency of a direct G120 call even when the fallback is never invoked.

In the disagreement case, sequential execution compounds with fallback escalation, producing more severe latency overhead than in the parallel regime. For the L70-backed setting, the sequential small-model stage completes at the same speed as L70 itself, so adding the fallback call doubles total latency—approximately $2\times$ that of calling L70 directly, compared to $1.5\times$ under parallel execution. For the G120-backed setting, sequential routing (N/280 seconds) followed by G120 fallback (N/500 seconds) yields a combined latency of approximately N/179 seconds, roughly $2.8\times$ that of querying G120 alone, compared to $2\times$ in the parallel case.

In the commercial setting, sequential execution substantially reduces but does not fully eliminate AmBR Pro's latency advantage. Under sequential execution, the two routing models complete in $N/280 + N/500 \approx N/180$ seconds combined, compared to $N/120$ for Gemini 2.5 Pro alone. In the agreement case (over 70% of queries), AmBR Pro therefore remains approximately $1.5\times$ faster than the baseline, though this is reduced from the $2.3\times$ speedup under parallel execution. In the escalation case, the sequential routing stage adds overhead on top of the Gemini call: total latency is $N/180 + N/120 = N/72$ seconds, approximately $1.7\times$ that of a direct Gemini call. Weighting by the empirical agreement rate (70% agreement, 30% escalation), the expected throughput of AmBR Pro under sequential routing is approximately 124 tokens per second—roughly on par with Gemini's 120 tokens per second—yielding an expected throughput of approximately $1.03\times$, essentially identical to the all-at-large baseline. This analysis highlights that the substantial latency gains reported for AmBR Pro are contingent on parallel small-model execution; sequential routing reduces average speedup from $1.4\times$ to near parity, with the benefit concentrated entirely in the agreement case.

[2] https://artificialanalysis.ai/models/gemini-2-5-pro

Table 19: Accuracy (%), mean ± std over 3 seeds, across eight benchmarks, with computed group averages and a per-row Average column.

Model / Spec.	GSM8K	P2	MATH	AQuA	CSQA	SQA-	SQA+	SciNLI	Average
P4	78.65±1.26	47.32±1.01	40.35±0.93	45.41±2.19	71.50±0.47	65.07±1.37	86.95±1.07	45.15±1.32	60.05
M8	80.06±0.89	48.65±2.88	50.94±0.95	59.71±6.12	68.47±2.44	68.41±1.60	89.08±1.52	48.97±0.23	64.29
L8	82.13±0.56	51.77±3.81	48.46±1.98	62.47±0.49	72.84±1.44	70.31±0.72	89.86±0.70	44.59±0.79	65.30
G9	86.99±0.50	48.69±0.34	48.69±0.34	62.47±1.62	76.69±0.81	72.93±1.02	91.90±0.18	50.57±0.08	67.37
Small models (avg)	81.96	49.11	47.11	57.52	72.38	69.18	89.45	47.32	64.25
L70	95.93±0.29	88.07±0.13	74.03±0.19	82.94±1.30	84.17±0.31	82.48±0.65	96.60±0.30	57.83±0.50	82.76
AmBR_1	94.90±0.31	85.27±0.41	73.66±0.28	82.41±1.30	83.18±0.40	79.04±0.82	95.63±0.31	56.27±0.25	81.30
AmBR_2	94.79±0.18	87.15±0.20	73.69±0.39	82.02±1.30	81.93±0.50	78.07±0.18	95.20±0.21	58.43±0.28	81.41
AmBR_3	94.24±0.64	84.41±0.21	73.41±0.18	83.46±1.47	82.69±0.14	76.86±1.25	94.76±0.43	56.61±0.53	80.80
AmBR_4	94.90±0.31	87.25±0.25	73.45±0.26	81.23±0.74	82.09±0.34	75.93±0.99	95.15±0.14	57.55±0.12	80.94
AmBR 1-4 avg (L70)	94.71	86.02	73.55	82.28	82.47	77.47	95.19	57.22	81.11
G120	90.67±0.77	85.37±1.30	82.85±0.76	89.76±0.32	85.42±0.20	79.18±0.54	95.63±0.52	62.53±0.58	83.93
AmBR_5	93.28±0.22	84.53±0.98	83.20±0.07	88.45±0.19	83.97±0.87	77.15±0.48	95.29±1.00	59.42±0.25	83.16
AmBR_6	92.70±0.78	86.12±0.64	83.17±0.19	87.93±0.67	83.10±0.43	75.55±0.74	95.05±0.48	61.24±0.36	83.11
AmBR_7	92.27±0.61	83.73±0.59	82.62±0.28	89.11±1.13	83.62±0.24	76.56±0.78	94.66±0.18	59.77±0.30	82.79
AmBR_8	92.54±0.04	85.83±0.74	82.70±0.19	86.88±0.98	83.32±0.54	73.90±0.34	94.57±0.48	61.07±0.49	82.60
AmBR 5-8 avg (G120)	92.70	85.05	82.92	88.09	83.50	75.79	94.89	60.38	82.92

G.3 Hybrid LLM Implementation on Reasoning Benchmarks

We implement HybridLLM following Ding et al. (2024) by fine-tuning a router based on DeBERTa-v3-Large (He et al., 2023) to decide whether each query should be answered by L8 or escalated to L70. Training set sizes range from roughly 5K to 8K examples, depending on the benchmark. Our main modification is the construction of the routing target. Whereas the original HybridLLM framework uses a semantic similarity-based quality score, we define a task-specific binary routing label based on answer correctness: an example is labeled for escalation to L70 if L8 answers it incorrectly and L70 answers it correctly; all other examples are labeled for routing to L8. This target directly reflects the cost-saving objective in our setting, where the router should invoke L70 only when it is expected to improve the final answer.

Where benchmark-specific training sets are available, we train the router on the corresponding training split. For P2, which does not provide a training set, we use the router trained on GSM8K because P2 is derived from GSM-style arithmetic word problems. For SciNLI, we train on a stratified subset of 8,000 training examples to limit evaluation cost. All costs reported in Table 2 include only inference-time model calls and assume zero routing overhead; they therefore do not include the cost of generating routing labels or fine-tuning the router.

H Statistical Analysis of Results

To assess the stability of our main results under repeated sampling, we re-ran the full evaluation suite in Table 1 for three independent seeds, covering all small models, both fallback models, and all eight AmBR configurations across all eight benchmarks. Table 19 reports mean accuracy and standard deviation, and Table 20 reports mean inference cost and standard deviation, both computed over the three seeds.

Results are stable across seeds. Across all eight benchmarks and all fourteen model configurations, accuracy standard deviations remain below 1.5 percentage points in the large majority of cells, with a small number of exceptions on the more volatile benchmarks (AQuA and SQA-, where std reaches 1.3–2.2 points for some configurations, and M8 on AQuA at 6.12 points, the single largest deviation observed). Cost standard deviations follow a similar pattern, remaining under 5% of the mean cost for most cells, with the notable exception of L8 on MATH (std of 14.5 against a mean of 87.4) and the AQuA column more broadly, where cost is comparatively more sensitive to seed-level variation across almost every model. These exceptions are consistent with the properties of the benchmarks themselves rather than instability in AmBR: AQuA's multiple-choice format and MATH's longer, more variable solution lengths naturally introduce more run-to-run variance in both correctness and token count.

Table 20: Inference cost, mean ± std over 3 seeds, across eight benchmarks, with computed group averages and a per-row Average column.

Model / Spec.	GSM8K	P2	MATH	AQuA	CSQA	SQA-	SQA+	SciNLI	Average
P4	31.4±0.3	47.3±0.3	64.8±1.5	30.3±3.3	18.8±1.7	22.2±0.5	27.5±0.4	45.8±0.4	36.0
M8	33.8±2.5	52.1±5.0	61.8±1.3	51.5±3.4	21.3±5.0	25.5±1.6	29.1±1.3	44.2±0.3	39.9
L8	29.9±0.3	46.8±2.7	87.4±14.5	54.2±8.3	24.8±0.5	27.6±0.8	24.4±0.3	36.4±0.6	41.4
G9	59.2±0.1	90.2±0.0	108.6±0.8	72.0±0.5	60.2±0.6	43.6±1.5	55.1±0.3	89.6±0.1	72.3
Small models (avg)	38.6	59.1	80.7	52.0	31.3	29.7	34.0	54.0	47.4
L70	272.4±4.5	424.1±4.3	661.9±3.9	445.1±62.0	348.1±2.3	314.7±2.8	292.8±0.8	418.0±5.4	397.1
AmBR_1	146.1±1.4	410.6±13.0	639.3±8.5	333.8±38.2	173.0±5.7	148.2±4.1	114.4±3.1	343.4±2.8	288.6
AmBR_2	136.4±3.7	381.9±14.2	589.3±6.1	330.3±56.2	162.3±11.6	137.1±4.6	92.7±3.1	327.5±3.3	269.7
AmBR_3	151.7±4.0	414.3±8.8	597.4±5.1	351.8±48.9	186.1±11.6	138.3±5.6	115.7±3.0	339.6±2.5	286.9
AmBR_4	137.2±2.5	378.8±14.9	619.1±11.9	356.8±47.0	147.8±4.9	135.2±2.8	96.3±4.3	349.6±4.2	277.6
AmBR 1-4 avg (L70)	142.8	396.4	611.3	343.2	167.3	139.7	104.8	340.0	280.7
G120	252.0±4.1	474.2±1.8	752.8±47.3	309.9±30.6	135.7±25.5	296.7±29.9	200.3±16.9	424.2±16.9	355.7
AmBR_5	141.9±1.1	442.9±12.3	700.6±26.1	270.7±23.2	119.4±7.5	143.5±3.1	103.2±1.7	346.8±11.8	283.6
AmBR_6	131.0±3.3	415.3±13.5	650.2±25.3	261.4±33.6	91.0±1.6	131.7±3.8	80.0±0.7	331.4±14.7	261.5
AmBR_7	147.3±3.5	446.4±8.7	656.3±33.3	281.9±25.9	121.8±0.9	133.7±0.5	105.5±1.2	342.6±8.1	279.4
AmBR_8	131.5±1.8	412.4±14.3	683.5±24.7	274.0±27.2	84.0±6.3	130.2±7.3	82.1±2.8	353.6±12.7	268.9
AmBR 5-8 avg (G120)	137.9	429.2	672.6	272.0	104.0	134.8	92.7	343.6	273.4

The accuracy–cost trade-off is preserved under repeated sampling. Averaged across benchmarks, AmBR 1–4 trades 1.65 accuracy points for a 29.3% reduction in cost relative to L70 (81.11 vs. 82.76 accuracy; 280.7 vs. 397.1 cost), and AmBR 5–8 trades 1.01 accuracy points for a 23.1% reduction relative to G120 (82.92 vs. 83.93 accuracy; 273.4 vs. 355.7 cost). Both figures are consistent with the single-seed results reported in Table 1 (1.4-point drop and 31% cost reduction under L70; 0.8-point drop and 23% cost reduction under G120), and the relative ranking of AmBR variants is preserved across seeds: AmBR_2 remains the strongest and cheapest L70-backed variant, and AmBR_6 remains the cheapest G120-backed variant, in both the single-seed and the 3-seed results.

Cost reductions are statistically significant; accuracy differences are small. To test whether the differences between each AmBR configuration and its fallback model exceed seed-level variation, we apply a two-sided paired t-test to the per-seed Average values (paired by seed, $n = 3$), with Holm correction over the eight comparisons for each metric. All eight AmBR configurations are significantly cheaper than their fallback after correction (Holm-adjusted $p \leq 0.010$ for AmBR 1–4 and $p = 0.035$ for AmBR 5–8, respectively). For accuracy, none of the differences between AmBR 5–8 and G120 is significant ($p = 0.052$–0.152 before correction), although AmBR_7 and AmBR_8 are borderline. The differences between AmBR 1–4 and L70 are significant before correction ($p = 0.005$–0.022), and those of AmBR_1 and AmBR_4 remain significant after correction (Holm-adjusted $p = 0.040$). Welch's t-test yields the same conclusion for cost and a similar picture for accuracy. With three seeds, the tests have limited power, so the absence of significance for AmBR 5–8 should not be read as evidence of equivalence. Overall, AmBR's cost savings are robust to seed-level variation, while its accuracy cost is small and statistically detectable only for some L70-backed configurations.

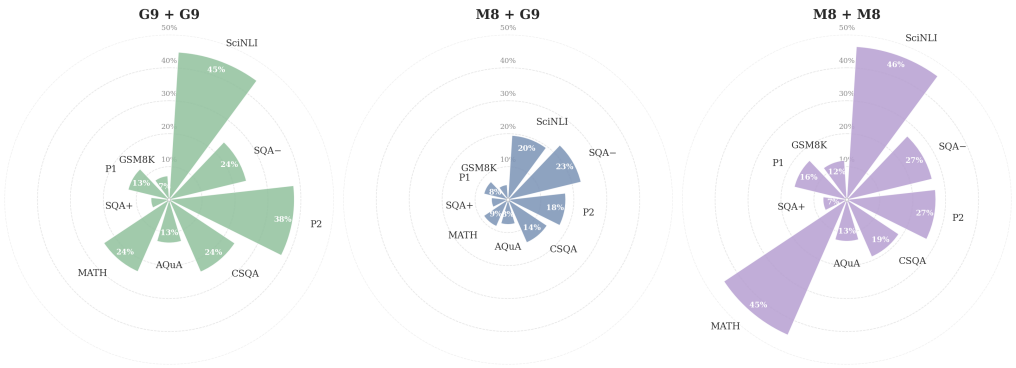


Figure 6: Agreement error rates across benchmarks for within-model and cross-model agreement. Each radial bar reports the percentage of agreed predictions that are incorrect; lower values indicate more reliable agreement. The cross-model pair M8+G9 substantially reduces agreement errors relative to the corresponding within-model pairs G9+G9 and M8+M8, showing that agreement between distinct model families is a stronger correctness signal than repeated agreement from the same model.

I Theoretical Analysis

Here we provide the proof for **Proposition 3.1.**

Proof. We decompose the agreement event into two mutually exclusive cases:

$$A = (C_1 \cap C_2) \ \cup \ (A \cap \neg C_1 \cap \neg C_2).$$

If the two models agree, they are either both correct, or they coincide on the same incorrect answer. Therefore,

$$\Pr(A) = \Pr(C_1 \cap C_2) + \Pr(A \cap \neg C_1 \cap \neg C_2).$$

Under conditional independence of the correctness events,

$$\Pr(C_1 \cap C_2) = p_1 p_2.$$

By assumption (2),

$$\Pr(A \cap \neg C_1 \cap \neg C_2) = \Pr(A \mid \neg C_1, \neg C_2)\ \Pr(\neg C_1, \neg C_2) \leq \rho(1-p_1)(1-p_2).$$

Combining these two bounds,

$$\Pr(A) \leq p_1 p_2 + \rho(1-p_1)(1-p_2).$$

Finally, by Bayes' theorem,

$$\Pr(C_1 \cap C_2 \mid A) = \frac{\Pr(C_1 \cap C_2)}{\Pr(A)} \geq \frac{p_1 p_2}{p_1 p_2 + \rho(1-p_1)(1-p_2)},$$

which completes the proof. □

We now provide a more complete theoretical treatment of the agreement signal under progressively relaxed assumptions, moving from the simplest binary setting to the general multiclass case and finally to the correlated-error regime.

Binary classification. Let the true label be $Y \in \{0, 1\}$ and let $\hat{Y}_1, \hat{Y}_2$ denote the outputs of the two small models. Assume model i has accuracy $p_i = \Pr(\hat{Y}_i = Y)$, and that errors are conditionally independent given the true label, i.e., $\hat{Y}_1 \perp \hat{Y}_2 \mid Y$.

When $\hat{Y}_1 = \hat{Y}_2$, the agreed-upon label is correct if and only if both models are correct. Since in the binary case both models being wrong forces them to agree on the same incorrect label, the agreement event satisfies

$$\Pr(\text{agree}) = p_1 p_2 + (1-p_1)(1-p_2),$$

and the conditional precision of the agreement prediction is

$$\Pr\big(\text{correct} \mid \hat{Y}_1 = \hat{Y}_2\big) = \frac{\Pr(\text{both correct})}{\Pr(\text{agree})} = \frac{p_1 p_2}{p_1 p_2 + (1-p_1)(1-p_2)}.$$

This confirms that agreement is strong evidence of correctness whenever both models perform better than chance and their errors are not too correlated.

When the two models disagree, $\hat{Y}_1 \neq \hat{Y}_2$, they cannot both be correct in the binary case, so at least one is always wrong. Without additional information, the posterior probability that model i is the correct one is given by the reliability-weighted rule

$$\Pr\big(\hat{Y}_1 \text{ correct} \mid \hat{Y}_1 \neq \hat{Y}_2\big) = \frac{p_1(1-p_2)}{p_1(1-p_2) + (1-p_1)p_2},$$

and symmetrically for model 2. This expression reduces to $\frac{1}{2}$ when $p_1 = p_2$, reflecting the absence of a tiebreaker when the two models are equally reliable — which is precisely the regime in which escalation to a large model L is most warranted.

Multiclass generalization. When there are $K > 2$ classes, two wrong models need not agree: each may err in a different direction. Under the simplifying assumption that a model which errs selects each of the $K-1$ incorrect labels uniformly at random with probability $\frac{1}{K-1}$, the relevant probabilities become

$$\Pr(\text{agree} \cap \text{correct}) = p_1 p_2, \qquad \Pr(\text{agree} \cap \text{wrong}) = (1-p_1)(1-p_2) \cdot \frac{1}{K-1},$$

yielding the conditional precision

$$\Pr(\text{correct} \mid \text{agree}) = \frac{p_1 p_2}{p_1 p_2 + \dfrac{(1-p_1)(1-p_2)}{K-1}}.$$

Crucially, this precision is increasing in K: as the number of classes grows, the probability that two independently erring models coincide on the same wrong label decreases, making agreement an even more informative correctness signal. This provides theoretical justification for the strong empirical performance of AmBR on harder, multi-way reasoning benchmarks.

J Relationship to Self-Consistency, Ensembles, and Hybrid LLM

AmBR is closely related to prior approaches that leverage multiple predictions at inference time, such as self-consistency and model ensembles, but differs fundamentally in both objective and compute allocation. Self-consistency samples multiple reasoning trajectories from a single model and aggregates them (e.g., via majority voting), improving robustness at the cost of linearly increasing inference compute. In contrast, AmBR uses one sample per model and exploits inter-model agreement rather than intra-model diversity. This distinction is critical: samples from the same model often share correlated failure modes, whereas independently trained models are less likely to converge on the same incorrect answer, making agreement a higher-precision signal.

Compared to classical ensembles, which combine predictions from multiple models (e.g., averaging or voting) and always incur the cost of evaluating all models, AmBR adopts a selective escalation strategy. Two small models are queried upfront, and the expensive model is invoked only on disagreement. As a result, AmBR can be viewed as a precision-oriented ensemble with conditional compute, trading recall for efficiency: it accepts only high-confidence (agreement) predictions and defers ambiguous cases to a larger model. This contrasts with ensembles that aim to improve average accuracy through aggregation but do not reduce compute.

Hybrid LLM (Ding et al., 2024) requires an explicit, learned estimate of $\Pr[H(x) \geq 0]$ to gate queries. AmBR replaces this learned gate with a training-free binary signal: agreement is high precisely when both models are confident and consistent — the same regime in which $\Pr[H(x) \geq 0]$ would be high for a well-trained router. In this sense, AmBR can be understood as a zero-shot surrogate for the Hybrid LLM router, one that exploits model diversity instead of labeled data to estimate query difficulty. This makes AmBR immediately applicable to any pair of off-the-shelf models without modification, fine-tuning, or access to model internals.