ORCHSLM: PROBING THE DYNAMICS OF SMALL LANGUAGE MODEL ORCHESTRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Although large language models (LLMs) have demonstrated remarkable capabilities, their reliance on cloud-scale infrastructure poses fundamental challenges for deployment in agentic pipelines, including latency, privacy, connectivity, and substantial computational cost. Small language models (SLMs) offer a compelling alternative: recent studies suggest that many repetitive and narrowly scoped subtasks in agentic workloads may be better served by specialized SLMs than by monolithic LLMs. However, the limited capacity and context windows of SLMs can constrain long-horizon reasoning and interaction-heavy orchestration strategies such as iterative verification and debate. This motivates a complementary, non-interactive paradigm in which heterogeneous SLMs independently generate candidate solutions and a router orchestrates their cached samples without further model interaction. To further understand the mechanisms of such orchestration, we introduce OrchSLM, a routing framework that unifies existing non-interactive orchestration methods and exposes their underlying design choices as controllable parameters. Using OrchSLM as a systematic probe, we reveal how orchestration behavior emerges from diverse knobs, including the task structure, model-pool composition, and multi-agent consensus.

1 INTRODUCTION

Large language models have demonstrated remarkable progress across reasoning, coding, tool use, and increasingly complex agentic tasks (OpenAI, 2023; Touvron et al., 2023; Grattafiori et al., 2024; DeepSeek-AI, 2025; Snell et al., 2024b). Scaling model capacity and inference-time computation has further extended their ability to solve challenging problems (Kaplan et al., 2020; Hoffmann et al., 2022; Lightman et al., 2024; Brown et al., 2024; Snell et al., 2024a). However, these capabilities come with substantial computational and deployment costs, particularly when language models must be invoked repeatedly as components of agentic pipelines. As an alternative, small language models are increasingly emerging as low-cost, deployable, and heterogeneous reasoning engines (Wang et al., 2024a; Belcak et al., 2025). Their modest computational footprint makes it practical to query multiple models at inference time, raising the possibility of treating a collection of SLMs as a pool of complementary specialists. However, most orchestration strategies developed for larger language models rely heavily on interaction: models exchange natural-language messages, critique intermediate solutions, synthesize one another's responses, or verify candidate answers before a final decision is made (Wang et al., 2024b; Du et al., 2023; Lifshitz et al., 2025). For SLMs, such interaction can be costly relative to the models themselves and relies on deliberation and self-correction capabilities that remain limited (Taubenfeld et al., 2024; Huang et al., 2024; Liu et al., 2023; Fu et al., 2025).

This motivates a complementary, *non-interactive*[1] setting: each model independently generates a set of cached samples, and final answers are obtained by combining the resulting answer distributions at inference time. Despite its simplicity, a systematic understanding of such non-interactive orchestration still remain unknown. Existing methods make fundamentally different implicit assumptions about what constitutes useful evidence. Query-level routers and cascades choose a model before generation,

[1]We use *non-interactive* to mean that models do not communicate with one another or condition their generations on other models' outputs. Each model generates independently, and cross-model information is combined only by the router after generation, without consuming additional context tokens for inter-model communication.

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.

typically based on predicted expertise, cost, or preference (Chen et al., 2023; Ong et al., 2025; Chen et al., 2024). Voting methods instead treat agreement across independently generated answers as evidence (Wang et al., 2023; Li et al., 2024). Routing-style routing estimates within-model consistency and favors answers from models that repeatedly support the same prediction (Wang et al., 2025). These approaches expose distinct signals (*expertise*, *self-consistency*, and *mutual agreement*) but entangle them through different routing rules.

To understand these signals is important because each can be informative in one regime and misleading in another. A strong expert may provide a reliable default answer, yet expert selection discards corroborating evidence from the rest of the pool. Majority agreement can reveal shared evidence, yet a collection of weaker models may reinforce the same plausible error (Li et al., 2024; Wang et al., 2025). High self-consistency can indicate confidence, but it can also arise from a model that repeatedly makes the same mistake (Wang et al., 2023; 2025). Even the composition of the model pool introduces a nontrivial tradeoff: adding another model may increase the probability that some model produces the correct answer while simultaneously introducing additional incorrect support that makes the answer harder to identify. Consequently, orchestration quality is governed not by the number of available models or the amount of agreement alone, but by how expertise, agreement, and pool composition interact with the structure of the task. This raises the central question of our work: *what governs the dynamics of small language model orchestration?* Rather than searching for a single universally optimal router, we seek to isolate the mechanisms that determine when heterogeneous SLMs complement one another, when collective agreement provides useful evidence, and when adding more models instead makes routing harder.

To probe these tradeoffs systematically, we decompose non-interactive multi-SLM orchestration along three research axis. *(1) Whom should we trust?* How should authority be distributed across models? *(2) How should evidence combine?* How should multiple beliefs form a collective decision? *(3) When does agreement matter?* When does consensus provide evidence beyond individual confidence? Existing routing methods implicitly make different choices along these dimensions. Majority voting weights models equally, accumulates support across the pool, and relies on mutual agreement. Single-best routing places all trust in a fixed expert. Router-style favors the model exhibiting the strongest self-consistency, while cross-support treats probability mass assigned by other models to a candidate answer as question-specific evidence. We make these choices explicit as controllable axes, allowing existing routers and their intermediate variants to be studied within a common design space.

We instantiate this design space as a generalized routing family in which common orchestration methods emerge as exact parameter settings, and use it as a controlled probe of orchestration dynamics. Across MATH-500, GSM8K, GPQA, and MMLU-Pro with a heterogeneous model pool, our systematic analysis reveals distinct dynamics of multi-SLM orchestration: effective routing emerges from the interplay between task structure, model expertise, pool composition, and agreement patterns, rather than from any single source of evidence.

Contributions. (1) We formulate non-interactive multi-SLM orchestration as a controlled design space organized around three coupled mechanisms: *whom to trust*, *how to aggregate evidence*, and *what form of consistency should constitute confidence*. (2) We introduce ORCHSLM, a generalized routing family that exposes these mechanisms as controllable axes and recovers majority voting, self-consistency routing, cross-support, expert-weighted routing, and the single-best limit as specific configurations within a common framework. (3) We use ORCHSLM to systematically probe the dynamics of multi-SLM orchestration across tasks and model pools. Our experiments reveal how expertise, agreement, and pool composition jointly determine routing behavior, including a persistent gap between *oracle coverage* and *routable support*: more models can make the correct answer available without making it easier to identify.

2 PRELIMINARIES AND METHODOLOGY

Motivation. Orchestration is a problem of collective decision-making (Arrow, 1951): given multiple imperfect and heterogeneous reasoners, how should their individual beliefs be transformed into a collective judgment? More voices do not necessarily imply more knowledge, agreement does not necessarily imply truth, and expertise does not necessarily imply correctness on every instance. The challenge is therefore not merely to combine model outputs, but to determine when individual authority should prevail, when evidence should accumulate, and when agreement should be trusted.

We study this problem through three fundamental questions. *Whom should we trust?* Models differ in expertise, raising a tension between democratic aggregation and expert authority. *How should evidence combine?* Support may derive its strength from accumulation across many independent sources, or from the conviction of a particularly reliable one. *When does agreement matter?* Repeated belief can reflect genuine confidence, but consensus can also amplify shared errors; its value therefore depends on who agrees and how their evidence is formed. These questions expose three distinct but coupled mechanisms underlying multi-SLM orchestration: *trust*, *aggregation*, and *consistency*. We next formalize them as controllable axes of a generalized routing family, allowing different orchestration strategies to be studied within a common framework.

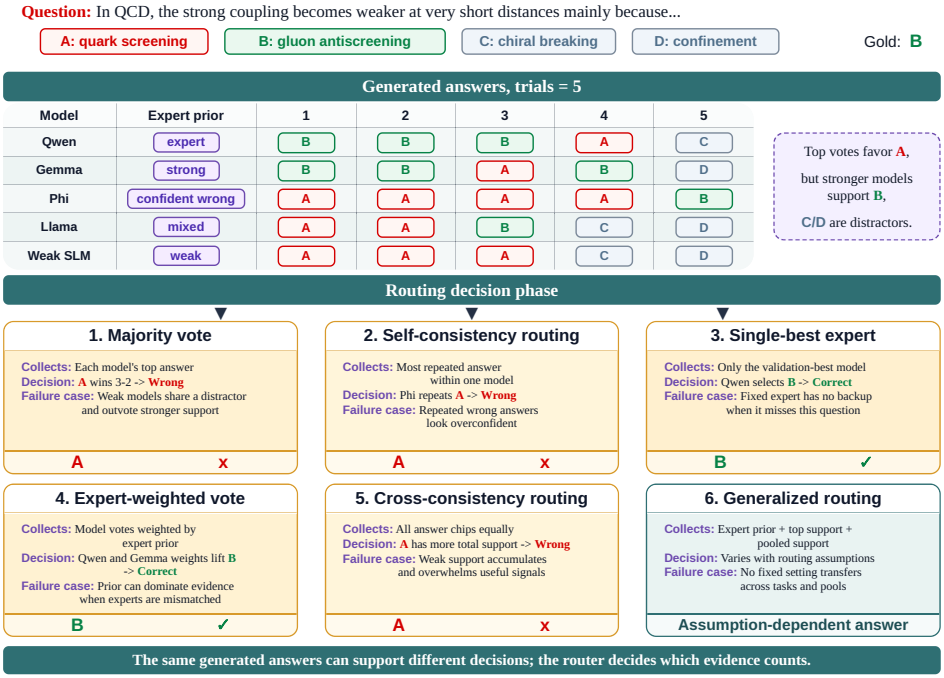

Figure 1: Different routers can pick different answers from the same cached outputs.

2.1 Evaluation of Orchestration

How should collective evidence become a decision, and whose evidence actually shapes that decision? We evaluate orchestration through these two complementary perspectives. The first concerns *decision*: we formalize how trust, aggregation, and agreement determine which answer is selected. The second concerns *influence*: we distinguish between models that are heard by the router and models whose presence actually improves its decision. Together, these views characterize not only what an orchestration system decides, but also how that collective decision emerges. Additional derivations and implementation details are provided in Appendix §A.

From samples to beliefs. For a question x, each model i independently generates K samples. After answer extraction and normalization, these samples induce an empirical distribution over candidate answers:

$$P_{ia}(x) = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}[y_{ik} = a]. \tag{1}$$

Here, $P_{ia}(x)$ represents the empirical support that model i assigns to answer a. Let $a_i^\star = \arg\max_a P_{ia}(x)$ denote model i's most frequent answer, with ties broken deterministically, and define $M_{ia} = \mathbf{1}[a = a_i^\star]$. Thus, P_{ia} captures the model's full answer distribution, while M_{ia} identifies the answer on which its own belief concentrates most strongly.

Table 2: Common routers mapped to $(\alpha, \kappa, \mathrm{Agg})$ in the generalized family.

Method	Examples	Trust	Aggregation	Position in family
Expert selection	Query routing, cascades, model routers	Expert	Single	$(1, \infty, -)$
Majority voting	Self-consistency voting and multi-agent majority	Average	Pooled sum	$(0, 0, -)$
Weighted voting	Cross-validation weighted probabilistic ensemble	Expert	Pooled sum	$(0, \kappa > 0, -)$
Self-consistency	SLM-MUX-style answer-level routing	Average	Max	$(1, 0, \mathrm{Max})$
Weighted consistency	Validation-aware consistency routing	Expert	Max	$(1, \kappa > 0, \mathrm{Max})$
Cross consistency	Answer-level cross-confidence and support variants	Average	Max + Pooled	$(0 < \alpha < 1, 0, \mathrm{Max})$
Generalized routing	This generalized family	Tunable	Tunable	$([0, 1], [0, \infty], \{\mathrm{Sum}, \mathrm{Max}\})$

Trust: whose evidence should count? Models need not be equally reliable. Let acc_i denote an expert prior for model i, estimated from a validation split or supplied as a fixed prior score. We translate these priors into normalized trust weights:

$$r_i(\kappa) = \frac{\max(\mathrm{acc}_i, \epsilon)^\kappa}{\sum_j \max(\mathrm{acc}_j, \epsilon)^\kappa}. \quad (2)$$

The parameter κ determines how concentrated authority is across the model pool. At $\kappa = 0$, every model receives equal weight, corresponding to a democratic prior over expertise. As κ increases, authority shifts toward models with stronger prior performance; in the limit $\kappa \to \infty$, it concentrates on the validation-best model. Thus, κ continuously interpolates between collective equality and expert authority.

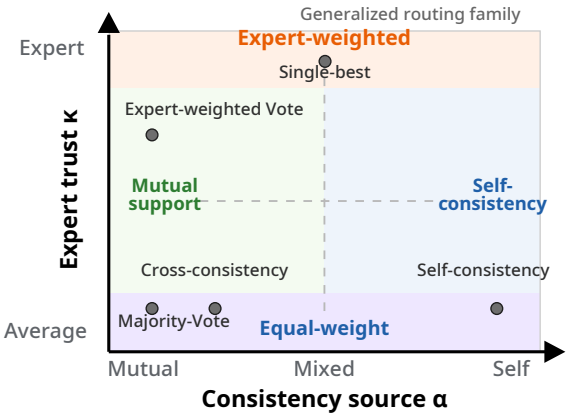

Figure 2: Existing routers as fixed corners of the $(\alpha, \kappa, \mathrm{Agg})$ generalized family; axes are defined in Section 2.1.

Aggregation and agreement: how should evidence combine? For each candidate answer a, we define the routing score

$$W(a) = \alpha \, \mathrm{Agg}_i \left[M_{ia} r_i(\kappa) P_{ia}(x) \right] + (1 - \alpha) \sum_i r_i(\kappa) P_{ia}(x), \quad (3)$$

and select $\hat{a} = \arg\max_a W(a)$. Here, Agg controls how evidence combines across models: Sum accumulates support from multiple models, whereas Max retains the strongest supporter. The parameter α controls the form of agreement, interpolating between support for each model's preferred answer and pooled support over its full answer distribution. Together, $(\kappa, \mathrm{Agg}, \alpha)$ specify whom to trust, how to combine their evidence, and which form of agreement to use.

Influence: being heard versus being helpful. The routing score determines which answer is selected, but does not tell us whether each model helps make that selection correct. We therefore measure model influence through attribution and contribution.

For a dataset D, let $\hat{a}_t$ be the routed answer for question t. We define the attribution of model i as

$$\mathrm{Atr}(i) = \frac{1}{|D|} \sum_{t \in D} P^{(t)}_{i, \hat{a}_t}. \quad (4)$$

Atr measures how strongly a model supports the answers selected by the router. A model with high Atr is frequently *heard*, but its support need not improve accuracy.

To measure whether a model is helpful, we remove it and rerun the router. We define its counterfactual contribution as

$$\mathrm{Ctr}(i) = \mathrm{Acc}(\text{full pool}) - \mathrm{Acc}(\text{pool without } i). \quad (5)$$

Equivalently, if $\hat{a}_t^{(-i)}$ is the routed answer without model i, then

$$\mathrm{Ctr}(i) = \frac{1}{|D|} \sum_{t \in D} \left(\mathbf{1}[\hat{a}_t = y_t] - \mathbf{1}[\hat{a}_t^{(-i)} = y_t] \right). \quad (6)$$

3 EXPERIMENTS

Setup. All routers operate on the same cached generations from a seven-model pool (Qwen2.5-7B-Instruct, Meta-Llama-3.1-8B-Instruct, Llama-3.1-Tulu-3-8B-DPO, DeepSeek-Math-7B-Instruct, Gemma-2-9B-IT, Phi-3.5-mini-instruct, Ministral-8B-Instruct-2410) across four benchmarks: MATH-500 (competition math, one dominant expert), GSM8K (saturated arithmetic), GPQA (hard, balanced science), and MMLU-Pro (broad multi-topic science). Implementation details are at Appendix §A.

3.1 SCALING UP TASK STRUCTURES

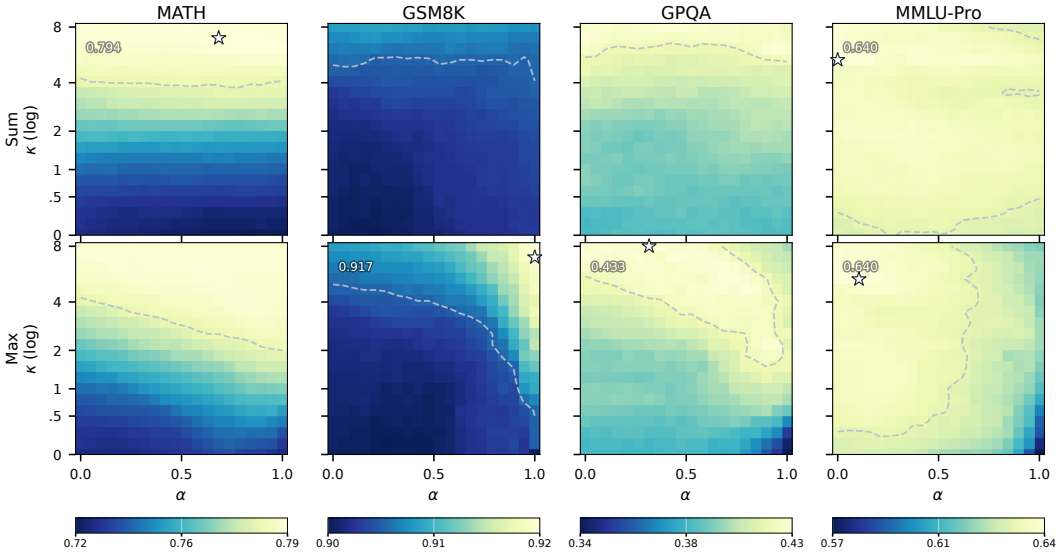

Figure 3: Accuracy heatmaps over (α, κ) for Sum and Max aggregation on the full pool. Dashed contour: cells within 1 pp of the best score; stars: best per task.

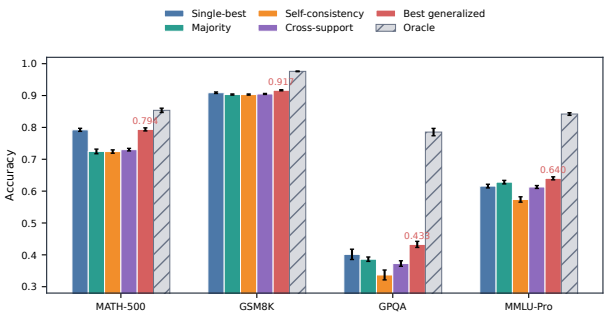

Figure 4: Router generalizations.

Table 3: Optimal-point behavior under the OrchSLM parameter scan. The selected optima emphasize expert trust on the two math benchmarks, while the science benchmarks rely more on mixed or mutual support.

Task	Trust	Consistency	Aggregation	Regime
MATH-500	Expert	Mixed	Tie	Expert
GSM8K	Expert	Self	Max	Saturated
GPQA	Expert	Mixed	Tie	Support
MMLU-Pro	Mixed	Mutual	Tie	Pooled

The best routing strategy varies across tasks. Figure 4 compares four cache-only routing strategies with the oracle upper bound. On MATH-500, single-best routing is near-optimal, indicating that routing is largely determined by the strongest expert. On GSM8K, several strategies perform similarly. In contrast, GPQA and MMLU-Pro retain substantial gaps to the oracle, suggesting that the presence of a correct answer does not make it easy to identify. These differences motivate us to study routing as a task-dependent decision rather than fixing a single strategy across tasks. We vary the three components of the generalized router: the trust concentration κ, the aggregation operator $\mathrm{Agg} \in \{\mathrm{Sum}, \mathrm{Max}\}$, and the balance α between top-answer and pooled support. Figure 3 shows the resulting routing landscape, and Table 3 summarizes the best regime for each task. We next isolate the effect of each component.

Trust concentration. Expert weighting has the largest effect on routing performance. Table 4 compares the best setting with $\kappa > 0$ against the best equal-weight setting with $\kappa = 0$. Concentrating trust consistently improves routing across all four tasks. On math, it prevents weaker models from diluting the strongest expert. On the harder science tasks, it reduces the influence of unreliable support. Thus, deciding which models to trust is the primary routing decision.

Table 4: Experts vs average models. Rel. Δ is the relative change from the best equal-weight cell ($\kappa = 0$) to the best full-sweep cell.

Task	Equal	Expert	Rel. Δ
MATH-500	0.7338	0.7938	+8.2%
GSM8K	0.9052	0.9166	+1.3%
GPQA	0.3909	0.4328	+10.7%
MMLU-Pro	0.6304	0.6402	+1.6%

Aggregation. The choice between Sum and Max has a smaller effect. As shown in Table 5, GSM8K clearly favors Max, while the best Sum and Max settings are close on the other three tasks. This is consistent with GSM8K being a saturated task, where selecting the strongest expert-supported answer is sufficient. In the remaining tasks, changing how support is aggregated matters less than changing which models receive weight.

Agreement. The usefulness of self-consistency and mutual support depends on which models the router trusts. Table 6 compares the two signals by setting $\alpha = 1$ for top-answer support and $\alpha = 0$ for pooled support. The left block optimizes κ and Agg separately at each endpoint; the right block fixes $\kappa = 0$ and optimizes Agg. With equal model weights, the two signals are similar and neither consistently dominates. After expert weighting, their roles separate: math favors self-consistent expert predictions, whereas the broad science task benefits more from pooled support. Agreement is therefore informative only relative to the models providing it.

Table 5: Majority vs coalition support. Rel. Δ is the relative change from Sum to Max aggregation.

Task	Sum	Max	Rel. Δ
MATH-500	0.7938	0.7936	0.0%
GSM8K	0.9089	0.9166	+0.8%
GPQA	0.4318	0.4328	+0.2%
MMLU-Pro	0.6398	0.6402	+0.1%

3.2 Scaling up Routers

Table 6: Self-consistency vs mutual agreement. Pooled-only uses $\alpha = 0$; top-only uses $\alpha = 1$. Rel. Δ is the relative change from pooled-only to top-only routing.

	All κ			$\kappa = 0$		
Task	Pooled	Top	Rel. Δ	Pooled	Top	Rel. Δ
MATH-500	0.7934	0.7936	0.0%	0.7246	0.7242	-0.1%
GSM8K	0.9086	0.9166	+0.9%	0.9032	0.9050	+0.2%
GPQA	0.4318	0.4308	-0.2%	0.3864	0.3879	+0.4%
MMLU-Pro	0.6398	0.6350	-0.8%	0.6282	0.6270	-0.2%

Table 7: Oracle coverage is not routable support.

Benchmark	Oracle	Best generalized	Gap
MATH-500	0.8536	0.7938	0.0598
GSM8K	0.9760	0.9166	0.0594
GPQA	0.7854	0.4328	0.3525
MMLU-Pro	0.8420	0.6402	0.2018

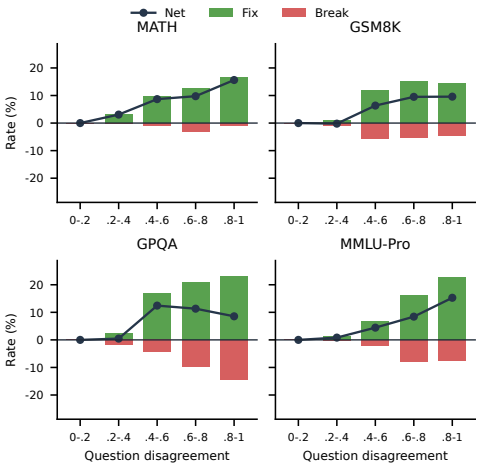

Figure 5: Per-bin fixes (green) and breaks (red) when switching self-consistency to the best generalized cell, with the net rate (line).

Oracle coverage and routability. Higher oracle coverage does not necessarily translate into higher routed accuracy. Table 7 compares the oracle upper bound with the best generalized routing result. The gap is small on the math tasks but substantially larger on GPQA and MMLU-Pro. On these science benchmarks, a correct answer is often present in the model pool but cannot be reliably identified from the cached predictions. Increasing coverage alone is therefore insufficient; the router must also distinguish useful evidence from competing support.

Disagreement. Disagreement creates both opportunities and risks for routing. Figure 5 groups questions by the level of disagreement among the seven models and measures how generalized

routing changes the predictions of self-consistency. The balance between corrected and newly introduced errors is task-dependent: net gains peak at moderate disagreement on GPQA, whereas MATH-500 and MMLU-Pro show the largest gains in the highest-disagreement bin. Disagreement therefore exposes useful alternatives, but its routing value depends on the task.

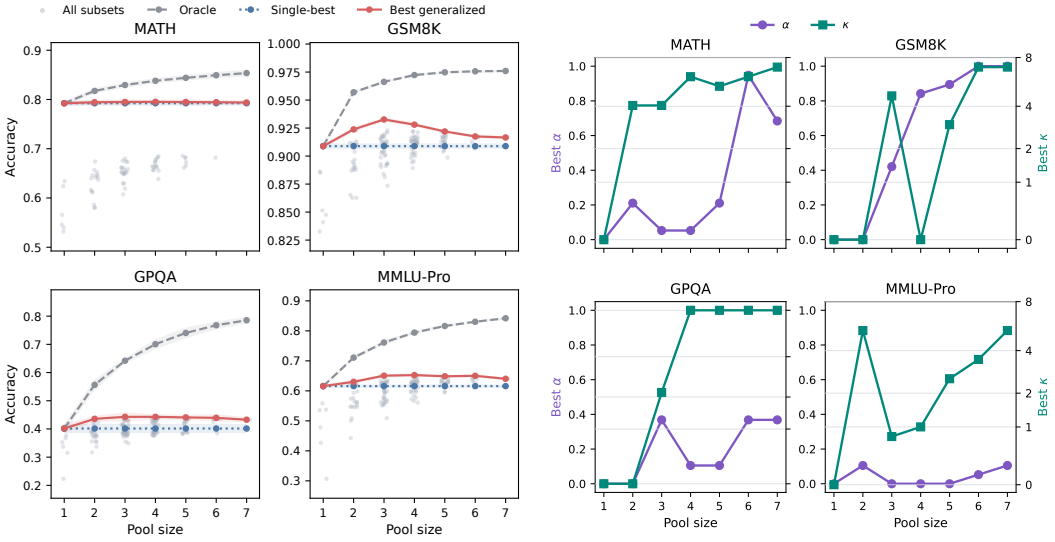

Figure 6: Pool-size scan. Left: routed accuracy vs. size (gray: each subset; lines: oracle, single-best, best generalized). Right: best-cell (α, κ) shifts with pool size.

Pool composition. Adding more models does not necessarily improve routed accuracy. As shown in Figure 6 (left), oracle coverage continues to increase with pool size, while routed accuracy typically peaks with three or four models. Additional models can introduce correct answers that increase oracle coverage, but they can also introduce stable incorrect support that makes those answers harder to identify. Consequently, the quality of the pool depends not only on the answers it contains, but also on the support patterns it creates. Pool composition also changes the preferred routing strategy. Figure 6 (right) shows that the optimal (α, κ) shifts as models are added or removed. Smaller curated pools generally require less trust concentration because the remaining models are already more reliable. Model selection and answer routing are therefore coupled decisions rather than independent stages.

3.3 Being Heard Is Not Being Helpful

Attribution and contribution capture different aspects of model influence. Atr measures how much probability mass a model assigns to the router's selected answer, whereas Ctr measures how routed accuracy changes when that model is removed. A model can therefore receive high attribution without making the router more accurate. Figure 7 shows that the two measures behave differently across tasks. On math, the strongest expert has both high Atr and positive Ctr: the model that most strongly supports the routed answers is also the one that improves them. On GPQA and MMLU-Pro, however, several models receive substantial Atr while having near-zero or negative Ctr. Their predictions frequently agree with the router's decisions, but removing them does not hurt accuracy and can sometimes improve it. Pool curation reduces this mismatch. After removing models with unhelpful support, attribution becomes more concentrated on models with positive counterfactual contribution. This explains why expert weighting and pool selection can improve routing on the harder science tasks: effective orchestration depends not only on how much support a model provides, but also on whether that support is useful for selecting the correct answer. Per-model results are reported in Table 30, with an accepted-support decomposition in Table 31.

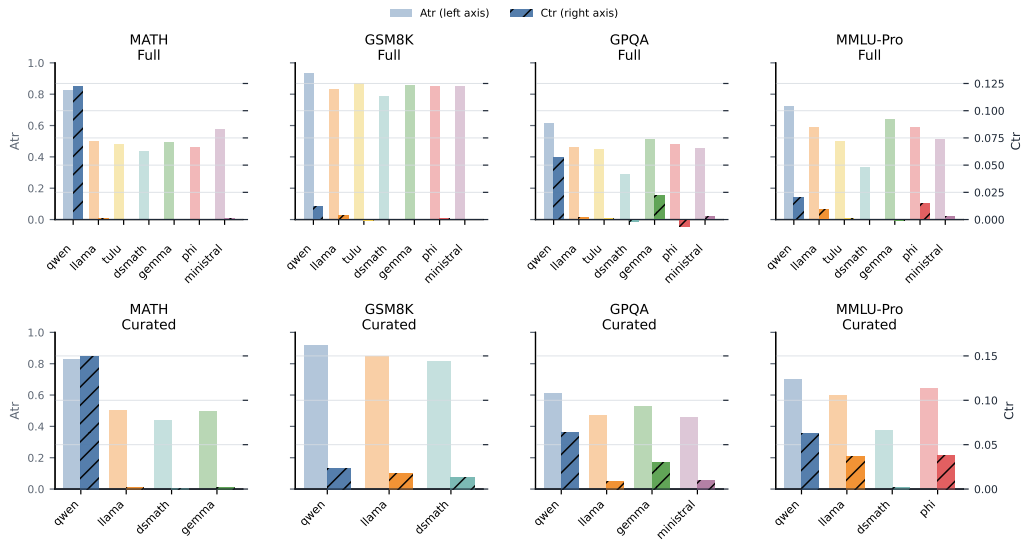

Figure 7: Per-model Atr (heard) vs. Ctr (helpful) under each canonical router.

3.4 In-Domain Task Transfer

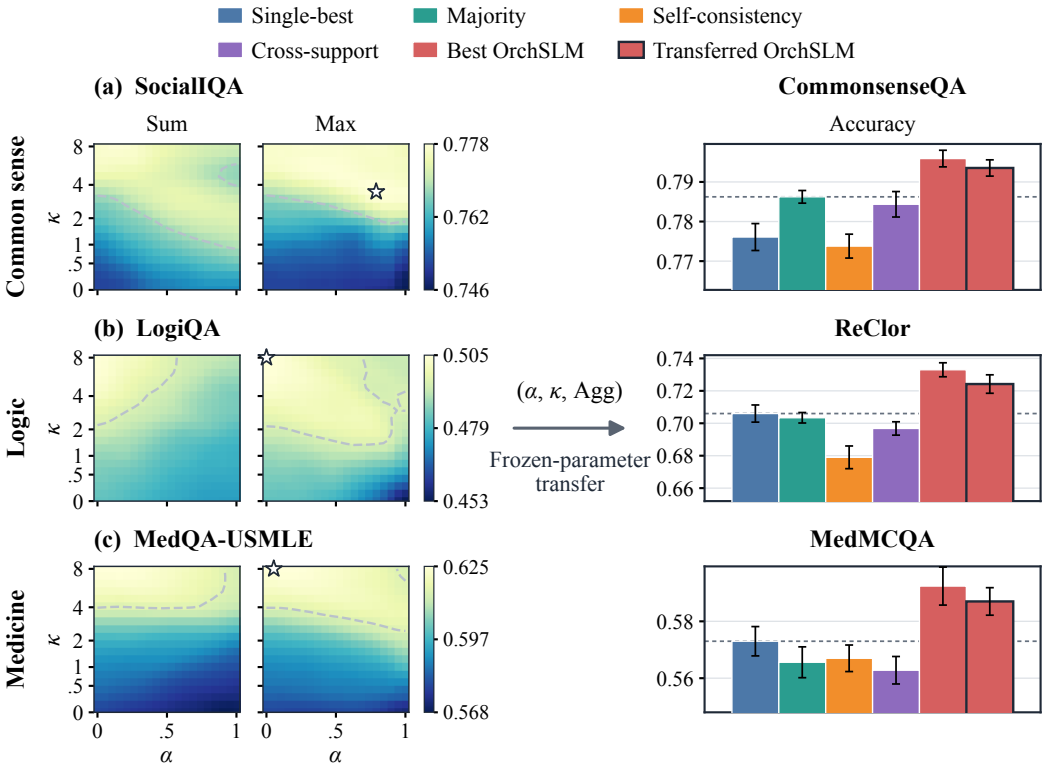

Figure 8: In-domain task transfer. Left: source neighborhood-mean accuracy; stars mark within-pool optima and dashed contours enclose scores within one percentage point of the best. Right: target accuracy using the same pool and target expert prior for all methods. Outlined bars show Transferred OrchSLM; Best OrchSLM is the target grid optimum. Dashed lines mark the strongest fixed baseline.

To demonstrate the broader applicability of our findings, we test in-domain transfer in commonsense, logical reasoning, and medicine. We optimize $(\alpha, \kappa, \text{Agg})$ on one benchmark and evaluate OrchSLM on another in the same domain, keeping the parameters and model pool fixed (Section B.3). Figure 8

shows that transferred OrchSLM outperforms all four fixed baselines from Figure 4 in every case: single-best, majority voting (Wang et al., 2023; Li et al., 2024), SLM-MUX(self-consistency) (Wang et al., 2025), and cross-support. Together with the earlier math and science results, these examples show task-dependent evidence preferences: math and commonsense emphasize top-answer support, whereas science, logical reasoning, and medicine favor pooled support under expert weighting. This deliberately simple experiment supports the cross-task generalizability of domain-specific evidence preferences; it is not intended to propose an optimized, state-of-the-art orchestration method.

4 Related Work

Small Language Models. Small language models (SLMs) trade capacity for lower latency, memory, and deployment cost (Abdin et al., 2024; Liu et al., 2024). Recent gains at this scale come from curated and synthetic data, knowledge distillation, scale-specific architectures, and targeted post-training (Wang et al., 2024a; Abdin et al., 2024; Allal et al., 2025; Gemma Team, 2024; Liu et al., 2024). This efficiency supports repeated or specialized calls on devices and in agentic systems (Abdin et al., 2024; Liu et al., 2024; Belcak et al., 2025; Wang et al., 2025), while chain-of-thought, repeated sampling, and test-time scaling improve reasoning (Wei et al., 2022; Wang et al., 2023; Snell et al., 2024b; Muennighoff et al., 2025). However, weak self-correction and long-context use constrain interaction-heavy orchestration (Huang et al., 2024; Liu et al., 2023). SLM-MUX instead uses independent sampling, confidence-based selection, and model-subset search (Wang et al., 2025); OrchSLM studies routing within this non-interactive setting.

Multi-Agent and Compound AI Systems. Routing and allocation determine which model calls are made, while prompt and workflow search configure their use (Chen et al., 2023; Ong et al., 2025; Chen et al., 2024; 2025; Poon et al., 2025; Khattab et al., 2023; Opsahl-Ong et al., 2024; Saad-Falcon et al., 2025; Zhang et al., 2025a). Interactive systems condition later calls on earlier outputs through debate, critique, synthesis, or verification (Du et al., 2023; Wang et al., 2024b; Lifshitz et al., 2025). MoA variants refine this pattern through proposer quality and diversity, sparse communication, and residual, attention, or memory-based flow (Li et al., 2025b;a; Xie et al., 2025; Wen et al., 2026; Ping et al., 2026). Non-interactive systems instead sample independently and aggregate only after generation (Wang et al., 2025). OrchSLM focuses on this post hoc decision.

Ensembling and Test-Time Inference. Test-time inference improves fixed models or model pools through additional sampling, search, verification, and aggregation (Snell et al., 2024b; Zhang et al., 2025b). In its non-interactive branch, the remaining problem is how to select or combine independently generated candidates. Existing rules range from sample voting and verifier-based selection to confidence-weighted consensus, multi-generator and reward-model ensembles, and reasoning-structure aggregation (Wang et al., 2023; Li et al., 2024; Lightman et al., 2024; Yao et al., 2026; Song et al., 2025; Parulekar et al., 2026; Wang et al., 2025). Attribution methods add a separate question: whether a model merely supports a decision or changes it when removed (Ribeiro et al., 2016; Lundberg & Lee, 2017; Koh & Liang, 2017). OrchSLM parameterizes these choices and separates answer availability from routability and received support from counterfactual contribution.

5 Conclusion and Limitations

We study SLM orchestration along three axes: whom to trust, how to aggregate evidence, and which agreement to use. Effective routing depends on both the task and model pool: expert weighting matters most, larger pools need not help, and agreement is valuable only among useful models. This calls for jointly selecting models and aggregating their evidence.

Our study is limited to cache-only, non-interactive orchestration and benchmarks with extractable answers. Extending the framework to interactive or open-ended settings would require additional generations and semantic equivalence mechanisms. Expert priors also require validation data or an estimate of model quality. Our all-subset analysis explores the pool design space on the evaluated benchmarks.

AI USE STATEMENT

We used large language models to assist with writing clarity, grammar, and experimental code development. All final wording, technical claims, code, results, figures' scientific content, and conclusions were reviewed, validated, and produced by the authors.

REFERENCES

Marah Abdin et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Kenneth J. Arrow. *Social Choice and Individual Values*. John Wiley & Sons, New York, 1951.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small language models are the future of agentic ai. *arXiv preprint arXiv:2506.02153*, 2025.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance, 2023. URL https://arxiv.org/abs/2305.05176.

Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Matei Zaharia, James Zou, and Ion Stoica. Optimizing model selection for compound AI systems. *arXiv preprint arXiv:2502.14815*, 2025.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James T. Kwok, and Yu Zhang. Routerdc: Query-based router by dual contrastive learning for assembling large language models, 2024. URL https://arxiv.org/abs/2409.19886.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate, 2023. URL https://arxiv.org/abs/2305.14325.

Tianyu Fu, Zihan Min, Hanling Zhang, Jichao Yan, Guohao Dai, Wanli Ouyang, and Yu Wang. Cache-to-cache: Direct semantic communication between large language models. *arXiv preprint arXiv:2510.03215*, 2025.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

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.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *Advances in Neural Information Processing Systems*, 35:30016–30030, 2022.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet, 2024. URL https://arxiv.org/abs/2310.01798.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021. doi: 10.3390/app11146421. URL https://www.mdpi.com/2076-3417/11/14/6421.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. Dspy: Compiling declarative language model calls into self-improving pipelines, 2023. URL https://arxiv.org/abs/2310.03714.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pp. 1885–1894, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Dawei Li, Zhen Tan, Peijia Qian, Yifan Li, Kumar Satvik Chaudhary, Lijie Hu, and Jiayi Shen. SMoA: Improving multi-agent large language models with sparse mixture-of-agents. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 54–65, 2025a.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. More agents is all you need, 2024. URL https://arxiv.org/abs/2402.05120.

Wenzhe Li, Yong Lin, Mengzhou Xia, and Chi Jin. Rethinking mixture-of-agents: Is mixing different large language models beneficial? *arXiv preprint arXiv:2502.00674*, 2025b.

Shalev Lifshitz, Sheila A. McIlraith, and Yilun Du. Multi-agent verification: Scaling test-time compute with multiple verifiers, 2025. URL https://arxiv.org/abs/2502.20379.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. LogiQA: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 3622–3628, 2020. doi: 10.24963/ijcai.2020/501. URL https://www.ijcai.org/proceedings/2020/501.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023. URL https://arxiv.org/abs/2307.03172.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. MobileLLM: Optimizing sub-billion parameter language models for on-device use cases. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

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 with preference data, 2025. URL https://arxiv.org/abs/2406.18665.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. Optimizing instructions and demonstrations for multi-stage language model programs, 2024. URL https://arxiv.org/abs/2406.11695.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260. PMLR, 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Amruta Parulekar, Jinu Lee, Dilek Hakkani-Tür, and Hari Sundaram. Reasoning consensus: Structural ensembling of LLM reasoning via weighted DAG aggregation. *arXiv preprint arXiv:2607.27783*, 2026.

Heng Ping, Arijit Bhattacharjee, Peiyu Zhang, Shixuan Li, Wei Yang, Ali Jannesari, Nesreen Ahmed, and Paul Bogdan. Rem-MoA: Reasoning memory sustains mixture-of-agents scaling. *arXiv preprint arXiv:2606.24437*, 2026.

Manhin Poon, Xiangxiang Dai, Xutong Liu, Fang Kong, John C. S. Lui, and Jinhang Zuo. Online multi-LLM selection via contextual bandits under unstructured context evolution. *arXiv preprint arXiv:2506.17670*, 2025.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016. URL https://doi.org/10.1145/2939672.2939778.

Jon Saad-Falcon, Adrian Gamarra Lafuente, Shlok Natarajan, Nahum Maru, Hristo Todorov, Etash Guha, E. Kelly Buchanan, Mayee Chen, Neel Guha, Christopher Ré, and Azalia Mirhoseini. Archon: An architecture search framework for inference-time techniques, 2025. URL https://arxiv.org/abs/2409.15254.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4463–4473. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454/.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024a.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024b. URL https://arxiv.org/abs/2408.03314.

Zhende Song, Shengji Tang, Peng Ye, Jiayuan Fan, and Tao Chen. CTTS: Collective test-time scaling. *arXiv preprint arXiv:2508.03333*, 2025.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge, 2019. URL https://arxiv.org/abs/1811.00937.

Amir Taubenfeld, Yaniv Dover, Roi Reichart, and Ariel Goldstein. Systematic biases in llm simulations of debates. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 251–267. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.16. URL http://dx.doi.org/10.18653/v1/2024.emnlp-main.16.

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.

Chenyu Wang, Zishen Wan, Hao Kang, Emma Chen, Zhiqiang Xie, Tushar Krishna, Vijay Janapa Reddi, and Yilun Du. Slm-mux: Orchestrating small language models for reasoning. *CoRR*, abs/2510.05077, 2025. URL https://arxiv.org/abs/2510.05077.

Fali Wang, Zhiwei Zhang, Xianren Zhang, Zongyu Wu, Tzuhao Mo, Qiuhao Lu, Wanjing Wang, Rui Li, Junjie Xu, Xianfeng Tang, Qi He, Yao Ma, Ming Huang, and Suhang Wang. A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness, 2024a. URL https://arxiv.org/abs/2411.03350.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities, 2024b. URL https://arxiv.org/abs/2406.04692.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2201.11903.

Jianyu Wen, Yang Wei, Xiongxi Yu, Changxuan Xiao, and Ke Zeng. Attention-MoA: Enhancing mixture-of-agents via inter-agent semantic attention and deep residual synthesis. *arXiv preprint arXiv:2601.16596*, 2026.

Zhentao Xie, Chengcheng Han, Jinxin Shi, Wenjun Cui, Xin Zhao, Xingjiao Wu, and Jiabao Zhao. RMoA: Optimizing mixture-of-agents through diversity maximization and residual compensation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6575–6602, 2025. doi: 10.18653/v1/2025.findings-acl.342.

Yu Yao, Jiayi Dong, Yang Yang, Ju Li, and Yilun Du. Roundtable policy: Confidence-weighted-consensus aggregation improves multi-agent-system reasoning, 2026. URL https://arxiv.org/abs/2509.16839.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. ReClor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgJtT4tvB.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. Aflow: Automating agentic workflow generation, 2025a. URL https://arxiv.org/abs/2410.10762.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, Irwin King, Xue Liu, and Chen Ma. A survey on test-time scaling in large language models: What, how, where, and how well?, 2025b. URL https://arxiv.org/abs/2503.24235.

Appendices

Evaluation convention. Unless otherwise stated, results use K20R10: for each question and model, $K = 20$ generations are subsampled from a fixed cache of 50 generations, and evaluation is repeated over $R = 10$ deterministic subsamples. Accuracy is averaged over the ten replicates; reported standard deviations measure cache-subsampling variability. Full-cache $K = 50$ results use all 50 generations once and are labeled separately.

Overview. The first sections provide protocol details, full-pool analyses, pool-composition results, robustness checks, and Atr/Ctr diagnostics for the seven-model experiments. The final section records baseline, fixed-prior sweep, and pool-composition checks under the same K20R10 protocol.

A Mathematical Interpretation

Generalized routing score contains two sources of evidence:

Top-answer support. The first term,

$$\mathrm{Agg}_i \left[M_{ia}\, r_i(\kappa)\, P_{ia}(x) \right], \tag{7}$$

only counts model i's support for answer a if a is that model's own top answer. This term measures how strongly the models that actually select a support it. The operator Agg determines how this support is combined across models:

- Sum adds support from all models whose top answer is a:

$$\mathrm{Sum}_i[z_i] = \sum_i z_i. \tag{8}$$

  This corresponds to coalition-style evidence, where several models can jointly support the same answer.

- Max keeps only the strongest individual supporter:

$$\mathrm{Max}_i[z_i] = \max_i z_i. \tag{9}$$

  This corresponds to strongest-supporter evidence, where one highly confident or highly trusted model can dominate.

Pooled answer support. The second term,

$$\sum_i r_i(\kappa) P_{ia}(x), \tag{10}$$

counts every model's probability mass on answer a, even if a is not that model's top answer. This term measures cross-model or mutual support for a.

The interpolation parameter $\alpha \in [0, 1]$ controls which type of evidence dominates. When $\alpha = 1$, routing relies only on top-answer support. When $\alpha = 0$, routing relies only on pooled support. Thus, κ controls whom to trust, Agg controls whether evidence comes from a coalition or the strongest supporter, and α controls whether confidence comes from top-answer self-consistency or pooled mutual agreement.

B Experimental Details

B.1 Stability and Robustness Check

We examine subsample-replicate variability (Table 22) and sampling-budget sensitivity (Tables 26 and 27). Optimal parameter settings can change with the sampling budget. Prior-transform, tie-breaking, exact cross-support, and fix/break checks are reported in Section B.6.

B.2 Protocol Details and Summary Extraction

The main generalized sweep uses fixed expert priors to analyze routing assumptions. The all-subset pool scans study how pool composition changes routed accuracy. The small main-text summary tables extract views of the same full-pool sweep to answer the three routing questions directly.

Table 8: Task-specific routing regimes from the full-pool generalized sweep. The table summarizes broad preferred regions; near-best plateau statistics are reported separately.

Benchmark	Best acc.	α	κ	Agg.	Interpretation
MATH-500	0.7938	0.6842	7.0171	Sum	single-best-like expert routing
GSM8K	0.9166	1.0000	7.0171	Max	expert self-consistency
GPQA	0.4328	0.3684	8.0000	Max	expert + support-aware routing
MMLU-Pro	0.6402	0.1053	5.3617	Max	expert-weighted pooled support

B.3 In-Domain Transfer Protocol

We evaluate transfer from SocialIQA to CommonsenseQA for commonsense reasoning (Sap et al., 2019; Talmor et al., 2019), from LogiQA to ReClor for logical reasoning (Liu et al., 2020; Yu et al., 2020), and from MedQA-USMLE to MedMCQA for medicine (Jin et al., 2021; Pal et al., 2022). For each question and model, we subsample 20 cached answers and report mean accuracy over ten repeated subsamples.

B.4 FULL-POOL ROUTING DECISIONS

This section contains the full-pool analyses behind the main three-decision analysis. The tables report plateau width, single-best gaps, no-expert-prior ablations, pooled-term and top-term ablations, Sum-vs-Max comparisons, and disagreement-bin behavior. These numbers are the table counterpart of the main heatmap and regime discussion.

Table 9: Full-pool plateau summary. Broad near-best regions support interpreting best cells as regimes with stable neighboring cells.

Benchmark	Best acc.	Best params	Cells @0.25pp	Area @0.25pp	Cells @0.5pp	Area @0.5pp	Cells @1pp	Area @1pp
GPQA	0.4328	Max a=0.3684 k=8.0	19	0.0238	78	0.0975	158	0.1975
GSM8K	0.9166	Max a=1.0 k=7.0171	15	0.0187	39	0.0488	210	0.2625
MATH-500	0.7938	Sum a=0.6842 k=7.0171	137	0.1713	199	0.2487	273	0.3412
MMLU-Pro	0.6402	Max a=0.1053 k=5.3617	38	0.0475	213	0.2662	566	0.7075

Table 10: Expert-prior axis: gap between the best generalized full-pool cell and the single-best limit.

Benchmark	Best gen.	Single-best	Gap	Best κ
MATH-500	0.7938	0.7924	0.0014	7.0171
GSM8K	0.9166	0.9089	0.0077	7.0171
GPQA	0.4328	0.4015	0.0313	8.0000
MMLU-Pro	0.6402	0.6158	0.0244	5.3617

Table 11: No-expert-prior comparison on the full seven-model pool, read from the $\kappa = 0$ best cell in the full-pool phase diagram.

Benchmark	Best κ=0	Std	Agg.	α	Full-family best	Gap
MATH-500	0.7338	0.0055	Max	0.7368	0.7938	-0.0600
GSM8K	0.9052	0.0013	Max	0.8947	0.9166	-0.0114
GPQA	0.3909	0.0120	Sum	0.6316	0.4328	-0.0419
MMLU-Pro	0.6304	0.0069	Sum	0.2632	0.6402	-0.0098

Table 12: Top-answer-term removal ablation on the full seven-model pool. Setting $\alpha = 0$ keeps only pooled answer support.

Benchmark	Best α=0	Agg.	κ	Full-family best	Gap
GPQA	0.4318	Sum	7.0171	0.4328	-0.0010
GSM8K	0.9086	Sum	8.0000	0.9166	-0.0080
MATH-500	0.7934	Sum	8.0000	0.7938	-0.0004
MMLU-Pro	0.6398	Sum	5.3617	0.6402	-0.0004

Table 13: Pooled-term removal ablation on the full seven-model pool. Setting $\alpha = 1$ removes pooled mutual support.

Benchmark	Best α=1	Agg.	κ	Full-family best	Gap
GPQA	0.4308	Sum	6.1416	0.4328	-0.0020
GSM8K	0.9166	Max	7.0171	0.9166	0.0000
MATH-500	0.7936	Sum	6.1416	0.7938	-0.0002
MMLU-Pro	0.6350	Sum	5.3617	0.6402	-0.0052

Table 14: Sum/Max summary. The aggregation switch measures whether support is distributed across models or concentrated in the strongest supporter.

benchmark	benchmark_label	pool_size	mean_sum_minus_max	frac_sum_wins
gpqa	GPQA	1.0000	0.0000	0.0000
gpqa	GPQA	2.0000	0.0000	0.0000
gpqa	GPQA	3.0000	-0.0002	0.2857
gpqa	GPQA	4.0000	0.0004	0.5143
gpqa	GPQA	5.0000	-0.0002	0.4286
gpqa	GPQA	6.0000	-0.0006	0.2857
gpqa	GPQA	7.0000	-0.0010	0.0000
gsm8k	GSM8K	1.0000	0.0000	0.0000
gsm8k	GSM8K	2.0000	0.0000	0.0000
gsm8k	GSM8K	3.0000	-0.0031	0.2857
gsm8k	GSM8K	4.0000	-0.0024	0.2000
gsm8k	GSM8K	5.0000	-0.0053	0.0952
gsm8k	GSM8K	6.0000	-0.0049	0.0000
gsm8k	GSM8K	7.0000	-0.0077	0.0000
math500	MATH-500	1.0000	0.0000	0.0000
math500	MATH-500	2.0000	0.0000	0.0000
math500	MATH-500	3.0000	0.0000	0.1429
math500	MATH-500	4.0000	0.0000	0.2571
math500	MATH-500	5.0000	0.0001	0.3810
math500	MATH-500	6.0000	0.0002	0.5714

Table 15: Full-pool best generalized accuracy by pair-disagreement bin.

Benchmark	Disagreement bin	N	Accuracy
GPQA	(-0.01, 0.2]	80	0.5250
GPQA	(0.2, 0.4]	214	0.4346
GPQA	(0.4, 0.6]	587	0.4889
GPQA	(0.6, 0.8]	795	0.4050
GPQA	(0.8, 1.01]	304	0.3717
GSM8K	(-0.01, 0.2]	8882	0.9895
GSM8K	(0.2, 0.4]	1894	0.9018
GSM8K	(0.4, 0.6]	1513	0.7072
GSM8K	(0.6, 0.8]	515	0.6738
GSM8K	(0.8, 1.01]	386	0.4560
MATH-500	(-0.01, 0.2]	1700	0.9859
MATH-500	(0.2, 0.4]	715	0.9259
MATH-500	(0.4, 0.6]	610	0.8361
MATH-500	(0.6, 0.8]	605	0.7008
MATH-500	(0.8, 1.01]	1370	0.5088
MMLU-Pro	(-0.01, 0.2]	659	0.8998
MMLU-Pro	(0.2, 0.4]	845	0.7988
MMLU-Pro	(0.4, 0.6]	1262	0.7013
MMLU-Pro	(0.6, 0.8]	1056	0.5284
MMLU-Pro	(0.8, 1.01]	1178	0.4160

B.5 Pool Composition and Structure

This section tests whether the model pool itself changes the routing problem. The tables compare pools selected for absolute accuracy with pools selected for gain over the self-consistency router, classify scanned pools by routing regime, and relate structural pool metrics to routing gains.

Table 16: Pool-composition targets. For each benchmark we report the pool with highest absolute accuracy and the pool with largest gain over the self-consistency router.

Benchmark	Target	Pool	Size	Best gen.	Std	Δ vs SLM	Δ vs SB	Params	Regime
GPQA	best acc.	qwen+llama+gemma+ministral	4	0.4429	0.0093	0.0212	0.0414	Sum a=0.1053 k=8.0	high-kappa expert/single-best-like
GPQA	best gain	qwen+tulu+dsmath+phi	4	0.4333	0.0182	0.1101	0.0318	Sum a=1.0 k=3.4968	mixed support-aware
GSM8K	best acc.	qwen+llama+dsmath	3	0.9327	0.0018	0.0060	0.0238	Sum a=0.4211 k=4.6669	high-kappa expert/single-best-like
GSM8K	best gain	qwen+gemma+ministral	3	0.9111	0.0021	0.0227	0.0022	Max a=0.9474 k=8.0	no stable gain
MATH-500	best acc.	qwen+llama+dsmath+gemma	4	0.7952	0.0053	0.0558	0.0028	Max a=0.0526 k=6.1416	no stable gain
MATH-500	best gain	qwen+llama+tulu+dsmath+gemma+phi+ministral	7	0.7938	0.0049	0.0696	0.0014	Sum a=0.6842 k=7.0171	no stable gain
MMLU-Pro	best acc.	qwen+llama+dsmath+phi	4	0.6526	0.0052	0.0652	0.0368	Sum a=0.0 k=1.0014	low/moderate-alpha Sum coalition/Majority-like
MMLU-Pro	best gain	llama+dsmath	2	0.5646	0.0078	0.0936	0.0060	Sum a=0.0 k=1.5222	low/moderate-alpha Sum coalition/Majority-like

Table 17: Best absolute pool from the all-subset pool-composition scan. The companion figure shows how accuracy changes with pool size, while this table records the selected model names.

Benchmark	Best absolute pool	Size	Best acc.	Full-pool best
MATH-500	qwen+llama+dsmath+gemma	4	0.7952	0.7938
GSM8K	qwen+llama+dsmath	3	0.9327	0.9166
GPQA	qwen+llama+gemma+ministral	4	0.4429	0.4328
MMLU-Pro	qwen+llama+dsmath+phi	4	0.6526	0.6402

Table 18: Regime classification over all scanned pools. Fractions are within benchmark.

Benchmark	Regime	Pools	Frac.
GPQA	high-alpha Max self-consistency-like	8	0.063
GPQA	high-kappa expert/single-best-like	72	0.567
GPQA	low/moderate-alpha Sum coalition/Majority-like	12	0.094
GPQA	mixed support-aware	14	0.110
GPQA	no stable gain	21	0.165
GSM8K	high-alpha Max self-consistency-like	3	0.024
GSM8K	high-kappa expert/single-best-like	74	0.583
GSM8K	low/moderate-alpha Sum coalition/Majority-like	12	0.094
GSM8K	mixed support-aware	4	0.031
GSM8K	no stable gain	34	0.268
MATH-500	high-kappa expert/single-best-like	47	0.370
MATH-500	low/moderate-alpha Sum coalition/Majority-like	10	0.079
MATH-500	no stable gain	70	0.551
MMLU-Pro	high-kappa expert/single-best-like	64	0.504
MMLU-Pro	low/moderate-alpha Sum coalition/Majority-like	44	0.346
MMLU-Pro	mixed support-aware	8	0.063
MMLU-Pro	no stable gain	11	0.087

Table 19: Structure-predictor correlations between pool metrics and generalized-routing gains.

benchmark_label	predictor	response	pearson_r
GPQA	pool_size	delta_best_vs_slm	0.6762
GPQA	pool_size	delta_best_vs_single_best	0.5491
GPQA	pool_size	delta_best_vs_majority	0.3595
GPQA	pool_size	best_gen_acc_mean	0.5517
GPQA	oracle_gap_mean	delta_best_vs_slm	0.6899
GPQA	oracle_gap_mean	delta_best_vs_single_best	0.5402
GPQA	oracle_gap_mean	delta_best_vs_majority	0.3930
GPQA	oracle_gap_mean	best_gen_acc_mean	0.5632
GPQA	wrong_answer_agreement_mean	delta_best_vs_slm	0.6572
GPQA	wrong_answer_agreement_mean	delta_best_vs_single_best	0.5492
GPQA	wrong_answer_agreement_mean	delta_best_vs_majority	0.3542
GPQA	wrong_answer_agreement_mean	best_gen_acc_mean	0.5714
GPQA	avg_pair_disagreement_mean	delta_best_vs_slm	0.5734
GPQA	avg_pair_disagreement_mean	delta_best_vs_single_best	0.2224
GPQA	avg_pair_disagreement_mean	delta_best_vs_majority	0.5663
GPQA	avg_pair_disagreement_mean	best_gen_acc_mean	0.3251
GPQA	stable_wrong_rate_mean	delta_best_vs_slm	0.8332
GPQA	stable_wrong_rate_mean	delta_best_vs_single_best	0.4150
GPQA	stable_wrong_rate_mean	delta_best_vs_majority	0.5489
GPQA	stable_wrong_rate_mean	best_gen_acc_mean	0.4800
GPQA	single_model_acc_mean	delta_best_vs_slm	-0.3758
GPQA	single_model_acc_mean	delta_best_vs_single_best	0.4190
GPQA	single_model_acc_mean	delta_best_vs_majority	-0.4097
GPQA	single_model_acc_mean	best_gen_acc_mean	0.6045
GPQA	single_model_acc_std	delta_best_vs_slm	0.7475
GPQA	single_model_acc_std	delta_best_vs_single_best	0.0559
GPQA	single_model_acc_std	delta_best_vs_majority	0.7916
GPQA	single_model_acc_std	best_gen_acc_mean	0.2273
GPQA	single_model_acc_spread	delta_best_vs_slm	0.8318
GPQA	single_model_acc_spread	delta_best_vs_single_best	0.1763
GPQA	single_model_acc_spread	delta_best_vs_majority	0.7839
GPQA	single_model_acc_spread	best_gen_acc_mean	0.3234
GSM8K	pool_size	delta_best_vs_slm	0.4439
GSM8K	pool_size	delta_best_vs_single_best	0.0014
GSM8K	pool_size	delta_best_vs_majority	0.3702
GSM8K	pool_size	best_gen_acc_mean	0.5492
GSM8K	oracle_gap_mean	delta_best_vs_slm	0.1643
GSM8K	oracle_gap_mean	delta_best_vs_single_best	0.7245
GSM8K	oracle_gap_mean	delta_best_vs_majority	-0.0527
GSM8K	oracle_gap_mean	best_gen_acc_mean	0.2886

B.6 FIX/BREAK, ROBUSTNESS, AND ABLATIONS

This section stress-tests the main conclusions. The fix/break table separates questions corrected by generalized routing from questions newly broken by it, including comparisons against several canonical reference routers. Corrected significance tests check whether selected-pool gains survive multiple-baseline correction. Replicate-stability tables summarize K20R10 variability. The remaining tables vary expert priors, tie-breaking and invalid-answer handling, exact cross-support bridge strength, K=50 evaluation, effective sample budget, and method-correlation structure.

Table 20: Fix/break summary for full and best-absolute pools. Counts are over all K20R10 question-replicate rows.

Benchmark	Pool	Comparison	N	Fix	Break	Target acc.	Base acc.	Net
GPQA	full	vs slm	1980	341	151	0.4328	0.3369	0.0960
GPQA	full	vs single	1980	155	93	0.4328	0.4015	0.0313
GPQA	full	vs Majority	1980	199	107	0.4328	0.3864	0.0465
GPQA	best_abs	vs slm	1980	139	97	0.4429	0.4217	0.0212
GPQA	best_abs	vs single	1980	163	81	0.4429	0.4015	0.0414
GPQA	best_abs	vs Majority	1980	108	59	0.4429	0.4182	0.0247
GSM8K	full	vs slm	13190	328	150	0.9166	0.9031	0.0135
GSM8K	full	vs single	13190	230	128	0.9166	0.9089	0.0077
GSM8K	full	vs Majority	13190	298	121	0.9166	0.9032	0.0134
GSM8K	best_abs	vs slm	13190	150	71	0.9327	0.9267	0.0060
GSM8K	best_abs	vs single	13190	447	133	0.9327	0.9089	0.0238
GSM8K	best_abs	vs Majority	13190	58	56	0.9327	0.9325	0.0002
MATH-500	full	vs slm	5000	384	36	0.7938	0.7242	0.0696
MATH-500	full	vs single	5000	9	2	0.7938	0.7924	0.0014
MATH-500	full	vs Majority	5000	425	79	0.7938	0.7246	0.0692
MMLU-Pro	full	vs slm	5000	532	200	0.6402	0.5738	0.0664
MMLU-Pro	full	vs single	5000	280	158	0.6402	0.6158	0.0244
MMLU-Pro	full	vs Majority	5000	217	157	0.6402	0.6282	0.0120

Table 21: Selected-pool paired tests with max-stat correction across baselines within each benchmark/pool role.

Benchmark	Pool role	Baseline	Δ	Corrected p
GPQA	best_abs	SLM-MUX_best_abs	0.0212	0.009
GPQA	best_abs	single-best_best_abs	0.0414	0.000
GPQA	best_abs	Majority_best_abs	0.0247	0.001
GPQA	best_abs	cross slice_best_abs	0.0268	0.000
GPQA	best_gap	SLM-MUX_best_gap	0.1101	0.000
GPQA	best_gap	single-best_best_gap	0.0318	0.003
GPQA	best_gap	Majority_best_gap	0.0753	0.000
GPQA	best_gap	cross slice_best_gap	0.0889	0.000
GSM8K	best_abs	SLM-MUX_best_abs	0.0060	0.001
GSM8K	best_abs	single-best_best_abs	0.0238	0.000
GSM8K	best_abs	Majority_best_abs	0.0002	0.826
GSM8K	best_abs	cross slice_best_abs	0.0003	0.765

Table 22: Replicate stability. Means and standard deviations are over the 10 K20R10 subsampling replicates.

Benchmark	Pool role	Method	Mean acc.	Std
GPQA	best_abs	Majority	0.4182	0.0144
GPQA	best_abs	Single-Best	0.4015	0.0170
GPQA	best_abs	SLM-MUX	0.4217	0.0122
GPQA	best_gap	Majority	0.3581	0.0155
GPQA	best_gap	Single-Best	0.4015	0.0170
GPQA	best_gap	SLM-MUX	0.3232	0.0203
GPQA	full	Majority	0.3864	0.0072
GPQA	full	Single-Best	0.4015	0.0170
GPQA	full	SLM-MUX	0.3369	0.0163
GSM8K	best_abs	Majority	0.9325	0.0009
GSM8K	best_abs	Single-Best	0.9089	0.0025
GSM8K	best_abs	SLM-MUX	0.9267	0.0019
GSM8K	full	Majority	0.9032	0.0019
GSM8K	full	Single-Best	0.9089	0.0025
GSM8K	full	SLM-MUX	0.9031	0.0019
MATH-500	full	Majority	0.7246	0.0077
MATH-500	full	Single-Best	0.7924	0.0051
MATH-500	full	SLM-MUX	0.7242	0.0056
MMLU-Pro	full	Majority	0.6282	0.0058
MMLU-Pro	full	Single-Best	0.6158	0.0061
MMLU-Pro	full	SLM-MUX	0.5738	0.0089

Table 23: Expert-prior transform variants. These checks test whether the qualitative regimes depend on a single prior transformation.

benchmark	benchmark_label	pool_role	pool_id	prior_variant	best_acc	best_agg	best_kappa	best_alpha
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	softmax_log_accuracy	0.7938	Sum	7.0171	0.6842
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	power_accuracy	0.7938	Sum	7.0171	0.6842
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	centered_logit	0.7938	Sum	2.1786	0.7368
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	rank_based	0.7936	Sum	5.3617	0.9474
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	uniform	0.7338	Max	0.0000	0.7368
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	softmax_log_accuracy	0.9166	Max	7.0171	1.0000
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	power_accuracy	0.9166	Max	7.0171	1.0000
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	centered_logit	0.9177	Sum	4.0481	0.0000
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	rank_based	0.9187	Max	4.0481	0.0526
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	uniform	0.9052	Max	0.0000	0.8947
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	softmax_log_accuracy	0.9327	Sum	4.6669	0.4211
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	power_accuracy	0.9327	Sum	4.6669	0.4211
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	centered_logit	0.9326	Sum	0.5882	0.4737
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	rank_based	0.9326	Sum	0.2602	0.5789
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	uniform	0.9325	Sum	0.0000	0.0000
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	softmax_log_accuracy	0.4328	Max	8.0000	0.3684
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	power_accuracy	0.4328	Max	8.0000	0.3684
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	centered_logit	0.4348	Max	8.0000	0.0526
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	rank_based	0.4379	Sum	3.4968	0.0000
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	uniform	0.3909	Sum	0.0000	0.6316
gpqa	GPQA	best_abs	qwen+llama+gemma+ministral	softmax_log_accuracy	0.4429	Sum	8.0000	0.1053
gpqa	GPQA	best_abs	qwen+llama+gemma+ministral	power_accuracy	0.4429	Sum	8.0000	0.1053
gpqa	GPQA	best_abs	qwen+llama+gemma+ministral	centered_logit	0.4424	Sum	4.6669	0.1053
gpqa	GPQA	best_abs	qwen+llama+gemma+ministral	rank_based	0.4419	Max	1.0014	0.2632

Table 24: Tie-break sensitivity. Tie-breaking changes small details while preserving the main regime interpretation.

benchmark	benchmark_label	pool_role	pool_id	tie_break	accuracy_mean	accuracy_std
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	pooled_lex	0.7938	0.0049
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	lex	0.7938	0.0049
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	fixed_model_order	0.7938	0.0049
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	random	0.7938	0.0049
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	pooled_lex	0.9166	0.0017
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	lex	0.9166	0.0017
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	fixed_model_order	0.9166	0.0017
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	random	0.9166	0.0017
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	pooled_lex	0.9327	0.0018
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	lex	0.9327	0.0018
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	fixed_model_order	0.9327	0.0018
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	random	0.9327	0.0018
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	pooled_lex	0.4328	0.0096
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	lex	0.4328	0.0096
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	fixed_model_order	0.4328	0.0096
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	random	0.4328	0.0096
gpqa	GPQA	best_abs	qwen+llama+gemma+ministral	pooled_lex	0.4429	0.0093
gpqa	GPQA	best_abs	qwen+llama+gemma+ministral	lex	0.4429	0.0093
gpqa	GPQA	best_abs	qwen+llama+gemma+ministral	fixed_model_order	0.4429	0.0093
gpqa	GPQA	best_abs	qwen+llama+gemma+ministral	random	0.4429	0.0093
gpqa	GPQA	best_gap	qwen+tulu+dsmath+phi	pooled_lex	0.4333	0.0182
gpqa	GPQA	best_gap	qwen+tulu+dsmath+phi	lex	0.4333	0.0182
gpqa	GPQA	best_gap	qwen+tulu+dsmath+phi	fixed_model_order	0.4333	0.0182
gpqa	GPQA	best_gap	qwen+tulu+dsmath+phi	random	0.4333	0.0182

Table 25: Exact cross-support bridge sweep. This sweep studies how the bridge varies with cross-support strength.

benchmark	benchmark_label	pool_role	pool_id	lambda	accuracy_mean	accuracy_std
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	0.0	0.7296	0.0043
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	0.16666666666666666	0.7304	0.0042
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	0.25	0.7334	0.0059
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	0.5	0.7280	0.0044
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	1.0	0.7236	0.0071
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	2.0	0.7202	0.0070
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	0.0	0.9096	0.0016
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	0.16666666666666666	0.9050	0.0014
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	0.25	0.9045	0.0017
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	0.5	0.9034	0.0019
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	1.0	0.9034	0.0017
gsm8k	GSM8K	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	2.0	0.9029	0.0016
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	0.0	0.9203	0.0022
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	0.16666666666666666	0.9292	0.0021
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	0.25	0.9297	0.0023
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	0.5	0.9308	0.0016
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	1.0	0.9308	0.0010
gsm8k	GSM8K	best_abs	qwen+llama+dsmath	2.0	0.9300	0.0017
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	0.0	0.3172	0.0084
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	0.16666666666666666	0.3737	0.0098
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	0.25	0.3808	0.0072
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	0.5	0.3864	0.0126
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	1.0	0.3864	0.0076
gpqa	GPQA	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	2.0	0.3894	0.0097

Table 26: K=50 distribution-level phase diagram. K=50 uses all cached samples once and serves as a robustness check for the main protocol.

Benchmark	Best acc.	Agg.	α	κ	Majority	SC	Single-best
MATH-500	0.8000	Sum	0.6316	7.0171	0.7200	0.7240	0.7980
GSM8K	0.9189	Max	1.0000	8.0000	0.9045	0.9052	0.9105
GPQA	0.4394	Max	0.6842	4.0481	0.3737	0.3030	0.4091
MMLU-Pro	0.6480	Sum	0.8947	0.0000	0.6280	0.5580	0.6180

Table 27: Effective sample-budget scaling under K20R-style subsampling.

benchmark	benchmark_label	pool_role	pool_id	K_eff	R	method	accuracy_mean	accuracy_std	accuracy_min	accuracy_max
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	1.0000	50	majority	0.6561	0.0105	0.6340	0.6780
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	1.0000	50	slm_mux	0.6561	0.0105	0.6340	0.6780
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	1.0000	50	cc_like	0.6561	0.0105	0.6340	0.6780
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	1.0000	50	single_best	0.7312	0.0111	0.7020	0.7540
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	1.0000	50	best_gen_step3_cell	0.7315	0.0111	0.7020	0.7540
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	1.0000	50	best_sum_step3_cell	0.7315	0.0111	0.7020	0.7540
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	1.0000	50	best_max_step3_cell	0.7315	0.0111	0.7020	0.7540
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	2.0000	50	majority	0.6915	0.0103	0.6700	0.7180
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	2.0000	50	slm_mux	0.6886	0.0102	0.6600	0.7100
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	2.0000	50	cc_like	0.6886	0.0102	0.6600	0.7100
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	2.0000	50	single_best	0.7354	0.0114	0.7160	0.7660
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	2.0000	50	best_gen_step3_cell	0.7355	0.0113	0.7160	0.7660
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	2.0000	50	best_sum_step3_cell	0.7355	0.0113	0.7160	0.7660
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	2.0000	50	best_max_step3_cell	0.7487	0.0091	0.7280	0.7700
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	3.0000	50	majority	0.7026	0.0082	0.6860	0.7260
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	3.0000	50	slm_mux	0.6984	0.0097	0.6680	0.7200
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	3.0000	50	cc_like	0.6985	0.0099	0.6680	0.7200
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	3.0000	50	single_best	0.7580	0.0083	0.7420	0.7760
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	3.0000	50	best_gen_step3_cell	0.7584	0.0085	0.7440	0.7760
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	3.0000	50	best_sum_step3_cell	0.7584	0.0085	0.7440	0.7760
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	3.0000	50	best_max_step3_cell	0.7616	0.0085	0.7440	0.7800
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	5.0000	50	majority	0.7147	0.0070	0.6980	0.7320
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	5.0000	50	slm_mux	0.7098	0.0094	0.6940	0.7300
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	5.0000	50	cc_like	0.7140	0.0080	0.6980	0.7300
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	5.0000	50	single_best	0.7715	0.0068	0.7580	0.7840
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	5.0000	50	best_gen_step3_cell	0.7718	0.0066	0.7580	0.7880
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	5.0000	50	best_sum_step3_cell	0.7718	0.0066	0.7580	0.7880
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	5.0000	50	best_max_step3_cell	0.7741	0.0069	0.7600	0.7920
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	10	50	majority	0.7204	0.0068	0.7000	0.7340
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	10	50	slm_mux	0.7213	0.0086	0.7040	0.7380
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	10	50	cc_like	0.7241	0.0083	0.7080	0.7440
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	10	50	single_best	0.7826	0.0067	0.7660	0.7940
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	10	50	best_gen_step3_cell	0.7837	0.0065	0.7680	0.7940
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	10	50	best_sum_step3_cell	0.7837	0.0065	0.7680	0.7940
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	10	50	best_max_step3_cell	0.7854	0.0071	0.7640	0.7980
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	20	20	majority	0.7254	0.0062	0.7140	0.7340
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	20	20	slm_mux	0.7271	0.0094	0.7100	0.7520
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	20	20	cc_like	0.7302	0.0063	0.7200	0.7420
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	20	20	single_best	0.7920	0.0041	0.7840	0.8000
math500	MATH-500	full	qwen+llama+tulu+dsmath+gemma+phi+ministral	20	20	best_gen_step3_cell	0.7932	0.0046	0.7840	0.8020

Table 28: Method-correlation analysis. High correlation indicates that two methods draw on overlapping evidence.

benchmark	method_a	method_b	prediction_agreement	correctness_corr
gpqa	single_best_full	majority_full	0.5955	0.4504
gpqa	single_best_full	slm_mux_full	0.5247	0.3339
gpqa	single_best_full	cc_like_full	0.5732	0.3978
gpqa	single_best_full	best_gen_full	0.7813	0.7442
gpqa	single_best_full	oracle_full	0.4015	0.4282
gpqa	majority_full	slm_mux_full	0.7869	0.7271
gpqa	majority_full	cc_like_full	0.8929	0.8685
gpqa	majority_full	best_gen_full	0.7742	0.6843
gpqa	majority_full	oracle_full	0.3864	0.4148
gpqa	slm_mux_full	cc_like_full	0.8914	0.8561
gpqa	slm_mux_full	best_gen_full	0.6354	0.4902
gpqa	slm_mux_full	oracle_full	0.3369	0.3726
gpqa	cc_like_full	best_gen_full	0.7101	0.5915
gpqa	cc_like_full	oracle_full	0.3727	0.4030
gpqa	best_gen_full	oracle_full	0.4328	0.4567
gsm8k	single_best_full	majority_full	0.9403	0.7302
gsm8k	single_best_full	slm_mux_full	0.9330	0.6933
gsm8k	single_best_full	cc_like_full	0.9382	0.7159
gsm8k	single_best_full	best_gen_full	0.9632	0.8306
gsm8k	single_best_full	oracle_full	0.9089	0.4956
gsm8k	majority_full	slm_mux_full	0.9788	0.9276
gsm8k	majority_full	cc_like_full	0.9890	0.9616
gsm8k	majority_full	best_gen_full	0.9591	0.8091
gsm8k	majority_full	oracle_full	0.9032	0.4793
gsm8k	slm_mux_full	cc_like_full	0.9896	0.9647
gsm8k	slm_mux_full	best_gen_full	0.9547	0.7818
gsm8k	slm_mux_full	oracle_full	0.9031	0.4791
gsm8k	cc_like_full	best_gen_full	0.9597	0.8057
gsm8k	cc_like_full	oracle_full	0.9050	0.4844
gsm8k	best_gen_full	oracle_full	0.9166	0.5202
math500	single_best_full	majority_full	0.8084	0.7331
math500	single_best_full	slm_mux_full	0.8146	0.7798
math500	single_best_full	cc_like_full	0.8228	0.7787
math500	single_best_full	best_gen_full	0.9974	0.9933
math500	single_best_full	oracle_full	0.7924	0.8091
math500	majority_full	slm_mux_full	0.8698	0.8347
math500	majority_full	cc_like_full	0.9134	0.9012
math500	majority_full	best_gen_full	0.8110	0.7393
math500	majority_full	oracle_full	0.7234	0.6642
math500	slm_mux_full	cc_like_full	0.9548	0.9316

B.7 ATR/CTR MODEL-SUPPORT ANALYSIS

This section asks which models the router listens to and whether that support is helpful. The Atr/Ctr table reports the leave-one-out counterfactual Ctr used in the main text. The Llama table separates pools that include Llama from comparable Llama-free pools. The accepted-support decomposition separates support on correct routed answers from support on wrong routed answers.

Table 29: Llama inclusion analysis comparing the best pool containing Llama with the best pool excluding Llama.

Benchmark	Comparison	Pool	Size	Best gen.	Δ vs SLM
GPQA	contains_llama	qwen+llama+gemma+ministral	4	0.4429	0.0212
GPQA	no_llama	qwen+gemma+phi	3	0.4429	0.0152
GSM8K	contains_llama	qwen+llama+dsmath	3	0.9327	0.0060
GSM8K	no_llama	qwen+dsmath	2	0.9231	0.0016
MATH-500	contains_llama	qwen+llama+dsmath+gemma	4	0.7952	0.0558
MATH-500	no_llama	qwen+dsmath+gemma	3	0.7944	0.0546
MMLU-Pro	contains_llama	qwen+llama+dsmath+phi	4	0.6526	0.0652
MMLU-Pro	no_llama	qwen+dsmath+gemma+phi	4	0.6384	0.0594

Table 30: Per-model Atr/Ctr diagnostics for the full pool and the selected curated pool in each benchmark. Atr is support assigned to the routed answer; Ctr is the leave-one-out counterfactual contribution defined in §2.

Benchmark	Setting	Model	Atr	Ctr
MATH-500	Full	qwen	0.8283	0.1220
MATH-500	Full	llama	0.5022	0.0014
MATH-500	Full	tulu	0.4817	0.0002
MATH-500	Full	dsmath	0.4392	0.0002
MATH-500	Full	gemma	0.4951	0.0002
MATH-500	Full	phi	0.4637	-0.0004
MATH-500	Full	ministral	0.5759	0.0012
MATH-500	Curated	qwen	0.8279	0.1496
MATH-500	Curated	llama	0.5047	0.0016
MATH-500	Curated	dsmath	0.4404	0.0004
MATH-500	Curated	gemma	0.4974	0.0012
GSM8K	Full	qwen	0.9320	0.0121
GSM8K	Full	llama	0.8340	0.0037
GSM8K	Full	tulu	0.8722	-0.0009
GSM8K	Full	dsmath	0.7876	-0.0004
GSM8K	Full	gemma	0.8577	0.0003
GSM8K	Full	phi	0.8494	0.0010
GSM8K	Full	ministral	0.8542	-0.0001
GSM8K	Curated	qwen	0.9179	0.0228
GSM8K	Curated	llama	0.8482	0.0177
GSM8K	Curated	dsmath	0.8121	0.0135
GPQA	Full	qwen	0.6139	0.0571
GPQA	Full	llama	0.4628	0.0020
GPQA	Full	tulu	0.4477	0.0010
GPQA	Full	dsmath	0.2933	-0.0015
GPQA	Full	gemma	0.5169	0.0222
GPQA	Full	phi	0.4828	-0.0066
GPQA	Full	ministral	0.4574	0.0025
GPQA	Curated	qwen	0.6093	0.0641
GPQA	Curated	llama	0.4682	0.0081
GPQA	Curated	gemma	0.5304	0.0303
GPQA	Curated	ministral	0.4578	0.0101
MMLU-Pro	Full	qwen	0.7273	0.0204
MMLU-Pro	Full	llama	0.5898	0.0094
MMLU-Pro	Full	tulu	0.5038	0.0010
MMLU-Pro	Full	dsmath	0.3341	0.0004
MMLU-Pro	Full	gemma	0.6430	-0.0008
MMLU-Pro	Full	phi	0.5892	0.0146
MMLU-Pro	Full	ministral	0.5156	0.0024
MMLU-Pro	Curated	qwen	0.6974	0.0624
MMLU-Pro	Curated	llama	0.5968	0.0364
MMLU-Pro	Curated	dsmath	0.3731	0.0022
MMLU-Pro	Curated	phi	0.6400	0.0382

Table 31: Accepted-support decomposition. For each benchmark and pool role, we show the three models with highest Atr under best generalized routing. Balance is correct accepted support minus wrong accepted support.

Benchmark	Pool	Model	Atr	Correct part	Wrong part	Balance	Precision
GPQA	best_abs	qwen	0.6093	0.2862	0.3231	-0.0368	0.4698
GPQA	best_abs	gemma	0.5304	0.2480	0.2824	-0.0343	0.4676
GPQA	best_abs	llama	0.4682	0.2105	0.2578	-0.0473	0.4496
GPQA	full	qwen	0.6139	0.2814	0.3326	-0.0512	0.4583
GPQA	full	gemma	0.5169	0.2419	0.2750	-0.0332	0.4679
GPQA	full	phi	0.4828	0.2148	0.2680	-0.0532	0.4449
GSM8K	best_abs	qwen	0.9179	0.8766	0.0413	0.8354	0.9550
GSM8K	best_abs	llama	0.8482	0.8149	0.0332	0.7817	0.9608
GSM8K	best_abs	dsmath	0.8121	0.7829	0.0292	0.7537	0.9640
GSM8K	full	qwen	0.9320	0.8757	0.0563	0.8194	0.9396
GSM8K	full	tulu	0.8722	0.8269	0.0453	0.7816	0.9481
GSM8K	full	gemma	0.8577	0.8069	0.0507	0.7562	0.9408
MATH-500	full	qwen	0.8283	0.7189	0.1094	0.6094	0.8679
MATH-500	full	ministral	0.5759	0.5335	0.0425	0.4910	0.9262
MATH-500	full	llama	0.5022	0.4697	0.0325	0.4372	0.9353
MMLU-Pro	full	qwen	0.7273	0.5271	0.2002	0.3269	0.7247
MMLU-Pro	full	gemma	0.6430	0.4635	0.1795	0.2840	0.7209
MMLU-Pro	full	llama	0.5898	0.4385	0.1513	0.2871	0.7435

C SEVEN-MODEL PROTOCOL CHECKS

These tables summarize the seven-model fixed-prior sweep, including baseline accuracies, and the highest-accuracy model subsets under the K20R10 protocol.

Table 32: Seven-model K20R10 fixed-prior sweep summary.

Benchmark	Majority	SC	Cross-support	Single-best	Best gen.	Best α	Best κ	Agg.
MATH-500	0.7246	0.7242	0.7302	0.7924	0.7938	0.6842	7.0171	Sum
GSM8K	0.9032	0.9031	0.9050	0.9089	0.9166	1.0000	7.0171	Max
GPQA	0.3864	0.3369	0.3727	0.4015	0.4328	0.3684	8.0000	Max
MMLU-Pro	0.6282	0.5738	0.6130	0.6158	0.6402	0.1053	5.3617	Max

Table 33: Seven-model top pool-composition results, with five best pools per benchmark by generalized accuracy.

Benchmark	Pool	Size	Best gen.	Single-best	SC	Oracle	Agg.
MATH-500	qwen+llama+tulu+gemma	4	0.7952	0.7924	0.7452	0.8366	Max
MATH-500	qwen+llama+dsmath+gemma	4	0.7952	0.7924	0.7394	0.8380	Max
MATH-500	qwen+llama+gemma	3	0.7950	0.7924	0.7504	0.8294	Sum
MATH-500	qwen+llama+tulu+dsmath+gemma	5	0.7948	0.7924	0.7360	0.8434	Sum
MATH-500	qwen+llama	2	0.7944	0.7924	0.7860	0.8174	Sum
GSM8K	qwen+llama+dsmath	3	0.9327	0.9089	0.9267	0.9660	Sum
GSM8K	qwen+llama+tulu+dsmath	4	0.9281	0.9089	0.9252	0.9678	Max
GSM8K	qwen+llama	2	0.9238	0.9089	0.9213	0.9479	Sum
GSM8K	qwen+dsmath	2	0.9231	0.9089	0.9215	0.9569	Sum
GSM8K	qwen+tulu+dsmath	3	0.9230	0.9089	0.9227	0.9635	Max
GPQA	qwen+gemma+phi	3	0.4429	0.4015	0.4278	0.6348	Max
GPQA	qwen+llama+gemma+ministral	4	0.4429	0.4015	0.4217	0.6788	Sum
GPQA	qwen+llama+gemma	3	0.4424	0.4015	0.4247	0.6343	Sum
GPQA	qwen+llama+gemma+phi	4	0.4409	0.4015	0.4162	0.6904	Max
GPQA	qwen+dsmath+gemma+phi	4	0.4409	0.4015	0.3591	0.6970	Max
MMLU-Pro	qwen+llama+dsmath+phi	4	0.6526	0.6158	0.5874	0.7846	Sum
MMLU-Pro	qwen+llama+phi	3	0.6506	0.6158	0.6192	0.7528	Sum
MMLU-Pro	qwen+llama+tulu+dsmath+phi+ministral	6	0.6502	0.6158	0.5660	0.8274	Sum
MMLU-Pro	qwen+llama+tulu+dsmath+phi	5	0.6486	0.6158	0.5672	0.8080	Sum
MMLU-Pro	qwen+llama+tulu+phi+ministral	5	0.6478	0.6158	0.5804	0.7992	Sum