Routing with Generated Data: Annotation-Free LLM Skill Estimation and Expert Selection

Tianyi Niu[1], **Justin Chih-Yao Chen**[1], **Genta Indra Winata**[2], **Shi-Xiong Zhang**[2], **Supriyo Chakraborty**[2], **Sambit Sahu**[2], **Yue Zhang**[1], **Elias Stengel-Eskin**[3], **Mohit Bansal**[1]
[1]UNC Chapel Hill [2]Capital One [3]The University of Texas at Austin

Abstract

Large Language Model (LLM) routers dynamically select optimal models for given inputs. Existing approaches typically assume access to ground-truth labeled data, which is often unavailable in practice, especially when user request distributions are heterogeneous and unknown. We introduce Routing with Generated Data (RGD), a challenging setting in which routers are trained exclusively on generated queries and answers produced from high-level task descriptions by generator LLMs. We evaluate query-answer routers (using both queries and labels) and query-only routers across four diverse benchmarks and 12 models, finding that query-answer routers degrade faster than query-only routers as generator quality decreases. Our analysis reveals two crucial characteristics of effective generators: they must accurately respond to their own questions, and their questions must produce sufficient performance differentiation among the model pool. We then show how filtering for these characteristics can improve the quality of generated data. We further propose CASCAL, a novel query-only router that estimates model correctness through consensus voting and identifies model-specific skill niches via hierarchical clustering. CASCAL is substantially more robust to generator quality, outperforming the best query-answer router by 4.6% absolute accuracy when trained on weak generator data.[1]

1 Introduction

As the ecosystem of large language models (LLMs) matures, there is an increasing number of open-weight LLMs available to the public (Sahana, 2025). While leaderboards provide aggregate pictures of model performance, a growing body of work in LLM routing has proposed ways to *dynamically* determine which model(s) to select for a given *instance*. At its core, routing relies on recognizing each model's fine-grained strengths and weaknesses, i.e., estimating model skills and selecting experts. Most prior routing methods generally follow one of three paradigms: (1) training a classifier to select models given an input (Agrawal and Gupta, 2025; Ong et al., 2025; Tsiourvas et al., 2025; Chen et al., 2024b), (2) profiling models using inferred natural language skills (Chen et al., 2025b,c; Maimon et al., 2025; Shah and Shridhar, 2025; Dong et al., 2025), or (3) clustering-based approaches that identify skills in the embedding space (Zhang et al., 2025c; Jitkrittum et al., 2025; Pichlmeier et al., 2024). Crucially, these ap-

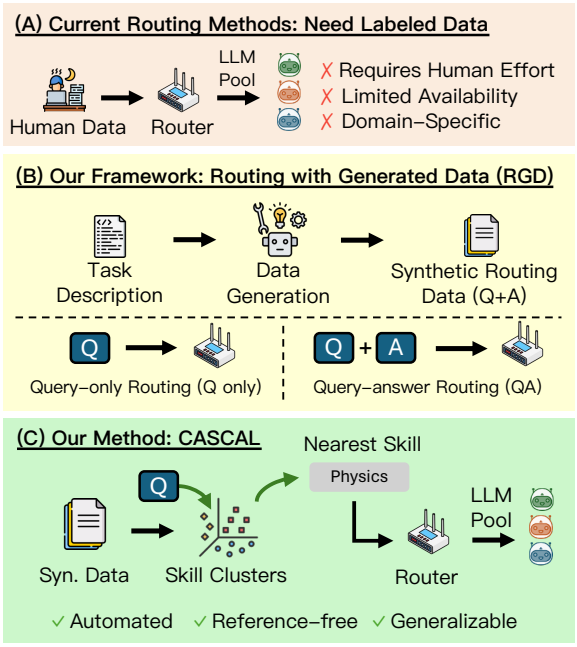

Figure 1: Overview of Routing with Generated Data (RGD). (A) Most existing routers require human-labeled data for skill estimation and expert selection. (B) RGD generates routing data from task descriptions, enabling both query-only and query-answer routing. (C) CASCAL extracts skill clusters from generated queries and routes to models without ground-truth labels.

[1]Code is available at https://github.com/tianyiniu/RoutingGenData.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 32441–32466

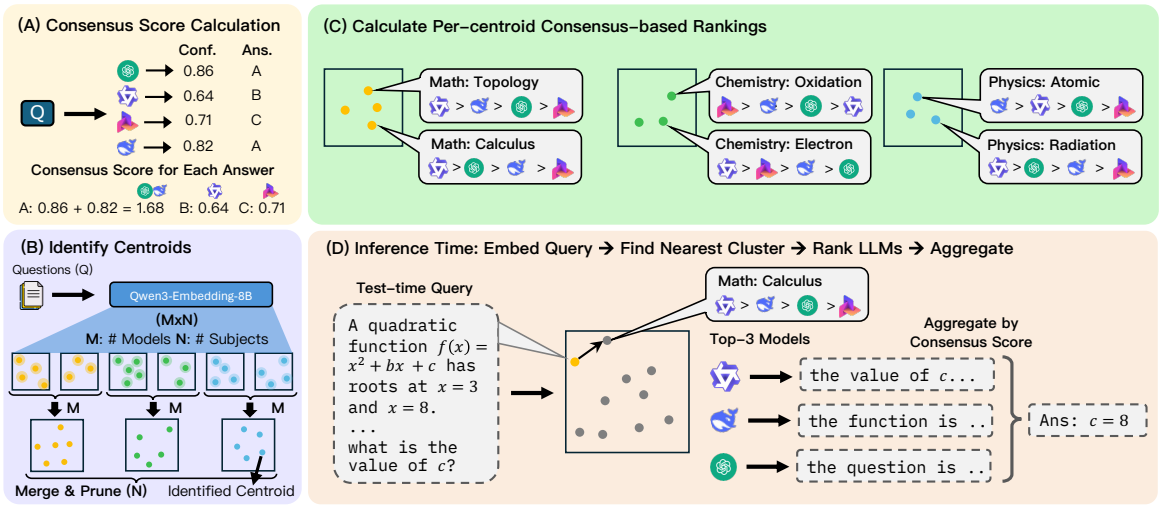

Figure 2: Overview of CASCAL. (A) Consensus Scoring: we extract model responses for each query and compute confidence-weighted consensus scores. (B) Centroid Identification: For each model and task, we cluster queries where the model demonstrates proficiency to obtain skill centroids, then we merge similar centroids across models. (C) Cluster Ranking: we assign queries to their nearest centroid and rank models within each cluster by average consensus score. (D) Inference: we route test queries to the nearest task and centroid, select the top-3 (or top-1) ranked models, and aggregate responses via consensus voting.

proaches all *assume access to labeled, in-domain data*, either for training and validating the classifier, or for inferring and validating skills. However, in real-world settings where the distribution of user requests is highly heterogeneous and not known *a priori*, this assumption may not hold, and data may be difficult to obtain. In other words, routing decisions may need to be made *without ground-truth in-domain data* (Figure 1A).

To capture this scenario, we lift the assumption of ground-truth data and introduce **Routing with Generated Data (RGD)**, a challenging new routing setting (Figure 1). RGD requires task-driven data generation, for which we provide a series of data domains, each with a natural language domain description, a pool of candidate models, and a held-out test set. Rather than providing in-domain data, RGD generates query-answer data pairs that seek to differentiate between strong and weak models. The generated queries – and, if required, their answers – are then used to fit routers to the model pool (see Figure 1B).

Given the strength of LLMs in generating data (Li et al., 2024b; Khan et al., 2025; Long et al., 2024; Nadăş et al., 2025), we ask *(RQ1)* how existing routers adapt to RGD, *(RQ2)* what makes generated profiling data effective, and *(RQ3)* how such data can be improved. Our analysis reveals two key insights: (1) weaker generators produce unreliable answer labels but useful queries, and (2) effective routing requires identifying fine-grained skill niches where models differ. Building on these findings, we develop a novel label-free routing method: Consensus-Aware Skill Clustering and Aggregation for LLMs (CASCAL) (see Figure 1C and Figure 2).

CASCAL is characterized by two core ideas: **(1) Consensus-based correctness estimation:** Given that majority vote is often a strong signal for correctness (Wang et al., 2022; Chen et al., 2024a), we calculate a confidence-weighted majority vote, yielding a continuous metric of how closely each model aligns with the majority opinion for a given input (Figure 2A).[2] **(2) Hierarchical clustering to identify niche skills:** To better capture model differences on niche topics, we propose a hierarchical clustering approach that groups queries into skill clusters and identifies niche skills for each model (Figure 2B). For each cluster centroid, we rank models by their average consensus score across all queries assigned to that cluster (Figure 2C). At test time, a new query is routed to its nearest centroid, and the top-3 models for that cluster are selected as experts (Figure 2D).

We evaluate CASCAL and other routing baselines on two model pools – one composed of larger models (>20B parameters), and one composed of smaller models (<10B) – and across diverse bench-

[2]This metric can also be understood as a form of self-consistency across multiple models.

marks, testing on MMLU-Pro (Wang et al., 2024b), MedMCQA (Pal et al., 2022), SuperGPQA (M-A-P Team et al., 2025), BigBench-Extra-Hard (Kazemi et al., 2025). We contrast CASCAL with competitive query-answer routing baselines (i.e., methods that use both queries and answers) like LLMRANK (Agrawal and Gupta, 2025) and AVENGERS (Zhang et al., 2025c), as well as another strong query-only baseline, SMOOTHIE-TRAIN (Guha et al., 2024).

First, addressing *RQ1*, we find that, while most routing methods perform similarly with ground-truth data, existing answer-based routing methods perform poorly in the RGD setting. For example, with data generated by Exaone-3.5-7.8B-Instruct, CASCAL achieves an average accuracy of 61.1%, whereas LLMRANK and AVENGERS achieve 57.1% and 58.9%, respectively, in the large model pool. Second, for *RQ2*, we find that query-only methods are more robust to data quality, and that weaker LLMs are more adept at generating routing queries than generating answers to those queries. Finally, addressing *RQ3*, we use data filtering to assess which qualities of the generated data transfer well. We find that filtering for variance-inducing questions with high consensus by stronger models produces more informative routing samples and improves CASCAL's performance. Taken together, our work provides further evidence that the strengths of a diverse model pool can be effectively estimated *without* ground-truth annotations.

2 Routing Formulation under RGD

We formalize the routing problem in the RGD setting as follows. Let $\mathcal{M} = \{m_1, \ldots, m_M\}$ be a pool of M candidate LLMs. Given a query q from a domain $\mathcal{D}$, the goal of a router π is to select a subset of K models from $\mathcal{M}$ that maximizes performance on q. Formally, the router learns a mapping:

$$\pi : q \rightarrow \{m_{i_1}, \ldots, m_{i_K}\} \subseteq \mathcal{M}, \qquad (1)$$

where the chosen models may be used alone ($K = 1$), aggregated via ensemble methods such as majority voting, or via a separate aggregator model.

2.1 Training and Inference Protocols

During training, routing methods construct internal representations of model capabilities using a training dataset $\mathcal{D}_{\text{train}} = \{(q_i, a_i^*, \{r_i^{(j)}\}_{j=1}^{M})\}_{i=1}^{N}$, where q_i is a query, a_i^* is an answer, and $r_i^{(j)}$ is the response from model m_j. Methods differ in what information they utilize:

- **Query-answer methods** assume access to the full tuple $(q, a^*, \{r^{(j)}\})$, using ground-truth labels to compute correctness signals.
- **Query-only methods** assume access only to $(q, \{r^{(j)}\})$, forgoing ground-truth labels a^*.

At inference time, given a test query q_{new}, the router selects the best model(s) based on the capabilities inferred during training.

2.2 Routing with Generated Data (RGD)

In RGD, routers are trained on model-generated data. We construct datasets by prompting a generator model to produce a training dataset $\{\hat{q}_i, \hat{a}_i\}$ from a set of task descriptions that briefly highlight each task's primary topics. Given a description d_t, the generator model m_{gen} generates a single output containing a query-answer pair conditioned on this description: $(\hat{q}, \hat{a})_{\text{synthetic}} = m_{\text{gen}}(d_t)$. From this, we can obtain $\hat{\mathcal{D}}_{train}$ by adding model responses from the model pool, with which we can train a router. The generator model m_{gen} may or may not be included in the model pool $\mathcal{M}$.

3 Routing Methodology: CASCAL

To tackle the challenges of RGD, we propose CASCAL (Figure 2), a consensus-based query-only router. The method builds on two core insights: (1) consensus among models serves as a reliable proxy for correctness, and (2) hierarchical clustering can identify model-specific "niche" skills where individual models excel. Unlike query-answer routing methods that require labeled data to compute correctness signals, CASCAL derives quality estimates entirely from model agreement patterns, making it well-suited to RGD where generated answers may be unreliable. Section 3.1 outlines how we derive the consensus score – the key scoring metric in CASCAL. Next, Section 3.2 describes the training and inference steps of the router.

3.1 Consensus Score Calculation

To estimate model accuracy in the absence of ground-truth labels, we employ a consensus scoring method that accounts for both the agreement between models and their individual confidence levels. Let $\mathcal{Q} = \{q_1, \ldots, q_n\}$ denote the set of queries and $\mathcal{M} = \{m_1, \ldots, m_M\}$ denote the set of models available in the model pool. For a given query q_i and model m_j, let $r_{i,j}$ represent the model's generated response and $L_{i,j}$ the corresponding log-probability score. From $r_{i,j}$, we extract the model's

answer $a_{i,j}$. For each model m_j, we calculate the mean μ_j and standard deviation σ_j of its log-probabilities across the entire dataset, then normalize into a Z-score $Z_{i,j} = (L_{i,j} - \mu_j)/\sigma_j$.

We define the consensus score $C_{i,j}$ for the answer provided by model m_j on query q_i as the sum of the normalized confidence scores from all models in the ensemble that generated an identical answer. Formally, this is given by $C_{i,j} = \sum_{k=1}^{M} \mathbb{I}(a_{i,j} = a_{i,k}) \cdot Z_{i,k}$. This formulation ensures that an answer is rewarded not only by the number of models that agree with it but also by the relative confidence of those models.

3.2 CASCAL Training and Inference

Training. For each task t, the training phase seeks to construct two mappings: (1) $\mathcal{F}^t_{\text{centroid}}$, which maps a discrete CLUSTERID to a dense Centroid Vector; and (2) $\mathcal{F}^t_{\text{models}}$, which maps a CLUSTERID to a ranked list of model experts. Models are ranked by their average consensus score over the queries associated with that specific cluster. We obtain these mappings using the following procedure:

1. **Identifying consensus-aligned queries:** For each model m and task t, we isolate a subset of queries $Q^{\text{strong}}_{m,t}$ where the model aligns with the majority consensus:

$$Q^{\text{strong}}_{m,t} = \left\{ q_i \in Q_t : a_{i,m} = a_i^{maj} \right\} \quad (2)$$

$$a_i^{maj} = \arg\max_a \sum_{k=1}^{M} \mathbb{I}(a_{i,k} = a) \quad (3)$$

   If ground-truth labels are available, this subset instead consists of queries the model answered correctly (CASCAL-GT).

2. **Finding Centroids:** We compute embeddings $E(Q^{\text{strong}}_{m,t})$ for these queries and apply k-means clustering to identify a set of skill centroids $\mathcal{C}^t_m$ specific to model m. We select $k \in \{2, \ldots, 5\}$ that maximizes the silhouette score; if no k exceeds the threshold of 0.05, we use a single centroid. We aggregate these centroids across models to form the global set $\mathcal{C}^t$. Each centroid $c \in \mathcal{C}^t$ is assigned a unique identifier (CLUSTERID).

3. **Merging Close Centroids:** We merge centroids that are within cosine distance $\tau_{\text{merge}} = 0.15$ of each other. When merging, we compute a weighted average based on seed count (number of queries that formed each centroid).

4. **Clustering and Mapping:** We assign every query in the training set to its nearest task-specific centroid $c^* \in \mathcal{C}^t$ based on cosine similarity. This partitions the task into distinct skill clusters.[3] Within each cluster, we calculate the average consensus score of every model and generate a ranked list of experts.

5. **Pruning Redundant Centroids:** We remove centroids whose top-3 model rankings are nearly identical (Jaccard similarity ≥ 0.95), keeping the centroid with more assigned queries. After pruning, we reassign queries and recompute centroids as the mean of their assigned query embeddings to ensure geometric consistency. These associations form the final mappings $\mathcal{F}^t_{\text{centroid}}$ and $\mathcal{F}^t_{\text{models}}$.

Inference. At inference time, an incoming user query q_{new} is routed through a three-step pipeline to select and aggregate the most suitable agents:

1. **Task and Centroid Routing:** We first identify the training query q_{train} with the nearest embedding to q_{new} and inherit the task label t from q_{train}. Next, we compute the cosine similarity between $E(q_{\text{new}})$ and the vectors in $\mathcal{F}^t_{\text{centroid}}$. The query is assigned to the CLUSTERID corresponding to the nearest centroid.

2. **Agent Selection:** We query the mapping $\mathcal{F}^t_{\text{models}}$ using the identified CLUSTERID to retrieve the ranked list of expert agents. The top $K = 3$ models from this list are selected as the active ensemble for q_{new}.

3. **Consensus Score Aggregation:** We calculate the consensus score of the K selected agents.[4] The answer with the highest final consensus score is selected as the output.[5]

CASCAL Variants. In addition, we evaluate two variants of CASCAL. First, **CASCAL (Top-1)** omits the aggregation stage and directly routes to the single top-ranked model ($K = 1$), enabling comparison against routers that select a single

[3] We provide a qualitative visualization of learned skill clusters from SuperGPQA in Appendix A.

[4] To prevent data leakage, we normalize the log-probabilities using the pre-calculated μ_j and σ_j from the training split.

[5] Aggregating via consensus score is very similar to majority voting in practice. Using the former, the most confident model's answer will be selected when there is no majority choice.

model per query. Second, **CASCAL-GT** replaces consensus scores with a ground-truth label for ranking models within clusters during training, making CASCAL-GT a query-answer router.

4 Experimental Results

4.1 Experiment Setup

Model Pools. We evaluate routing methods on two disjoint pools of six models each, one large-scale pool with models > 20B parameters (POOL-LARGE), and one small-scale pool with models < 10B (POOL-SMALL). Full model lists are summarized in Table 4. We use Qwen3-Embedding-8B (Zhang et al., 2025b) to obtain query embeddings for the clustering pipeline.

Datasets and Tasks. To comprehensively examine how generated data affects routing quality, we construct scenarios using RGD across four distinct datasets: **MMLU-Pro** (Wang et al., 2024b), **MedMCQA** (Pal et al., 2022), **SuperGPQA** (M-A-P Team et al., 2025), and **BigBench-Extra-Hard** (BBEH; Kazemi et al., 2025). We subdivide each dataset into tasks; for MMLU-Pro, MedMCQA, SuperGPQA, each task represents a subject. Additionally, we evaluate CASCAL on MATH (Hendrycks et al., 2021) in Appendix D.1. We divide each dataset into 6:4 train-test splits, stratified by tasks.

RGD Data Generation. We evaluate routing methods across four RGD scenarios, each defined by a different generator model m_{gen}: Real data, Gemini-2.5-Flash, Qwen3-32B, and Exaone-3.5-7.8B-Instruct. These models were chosen to represent one frontier LLM, one stronger open-weight model, and one smaller 8B parameter model. We provide some additional notes regarding generator selection in Appendix C.4. First, for each scenario, we obtain a set of task descriptions by prompting a descriptor model to summarize a given task t using a small number of seed examples.[6] We include all task descriptions in Appendix F.2 and prompts in Appendix F.3. We manually verify that the generated descriptions do not leak validation examples. Next, we independently prompt each generator to generate 5,000 queries $\hat{q}$ and answers $\hat{a}$ per domain.

[6]We use Gemini-2.5-Flash as the descriptor model with 5 in-context queries for all experiments. This is for efficiency; task descriptions can also be human-written. In Appendix D.6, we further show that the pipeline demonstrates a fair degree of robustness towards different initial task descriptions.

4.2 Baselines

Learning-free Baselines. **Top-1** selects the single best-performing model as measured by average validation set performance for all tasks. **Top-3 Vote** applies majority voting over the top three models. These baselines provide a strong reference point in the RGD setting, as they use real data to determine the top-1 and top-3 models. **Random-1** samples a model from the pool, and **Random-3 Vote** applies majority voting over three sampled models.

Query-answer Routers. Routing methods that assume access to ground-truth labels during training represent the standard paradigm in LLM routing, where correctness signals are used to learn model strengths. We evaluate the following representative approaches:

- **LLMRANK** (Agrawal and Gupta, 2025): Extracts interpretable features from prompts (task type, complexity, etc.) and trains a neural ranking model to predict per-model capability scores. We document the features we used in Section C.3 and denote our implementation as LLMRANK.
- **AVENGERS** (Zhang et al., 2025c): Embeds queries, clusters them via k-means ($k = 64$), and scores each model's accuracy per cluster. At test time, queries are routed to the nearest cluster, the top-3 models are selected, and their outputs are aggregated via a majority vote.

For all query-answer routers, training involves computing correctness signals $\mathbb{1}[r_i^{(j)} = a_i^*]$ for each model response and using these to fit routers. In the RGD setting, this becomes $\mathbb{1}[r_i^{(j)} = \hat{a}_i]$, where $\hat{a}_i$ is a model-generated answer.

Query-only Routers. Query-only routers operate without ground-truth labels and are particularly suited to RGD settings as generated data may lack reliable answer labels. We also consider **SMOOTHIE** (Guha et al., 2024), a weak supervision-inspired approach that models embedding differences between LLM outputs as a multivariate Gaussian to derive per-model quality scores. Since we evaluate routing under domain shift, we use the SMOOTHIE-TRAIN variant of the algorithm. As CASCAL by default aggregates 3 models, we further show results from majority voting using the top-3 ranked by SMOOTHIE-TRAIN. For query-only methods, training involves computing proxy quality metrics from model responses $\{r_i^{(j)}\}_{j=1}^{M}$ without reference to ground-truth.

4.3 Main Results and Discussions

4.3.1 RQ1: How do routing methods adapt to the RGD setting?

We present our main results in Figure 3, which compares the average performance of different routing methods across all RGD scenarios for both model pools, with three key observations. We present detailed results for all routers in Appendix D.

Stronger generators consistently produce better routers. Stronger generators produce training data that yields superior routing performance. Weak generators never outperform stronger ones and generated data almost never outperforms ground-truth data, with some rare exceptions for query-only methods trained on Gemini-generated data. In POOL-SMALL, CASCAL Top-1 improves from 46.6% to 47.7% (+1.1%), SMOOTHIE-TRAIN Top-1 improves from 43.4% to 44.4% (+1.0%).[7] This suggests that high-quality generated data can occasionally match or exceed real data for fitting routers, particularly for query-only methods that do not rely on potentially noisy generated labels.

CASCAL is more robust to generator degradations. Among query-only methods, CASCAL consistently outperforms SMOOTHIE-TRAIN across all RGD scenarios in both model pools. On POOL-LARGE, CASCAL achieves 63.1%, 61.1%, and 61.1% accuracy under Gemini, Qwen, and Exaone generators respectively, compared to SMOOTHIE-TRAIN Top-3's 59.9%, 59.2%, and 59.4%. On POOL-SMALL, CASCAL similarly beats SMOOTHIE-TRAIN Top-3 across all scenarios (46.5% vs 46.1% for Gemini, 45.6% vs 45.0% for Qwen, and 45.3% vs 43.4% for Exaone). Critically, on POOL-LARGE, CASCAL is the only method to consistently match or exceed the Random-3 Vote baseline across all RGD scenarios. With Exaone-generated data on POOL-SMALL, query-answer methods like AVENGERS (40.7%) and LLMRANK (41.0%) fall to or below the Random-3 Vote baseline of 41.6%, while CASCAL (45.3%) maintains a substantial margin above it. This makes CASCAL valuable in budget-constrained cases where only weak generators are available.

Weak generators disproportionately harm query-answer routers. We observe a clear asymmetry between query-answer and query-only methods when generator quality degrades. On POOL-LARGE, query-answer methods suffer substantial accuracy losses between validation and Exaone-generated data: LLMRANK drops 8.5% (65.6% → 57.1%), CASCAL-GT Top-1 drops 9.2% (65.7% → 56.5%), and AVENGERS drops 4.5% (63.4% → 58.9%). In contrast, query-only methods exhibit much higher stability: CASCAL drops only 2.5% (63.6% → 61.1%), while SMOOTHIE-TRAIN Top-3 actually remains consistent (59.3% → 59.4%). This pattern is even more pronounced on POOL-SMALL, where query-answer methods experience severe degradation: AVENGERS drops 9.5% (50.2% → 40.7%), CASCAL-GT Top-1 drops 10.5% (49.7% → 39.2%), and LLMRANK drops 7.8% (48.8% → 41.0%). Similarly, query-only methods remain far more resilient: CASCAL drops only 1.1% (46.4% → 45.3%), maintaining its performance even under the weakest generator. These results demonstrate that query-only methods are more robust to generated data quality, making them better suited to RGD settings with unreliable labels from weaker generators.

4.3.2 RQ2: What are the characteristics of strong profiling data in RGD?

Generator	MMLU-Pro	MedMCQA
Exaone-3.5 7.8B	65.6	75.4
Qwen3-32B	**75.1**	**79.0**

Table 1: Generator label quality measured using agreement with Gemini-3-Flash answers. The stronger generator (Qwen3-32B) has consistently higher agreement.

Here, we analyze why stronger generators lead to better routers. We show that (1) weak generators struggle to answer their own queries, and (2) queries from stronger generators induce better model rankings.

Weaker generators struggle to answer their own queries. For each generated query, we compare the generated answer provided by the generator model (either Exaone-3.5-7.8B-Instruct or Qwen3-32B) against an answer obtained using Gemini-3-Flash as a "silver" reference. Here, we make the assumption that Gemini-3-Flash is a much stronger QA model (i.e., a better approximation to a "ground-truth" label) compared to the generator models on MMLU-Pro and MedMCQA; this is based on its high validation accuracy on these datasets (86% and 97%, respectively). To further ensure the quality of Gemini's output, we generate 3 responses per query and take the majority

[7] All percentage differences are absolute.

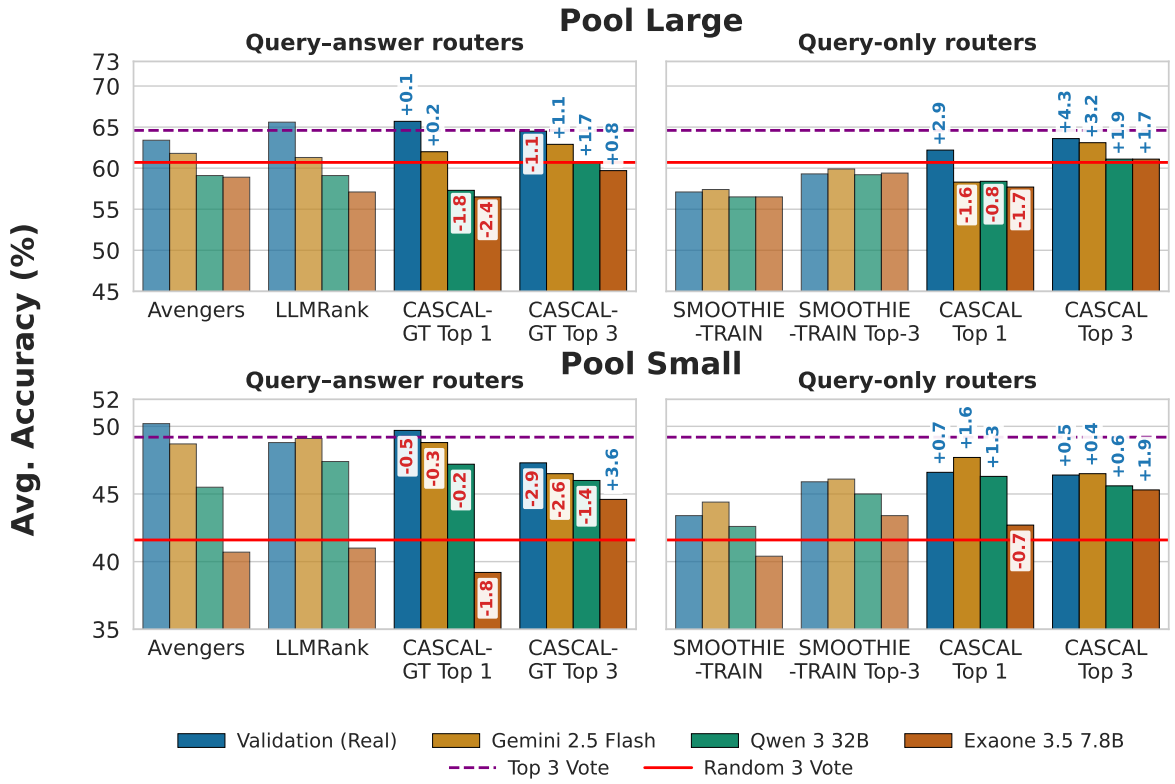

Figure 3: Routing accuracy across RGD scenarios for POOL-LARGE (left) and POOL-SMALL (right). Colors indicate the source of routing data: validation data or data generated by different LLMs. Each bar represents the router's average test accuracy across four datasets (MMLU-Pro, SuperGPQA, MedMCQA, and BigBench-Extra-Hard). Annotations indicate the absolute accuracy improvement of CASCAL variants over the strongest non-CASCAL baseline within the same routing family (subplot) under the same data source (color).

vote response. We exclude BBEH and SuperGPQA from this analysis as Gemini-3-Flash itself cannot reliably answer the queries (Section D.5). This experiment allows us to estimate each generator's answer quality and understand how label errors propagate to router performance.

Results. In Table 1, we find that weaker generators produce substantially less accurate labels on queries they themselves generated. Exaone-3.5-7.8B-Instruct achieves only 65.6% agreement on MMLU-Pro and 75.4% on MedMCQA with Gemini-3-Flash. In contrast, Qwen3-32B achieves higher agreement: 75.1% on MMLU-Pro and 79.0% on MedMCQA. These errors result in noise for query-answer routers, leading to degraded performance compared to query-only routers.

Weaker generators struggle to generate difficult queries that isolate strong models. To understand why query-only methods also exhibit performance degradation in RGD scenarios with weak generators, we analyze whether the generated data preserves the relative model rankings (by average consensus score on the dataset) needed for effective routing. Specifically, we ask: *do model rankings derived from generated queries correlate with rankings from real validation data?* For each RGD scenario (validation, Gemini, Qwen, Exaone), we compute model rankings by averag-

GEN.	MMLU	SGPQA	MMCQA	BBEH	AVG.
Pool Large					
GEM.	**.47**	.33	**.60**	**-.20**	**.30**
QWEN	**.47**	**.47**	.07	-.89	.03
EXA.	-.07	-.33	.33	-.47	-.14
Pool Small					
GEM.	**1.0**	**1.0**	**.87**	.47	**.84**
QWEN	.87	**1.0**	**.87**	**.60**	**.84**
EXA.	.87	.87	.60	**.60**	.74

Table 2: Kendall's τ between model rankings from generated vs. validation data. Stronger generators induce better rankings, and all generators induce better rankings for small models than large ones.

ing each model's consensus score across all queries. We then measure the agreement between validation-derived rankings and generated-data-derived rankings using Kendall's τ correlation coefficient. A τ close to 1 indicates that the generated data induces model rankings consistent with real data, while τ near 0 or negative suggests the generated queries fail to differentiate models in a meaningful way.

Results. Table 2 reveals that ranking quality degrades with weaker generators in both POOL-SMALL and POOL-LARGE. Noticeably, on POOL-LARGE, ranking correlation degrades substantially. Gemini achieves only $\tau = 0.3$ on average, Qwen drops to $\tau = 0.03$ (near random), and Exaone produces negatively correlated rankings ($\tau = -0.14$). The degradation is particularly severe on BBEH, where all generators produce negative correlations (ranging from -0.2 to -0.89). In POOL-SMALL all generators maintain strong ranking correlation with validation data (avg. $\tau > 0.7$), explaining why CASCAL degrades less in POOL-SMALL. This asymmetry between model pools may arise because distinguishing among stronger models requires sufficiently challenging queries. When weaker generators produce queries that larger models answer uniformly well, this collapses the variance needed for differentiation, while smaller models may exhibit varied performance even on easier queries. Nevertheless, we observe that while routers outperform random baselines on POOL-LARGE less often than POOL-SMALL, they can still do so despite low ranking correlations. We identify that this is due to the top 2 rank positions being relatively stable and accurate, even in low-quality data. Therefore, routers that capture these top positions can still achieve effective routing.

4.3.3 RQ3: What factors can improve the quality of generated profiling data?

Setup. Our analysis in RQ2 reveals that weaker generators may fail to produce queries that meaningfully differentiate between all models in the pool, but may still correctly rank the top models. This raises the question of distribution sharpening: can we recover routing performance by filtering generated data to retain only high-quality queries?

To test this, we generate 20k queries using Exaone-3.5-7.8B-Instruct, then apply two filtering criteria. We first compute each model's average consensus score across all 20k queries and designate the two highest-scoring models as the "top-2" models. Next, we filter the 20k queries, retaining a query if we find: (1) **Strong model agreement:** Both top-2 models align with the majority answer among the model pool, ensuring the consensus forms around a likely correct response. (2) **Sufficient difficulty:** At most two other models share this majority answer, ensuring the query is challenging enough that weaker models fail. Together, these criteria select queries where strong models reliably succeed while weaker models diverge. After filtering, we obtain approximately 3k MMLU-Pro queries, 1.8k SuperGPQA queries, 2.5k MedMCQA queries, and 4.2k BBEH queries.

METHOD	MMLU	SGPQA	MMCQA	BBEH	AVG.
Validation					
TOP-1	78.9	53.4	85.7	30.7	62.2
TOP-3	81.0	53.8	86.5	33.0	63.6
Exaone (5k)					
TOP-1	74.6	42.6	81.6	**32.1**	57.7
TOP-3	78.7	49.2	84.1	32.3	61.1
Exaone (20k)					
TOP-1	75.3	41.6	**86.7**	**32.4**	58.9
TOP-3	77.8	49.6	83.9	**33.6**	61.2
Exaone+Filter					
TOP-1	75.1	44.8	**85.9**	**32.6**	59.6
TOP-3	78.2	52.7	86.1	32.1	62.3

Table 3: CASCAL performance on POOL-LARGE with consensus-based filtering. Filtering data generated by Exaone-3.5-7.8B-Instruct recovers performance, at times exceeding real data (bold).

Results. Table 3 shows results on POOL-LARGE, comparing the average accuracy of CASCAL trained using validation questions, 5k Exaone-generated questions, 20k Exaone-generated questions, and the filtered questions. CASCAL trained using 5k unfiltered Exaone-generated queries yields 57.7% (Top-1) and 61.1% (Top-3) average accuracy, representing drops of 4.5% and 2.5% compared to validation data. After filtering, performance recovers to 59.6% (Top-1) and 62.3% (Top-3), respectively. Notably, filtered Exaone data outperforms unfiltered data on SuperGPQA by 3.5% (Top-3) and on MedMCQA by 2.0% (Top-3), demonstrating that targeted filtering can improve routing quality even when starting from a weak generator.[8] These results point to a promising future direction: optimizing data generation for routing quality, e.g., via reinforcement learning approaches

[8]We also demonstrate size-matched domain-specific generators can lead to stronger data compared to generalist models in Appendix D.2.

that reward query differentiation.

5 Related Work

LLM Routing. Prior routing approaches fall into multiple classes, including work training classifiers on human preferences or ground-truth labels to predict the best model for a given query (Ong et al., 2025; Agrawal and Gupta, 2025; Yu et al., 2025), or using reinforcement learning to handle multi-round routing and aggregation (Zhang et al., 2025a). Other work focuses on maximizing the cost-to-performance trade-off, e.g., Chen et al. (2024b) introduce a cascade going from cheaper to costlier models. More in line with RGD, a subset of work has explored routing without ground-truth signals (Guha et al., 2024; Jitkrittum et al., 2025); these approaches have not explored generated data. CASCAL differs from these lines of prior work as (1) we use consensus to rank models, (2) we propose a hierarchical clustering approach that dynamically finds clusters of queries that represent a specific "skill", and most importantly (3) CASCAL is designed for the RGD setting.

Multi-agent frameworks. Multi-agent frameworks use ensembles of LLMs, as in our Top-3 settings. A prominent example is Multi-Agent Debate (MAD), where agents generate arguments over multiple rounds to arrive at a superior consensus-based solution (Du et al., 2024; Liang et al., 2024; Chen et al., 2024a; Xiong et al., 2023; Chan et al., 2023). Similar model ensembling approaches have been applied without debate (Wang et al., 2024a), including in routing settings where expert models are chosen based on natural language skill descriptors (Chen et al., 2025b). Our work complements these prior efforts by investigating how expert agents can be selected in the absence of validation data.

Data generation. High-quality generated data is crucial for efficiently training and aligning LLMs. Prior work has examined models generating their own synthetic data (Zelikman et al., 2022) as well as creating large-scale, high-quality datasets for tasks like mathematical reasoning (Yu et al., 2024; Li et al., 2024a; Chen et al., 2025a). Such approaches have been instrumental in building frontier LLMs (Abdin et al., 2024; Yang et al., 2025). Khan et al. (2025) explore data generation as an interactive task, where a teacher agent, conditioned on the student model's current errors and skills, plans and synthesizes new training examples to maximally improve the student over iterative re-training loops. Our work addresses a gap in the generation literature, exploring how generated data can be used in LLM routing.

6 Conclusion

We introduce Routing with Generated Data (RGD), a challenging setting for LLM routing that lifts the assumption of access to ground-truth labeled data. Evaluating across four diverse benchmarks and two model pools with 6 models each, we show that existing answer-dependent routing methods suffer significant performance degradation when trained on generated data, particularly from smaller generator LLMs. In other words, query-answer routing is preferable when a strong generator can be identified, whereas query-only routing is more reliable when generator quality is uncertain. Our proposed method, CASCAL, addresses this tension by avoiding answer labels in favor of consensus-based correctness estimation and using hierarchical clustering to identify model-specific skill niches. CASCAL achieves improved robustness to generated data quality when compared to answer-based and query-only baselines. Our analysis further reveals that weaker LLMs are better at generating informative routing queries than answering those queries, explaining why query-only methods exhibit greater resilience in RGD settings. Moreover, stronger generator models induce better rankings over both small and large model pools. Finally, we show that a filtered subset of high-quality queries, even when generated by small LLMs, can result in performance comparable to real queries. This work opens new directions for generalizable LLM routing in settings where user request distributions are unknown and labeled data is unavailable.

Limitations

We note some limitations that suggest directions for future research. First, as previously mentioned in Section 4.1, our task descriptions are LLM-generated instead of hand-written. To ensure that they are high-quality and consistent with the target tasks, we have manually verified these descriptions and provided them in Appendix F. At the same time, we provide results using alternative descriptions, demonstrating that RGD exhibits a degree of robustness to initial task descriptions (Appendix D.6). Second, we only evaluated on English-language benchmarks with closed-ended questions. Future work can extend RGD to ad-

dress open-ended long-form generations and explore routing in multi-lingual settings. Finally, we show that a data filtering approach can improve the discriminativeness of generated data, leading to better downstream routers. Future experiments can examine the possibility of directly optimizing the generator model via RL.

Ethical Considerations

Our work is focused on routing and combining multiple existing models via data generation. As such, we do not foresee any particular ethical considerations beyond those that apply to the LLMs (both in the pool and the generator models) themselves.

Acknowledgements

We thank Bill Campbell, Stephen Rawls, Anirban Das, and Kartik Balasubramaniam for their feedback. This work was supported by NSF-CAREER Award 1846185, NSF AI Engage Institute DRL-2112635, and Capital One Faculty Award. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the sponsors.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Shubham Agrawal and Prasang Gupta. 2025. Llm-rank: Understanding llm strengths for model routing. *Preprint*, arXiv:2510.01234.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, and 14 others. 2025. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

Meta AI. 2024. Llama 3.3 70b | model card and prompt formats. https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/. Text-only 70B instruction-tuned model.

Zhipu AI and Tsinghua University KEG. 2024. Glm-4: Open multilingual multimodal chat large language models. https://github.com/zai-org/GLM-4. Model: GLM-4-32B-0414.

Soyoung An, Kyunghoon Bae, Eunbi Choi, Kibong Choi, Stanley Jungkyu Choi, Seokhee Hong, Junwon Hwang, Hyojin Jeon, Gerrard Jeongwon Jo, Hyunjik Jo, Jiyeon Jung, Yountae Jung, Hyosang Kim, Joonkee Kim, Seonghwan Kim, Soyeon Kim, Sunkyoung Kim, Yireun Kim, Yongil Kim, and 13 others. 2026. Exaone 3.5: Series of large language models for real-world use cases. *Preprint*, arXiv:2412.04862.

Kyunghoon Bae, Eunbi Choi, Kibong Choi, Stanley Jungkyu Choi, Yemuk Choi, Kyubeen Han, Seokhee Hong, Junwon Hwang, Taewan Hwang, Joonwon Jang, Hyojin Jeon, Kijeong Jeon, Gerrard Jeongwon Jo, Hyunjik Jo, Jiyeon Jung, Euisoon Kim, Hyosang Kim, Jihoon Kim, Joonkee Kim, and 21 others. 2026. Exaone 4.0: Unified large language models integrating non-reasoning and reasoning modes. *Preprint*, arXiv:2507.11407.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *Preprint*, arXiv:2308.07201.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. 2024a. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085, Bangkok, Thailand. Association for Computational Linguistics.

Justin Chen, Zifeng Wang, Hamid Palangi, Rujun Han, Sayna Ebrahimi, Long Le, Vincent Perot, Swaroop Mishra, Mohit Bansal, Chen-Yu Lee, and Tomas Pfister. 2025a. Reverse thinking makes LLMs stronger reasoners. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8611–8630, Albuquerque, New Mexico. Association for Computational Linguistics.

Justin Chih-Yao Chen, Sukwon Yun, Elias Stengel-Eskin, Tianlong Chen, and Mohit Bansal. 2025b. Symbolic mixture-of-experts: Adaptive skill-based routing for scalable heterogeneous reasoning. *arXiv preprint arXiv:2503.05641*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2024b. Frugalgpt: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*.

Zhou Chen, Zhiqiang Wei, Yuqi Bai, Xue Xiong, and Jianmin Wu. 2025c. TagRouter: Learning route to LLMs through tags for open-domain text generation

tasks. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21539–21564, Vienna, Austria. Association for Computational Linguistics.

Jiangwen Dong, Zehui Lin, Wanyu Lin, and Mingjin Zhang. 2025. S-dag: A subject-based directed acyclic graph for multi-agent heterogeneous reasoning. *Preprint*, arXiv:2511.06727.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

GLM Team, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, and 40 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Neel Guha, Mayee F. Chen, Trevor Chow, Ishan S. Khare, and Christopher Ré. 2024. Smoothie: Label free language model routing. In *Advances in Neural Information Processing Systems*, volume 37, pages 127645–127672. Curran Associates, Inc.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Jack Hessel, Ana Marasović, Jena D Hwang, Lillian Lee, Jeff Da, Rowan Zellers, Robert Mankoff, and Yejin Choi. 2022. Do androids laugh at electric sheep? humor" understanding" benchmarks from the new yorker caption contest. *arXiv preprint arXiv:2209.06293*.

Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Congchao Wang, Zifeng Wang, Alec Go, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, Aditya Krishna Menon, and Sanjiv Kumar. 2025. Universal model routing for efficient llm inference. *Preprint*, arXiv:2502.08773.

Mehran Kazemi, Hamidreza Alvari, Ankit Anand, Jialin Wu, Xi Chen, and Radu Soricut. 2023. Geomverse: A systematic evaluation of large models for geometric reasoning. *arXiv preprint arXiv:2312.12241*.

Mehran Kazemi, Bahare Fatemi, Hritik Bansal, John Palowitch, Chrysovalantis Anastasiou, Sanket Vaibhav Mehta, Lalit K. Jain, Virginia Aglietti, Disha Jindal, Peter Chen, Nishanth Dikkala, Gladys Tyen, Xin Liu, Uri Shalit, Silvia Chiappa, Kate Olszewska, Yi Tay, Vinh Q. Tran, Quoc V. Le, and Orhan Firat. 2025. Big-bench extra hard. *arXiv preprint arXiv:2502.19187*.

Mehran Kazemi, Quan Yuan, Deepti Bhatia, Najoung Kim, Xin Xu, Vaiva Imbrasaite, and Deepak Ramachandran. 2024. Boardgameqa: A dataset for natural language reasoning with contradictory information. *Advances in Neural Information Processing Systems*, 36.

Zaid Khan, Elias Stengel-Eskin, Jaemin Cho, and Mohit Bansal. 2025. Dataenvgym: Data generation agents in teacher environments with student feedback. In *International Conference on Learning Representations (ICLR)*.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*.

Hyunjae Kim, Hyeon Hwang, Jiwoo Lee, Sihyeon Park, Dain Kim, Taewhoo Lee, Chanwoong Yoon, Jiwoong Sohn, Donghee Choi, and Jaewoo Kang. 2024. Small language models learn enhanced reasoning skills from medical textbooks. *Preprint*, arXiv:2404.00376.

Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. 2024a. Common 7b language models already possess strong math capabilities. *Preprint*, arXiv:2403.04706.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Lample Guillaume, and Stanislas Polu. 2024b. Numinamath.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA. Association for Computational Linguistics.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On

LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082, Bangkok, Thailand. Association for Computational Linguistics.

M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Kang Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixin Deng, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, and 77 others. 2025. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *Preprint*, arXiv:2502.14739.

Aviya Maimon, Amir DN Cohen, Gal Vishne, Shauli Ravfogel, and Reut Tsarfaty. 2025. Iq test for llms: An evaluation framework for uncovering core skills in llms. *Preprint*, arXiv:2507.20208.

Mihai Nadăş, Laura Dioşan, and Andreea Tomescu. 2025. Synthetic data generation using large language models: Advances in text and code. *IEEE Access*, 13:134615–134633.

Allen Nie, Yuhui Zhang, Atharva Shailesh Amdekar, Chris Piech, Tatsunori B Hashimoto, and Tobias Gerstenberg. 2024. Moca: Measuring human-language model alignment on causal and moral judgment tasks. *Advances in Neural Information Processing Systems*, 36.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Kadous, and Ion Stoica. 2025. Routellm: Learning to route llms from preference data. In *International Conference on Representation Learning*, volume 2025, pages 34433–34448.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. 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*, pages 248–260. PMLR.

Josef Pichlmeier, Philipp Ross, and Andre Luckow. 2024. Performance characterization of expert router for scalable llm inference. *Preprint*, arXiv:2404.15153.

Lokesh R Sahana. 2025. Open-source large language models: A comprehensive survey. In *2025 International Conference on Machine Learning and Autonomous Systems (ICMLAS)*, pages 1096–1101.

Soham Shah and Kumar Shridhar. 2025. Select-then-route : Taxonomy guided routing for LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 425–441, Suzhou (China). Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Asterios Tsiourvas, Wei Sun, and Georgia Perakis. 2025. Causal llm routing: End-to-end regret minimization from observational data. *Preprint*, arXiv:2505.16037.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2024a. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pages 95266–95290. Curran Associates, Inc.

Kai Xiong, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. 2023. Examining inter-consistency of large language models collaboration: An in-depth analysis via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7572–7590, Singapore. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Longhui Yu, Weisen JIANG, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *International Conference on Representation Learning*, volume 2024, pages 45040–45061.

Shoubin Yu, Yue Zhang, Ziyang Wang, Jaehong Yoon, and Mohit Bansal. 2025. Mexa: Towards general multimodal reasoning with dynamic multi-expert aggregation. *arXiv preprint arXiv:2506.17113*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STar: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*.

Haozhen Zhang, Tao Feng, and Jiaxuan You. 2025a. Router-r1: Teaching llms multi-round routing and

aggregation via reinforcement learning. *Preprint*, arXiv:2506.09033.

Jifan Zhang, Lalit Jain, Yang Guo, Jiayi Chen, Kuan Lok Zhou, Siddharth Suresh, Andrew Wagenmaker, Scott Sievert, Timothy Rogers, Kevin Jamieson, and 1 others. 2024. Humor in ai: Massive scale crowd-sourced preferences and benchmarks for cartoon captioning. *arXiv preprint arXiv:2406.10522*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Yiqun Zhang, Hao Li, Chenxu Wang, Linyao Chen, Qiaosheng Zhang, Peng Ye, Shi Feng, Daling Wang, Zhen Wang, Xinrun Wang, Jia Xu, Lei Bai, Wanli Ouyang, and Shuyue Hu. 2025c. The avengers: A simple recipe for uniting smaller language models to challenge proprietary giants. *Preprint*, arXiv:2505.19797.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. *Preprint*, arXiv:2310.06117.

A Qualitative Example

To qualitatively examine the skill clusters learned during the clustering process, we visualize query embeddings and identified centroids from SuperGPQA using PCA in Figure 4. In the projection, queries from each region (grouped by color) correspond to broad domains such as science, engineering, and law – labels already available in SuperGPQA. The identified centroids, however, capture more fine-grained subtopics within these domains. For instance, legal queries separate into distinct clusters corresponding to contract & case law (red square marker) and constitutional law (orange triangle marker). These centroids demonstrate that CASCAL identifies subtle skill distinctions, capturing niche competencies across models.

B Note on AI Usage

We used AI tools for grammar correction and code completion.

C Further Experimental Details

We provide further implementation details for our experiments in this section. Section C.1 lists all models in each pool. Section C.2 outlines the features we used for our LLMRANK implementation.

Pool	Model	Params
POOL-LARGE	GPT-OSS (Agarwal et al., 2025)	120B
	LLaMA-3.3 (AI, 2024)	70B
	Qwen-3 (Yang et al., 2025)	32B
	GLM-4 (AI and KEG, 2024)	32B
	Exaone-4 (Bae et al., 2026)	32B
	Gemma-3 (Gemma Team et al., 2025)	27B
POOL-SMALL	Gemma-2 (Gemma Team et al., 2024)	9B
	GLM-4 (GLM Team et al., 2024)	9B
	Yi-1.5 (AI et al., 2025)	9B
	Qwen-3 (Yang et al., 2025)	8B
	Exaone-3.5 (An et al., 2026)	7.8B
	DeepSeek-Math (Shao et al., 2024)	7B

Table 4: All models used in routing experiments. We evaluate routing methods on two disjoint model pools grouped by parameter scale.

Section C.3 further describes our preprocessing steps for BBEH. Section C.4 provides additional context on our selection of generator models.

C.1 Model Pool Details

Table 4 lists all models in POOL-LARGE and POOL-SMALL.

C.2 LLMRANK Features

We implement the same neural architecture proposed in LLMRANK (Agrawal and Gupta, 2025). However, we use *natural language skills* as features. We further express skills in two ways, as a list of natural language skill descriptors, and as a single well-formed step-back prompt (Zheng et al., 2024). We used Figure 5 to obtain skills, and Figure 6 to get the step-back prompt. Thus, LLMRANK makes a routing decision conditioned on the embeddings of the queries, a list of skills, and a step-back prompt.

C.3 Dataset Details

To ease implementation and ensure simplicity in the evaluation pipeline, we filter out tasks in BigBench-Extra-Hard which cannot be easily formatted as multiple-choice questions. Below are the tasks that we include in BBEH: Boardgame QA (Kazemi et al., 2024), Boolean Expressions, Causal Understanding (Nie et al., 2024; Kıcıman et al., 2023), Disambiguation QA, Geometric Shapes (Kazemi et al., 2023), Hyperbaton, Movie Recommendation, Shuffled Objects, NYCC (Hessel et al., 2022; Zhang et al., 2024).

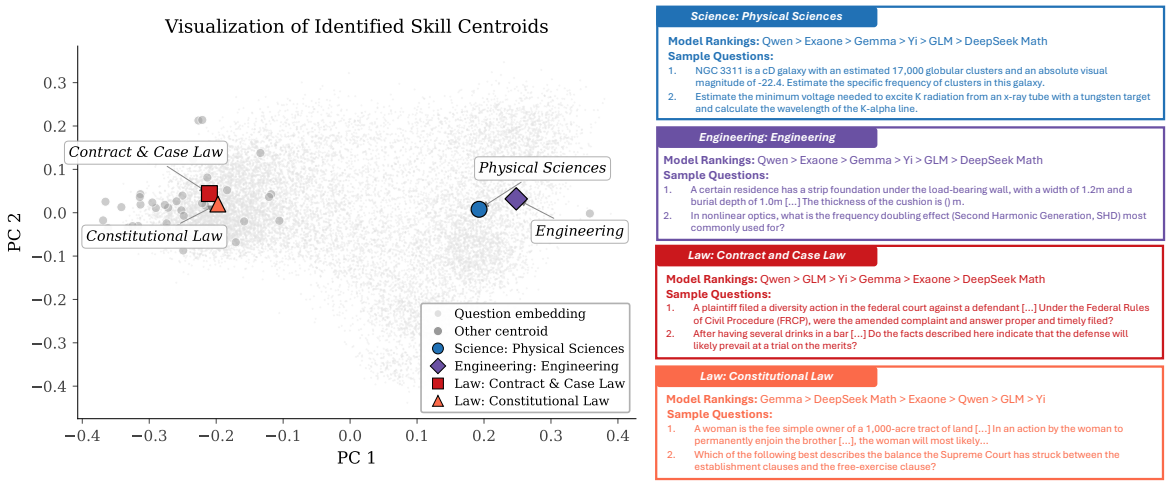

Figure 4: PCA projection of all training questions (gray dots) and learned centroids (circles). Four centroids are highlighted to illustrate sub-topic specialization across disciplines.

C.4 Notes on Generator Selection

In our experiments, we select Qwen3-32B and Exaone-3.5-7.8B-Instruct as the two open-weight generators. We note that these models are selected due to logistical considerations rather than benchmark performance. Indeed, neither model is the strongest LLM within POOL-LARGE or POOL-SMALL (Appendix D.3). Our design choice aligns with potential downstream applications of RGD. While standard benchmarks can serve as a rough proxy for generator strength, this knowledge cannot be assumed for novel or niche tasks. In such cases, practitioners may have to use a readily available model to generate routing data without knowing whether it is strong or weak relative to the pool – which is precisely the problem that CASCAL is tackling. Furthermore, we show that the quality of generated data can be improved through filtering, even when starting from a weak generator (Section 4.3.3), thereby providing a practical mitigation that does not require access to a stronger generator.

D Additional Results

D.1 Results on MATH

Method	Pool-Large	Pool-Small
Oracle	98.8	89.8
Random 1 model	90.7	58.0
Random 3 model vote	94.0	66.1
CASCAL Top 1	92.9	83.0
CASCAL Top 3	95.4	80.4

Table 5: Performance of CASCAL on the MATH dataset across POOL-LARGE and POOL-SMALL.

While our evaluated datasets (MMLU-Pro, MedMCQA, SuperGPQA) are primarily comprised of multiple-choice questions, RGD is applicable to a wide range of tasks. The sole requirement is that the model responses can be binned into a certain number of categories, i.e., that we have a means to measure equivalence between two responses. To demonstrate this empirically, we report additional experiments on an open-ended math task. To match the dataset size used in our main experiments, we randomly sample 5k questions from the train split of the MATH dataset (Hendrycks et al., 2021) and evaluate on 5k questions from the test split. Table 5 shows CASCAL remains effective for non-MCQ datasets in both POOL-LARGE and POOL-SMALL.

D.2 Domain-Specific Generator

To examine whether domain-specific generators outperform general ones, we additionally experiment with the routing data generated using Meerkat-8B—a Llama-based LLM finetuned for medical tasks (Kim et al., 2024)—and Exaone-3.5-7.8B-Instruct (the weakest of the 3 generators tested in the paper) for MedMCQA. Using these two models, we separately generate routing data and compare downstream RGD performance on MedMCQA's test set. We observe performance improvements on POOL-SMALL but not POOL-LARGE, indicating that domain-specific generators outperform size-matched generalist models, but not necessarily larger models that are stronger overall.

D.3 RGD Accuracies

Full RGD accuracies, per pool and dataset, are reported in Table 7, Table 8, Table 9, and Table 10.

Method	MedMCQA (POOL-LARGE)	MedMCQA (POOL-SMALL)
Generator: Exaone	—	—
Cascal Top 1	81.6	60.7
Cascal Top 3	84.1	65.3
Generator: Meerkat	—	—
Cascal Top 1	81.6	65.0
Cascal Top 3	83.5	67.6

Table 6: Performance of CASCAL under general and domain-specific generator models on MedMCQA across POOL-LARGE and POOL-SMALL.

Oracle represents the upper bound, selecting the best-performing model for each individual query. Note that this is treated as an absolute upper bound oracle, since the performance of models on test queries is not known *a priori*: it is the highest number that could be obtained from the model pool, assuming absolutely perfect routing.

D.4 Single-model Accuracies

Single-model accuracies across benchmarks are reported in Table 11.

D.5 RGD Generator Alignment with Gemini-3-Flash

Table 12 shows alignment between generators and Gemini-3-Flash.

D.6 Robustness to Initial Description

We also examine whether our pipeline is robust to the quality of the initial task descriptions. To do so, we generate multiple task descriptions for each task and compare the downstream performance of routers trained on questions derived from each description. For each task, using the current task description as a reference, we use Gemini-2.5-Flash to further generate a more concise description and a one-sentence description. We then re-run experiments on SuperGPQA and BBEH. We show that CASCAL's performance is relatively stable across description granularities (Table 13).

E Licenses

We publicly release our code in the supplementary. We provide the following links to the standard licenses for the datasets, code, and models used in this project.

- **MMLU-Pro:** Apache License v2.0.
- **SuperGPQA:** ODC-by.
- **MedMCQA:** MIT.
- **BigBench Extra Hard:** Apache License v2.0.
- **MATH:** MIT.
- **GPT-OSS-120B:** Apache License v2.0.
- **GLM-4-32B-0414:** MIT.
- **Llama-3.3-70B-Instruct:** Llama 3.3 Community License Agreement.
- **Exaone-4-32B:** EXAONE AI Model License Agreement 1.2.
- **Gemma-2-9B-It, Gemma-3-27B:** Gemma Terms of Use.
- **Exaone-3.5-7.8B-Instruct:** EXAONE AI Model License Agreement 1.1.
- **Deepseek-Math-7B-Instruct:** DeepSeek License.
- **GLM-4-9B-Chat:** The glm-4-9b License.
- **Yi-1.5-9B-Chat-16K:** Apache License v2.0.
- **Qwen3-8B, Qwen3-Embedding-8B, Qwen3-32B:** Apache License v2.0.
- **Gemini-2.5-Flash, Gemini-3-Flash:** Gemini APIs Terms of Service and Gemini API Additional Terms of Service.
- **Meerkat-8B:** Creative Commons Attribution Non Commercial 4.0 International.

METHOD	MMLU-PRO	SUPERGPQA	MEDMCQA	BBEH	AVG.
			Pool Large		
ORACLE	91.5	77.6	96.3	66.4	83.0
TOP-1 / TOP-3	79.4 / 80.9	54.4 / 54.7	92.7 / 88.5	34.0 / 34.3	65.1 / 64.6
Query-answer Routers					
CASCAL-GT (TOP-1/3)	78.0 / 80.6	54.7 / 54.7	92.7 / 87.0	37.4 / 35.5	65.7 / 64.5
AVENGERS	78.0	53.7	92.7	36.5	63.4
LLMRANK	78.5	53.6	92.7	37.5	65.6
Query-only Routers					
SMOOTHIE (TOP-1/3)	74.8 / 77.6	43.5 / 46.6	78.0 / 80.8	32.0 / 32.0	57.1 / 59.3
CASCAL (TOP-1/3)	78.9 / 81.0	53.4 / 53.8	85.7 / 86.5	30.7 / 33.0	62.2 / 63.6
			Pool Small		
ORACLE	84.4	67.8	91.2	56.7	75.0
TOP-1 / TOP-3	66.2 / 65.7	37.3 / 35.7	71.0 / 70.2	25.1 / 25.0	49.9 / 49.2
Query-answer Routers					
CASCAL-GT (TOP-1/3)	66.0 / 63.0	36.9 / 31.0	71.0 / 68.7	25.0 / 26.2	49.7 / 47.3
AVENGERS	65.9	37.1	70.6	27.3	50.2
LLMRANK	65.0	31.8	71.0	27.2	48.8
Query-only Routers					
SMOOTHIE (TOP-1/3)	55.8 / 60.9	28.6 / 30.5	67.9 / 69.1	21.1 / 23.1	43.4 / 45.9
CASCAL (TOP-1/3)	62.9 / 62.8	32.4 / 30.8	65.5 / 68.4	25.4 / 23.5	46.6 / 46.4

Table 7: Performance of different routers when trained using **validation** data across model pools. Results shown as Top-1 / Top-3 where applicable, where top-3 represents consensus-voting with the top 3 ranked models in CASCAL-variants, and majority voting in SMOOTHIE-TRAIN.

METHOD	MMLU-PRO	SUPERGPQA	MEDMCQA	BBEH	AVG.
			Pool Large		
ORACLE	91.5	77.6	96.3	66.4	83.0
TOP-1 / TOP-3	79.4 / 80.9	54.4 / 54.7	92.7 / 88.5	34.0 / 34.3	65.1 / 64.6
Query-answer Routers					
CASCAL-GT (TOP-1/3)	78.8 / 79.6	51.3 / 53.4	84.7 / 86.0	33.1 / 32.6	62.0 / 62.9
AVENGERS	78.6	51.7	84.8	32.0	61.8
LLMRANK	76.7	52.6	81.2	34.5	61.3
Query-only Routers					
SMOOTHIE (TOP-1/3)	75.1 / 78.6	44.5 / 48.5	77.7 / 80.3	32.1 / 32.0	57.4 / 59.9
CASCAL (TOP-1/3)	76.7 / 80.3	42.9 / 52.3	83.2 / 85.9	30.5 / 33.9	58.3 / 63.1
			Pool Small		
ORACLE	84.4	67.8	91.2	56.7	75.0
TOP-1 / TOP-3	66.2 / 65.7	37.3 / 35.7	71.0 / 70.2	25.1 / 25.0	49.9 / 49.2
Query-answer Routers					
CASCAL-GT (TOP-1/3)	65.6 / 61.6	36.3 / 31.4	68.4 / 68.5	25.0 / 24.3	48.8 / 46.5
AVENGERS	64.9	36.3	69.2	24.3	48.7
LLMRANK	65.0	36.4	67.5	27.3	49.1
Query-only Routers					
SMOOTHIE (TOP-1/3)	60.6 / 62.2	31.8 / 32.2	62.9 / 67.0	22.1 / 23.1	44.4 / 46.1
CASCAL (TOP-1/3)	64.8 / 62.2	36.2 / 31.7	65.9 / 68.0	23.8 / 24.1	47.7 / 46.5

Table 8: Performance of different routers when trained using **Gemini-2.5-Flash** generated queries across model pools. Results shown as Top-1 / Top-3 where applicable, where top-3 represents consensus-voting with the top 3 ranked models in CASCAL-variants, and majority voting in SMOOTHIE-TRAIN.

METHOD	MMLU-PRO	SUPERGPQA	MEDMCQA	BBEH	AVG.
			Pool Large		
ORACLE	91.5	77.6	96.3	66.4	83.0
TOP-1 / TOP-3	79.4 / 80.9	54.4 / 54.7	92.7 / 88.5	34.0 / 34.3	65.1 / 64.6
Query-answer Routers					
CASCAL-GT (TOP-1/3)	75.9 / 78.5	40.5 / 47.9	81.1 / 84.1	31.5 / 32.6	57.3 / 60.8
AVENGERS	76.5	45.6	82.2	32.1	59.1
LLMRANK	75.1	44.9	82.6	33.7	59.1
Query-only Routers					
SMOOTHIE (TOP-1/3)	74.2 / 78.3	43.8 / 47.9	77.2 / 80.6	30.7 / 29.8	56.5 / 59.2
CASCAL (TOP-1/3)	74.2 / 78.7	43.9 / 49.3	82.7 / 84.1	32.6 / 32.1	58.4 / 61.1
			Pool Small		
ORACLE	84.4	67.8	91.2	56.7	75.0
TOP-1 / TOP-3	66.2 / 65.7	37.3 / 35.7	71.0 / 70.2	25.1 / 25.0	49.9 / 49.2
Query-answer Routers					
CASCAL-GT (TOP-1/3)	61.1 / 61.2	34.5 / 31.0	67.8 / 68.1	25.3 / 23.5	47.2 / 46.0
AVENGERS	63.0	29.0	64.7	25.2	45.5
LLMRANK	60.2	34.5	70.5	24.3	47.4
Query-only Routers					
SMOOTHIE (TOP-1/3)	56.3 / 60.5	27.5 / 29.3	63.9 / 67.5	22.8 / 22.7	42.6 / 45.0
CASCAL (TOP-1/3)	61.0 / 61.0	35.4 / 30.8	64.5 / 67.4	24.1 / 23.3	46.3 / 45.6

Table 9: Performance of different routers when trained using **Qwen3-32B** generated queries across model pools. Results shown as Top-1 / Top-3 where applicable, where top-3 represents consensus-voting with the top 3 ranked models in CASCAL-variants, and majority voting in SMOOTHIE-TRAIN.

METHOD	MMLU-PRO	SUPERGPQA	MEDMCQA	BBEH	AVG.
			Pool Large		
ORACLE	91.5	77.6	96.3	66.4	83.0
TOP-1 / TOP-3	79.4 / 80.9	54.4 / 54.7	92.7 / 88.5	34.0 / 34.3	65.1 / 64.6
Query-answer Routers					
CASCAL-GT (TOP-1/3)	71.9 / 77.5	39.2 / 44.5	80.7 / 84.1	34.3 / 32.6	56.5 / 59.7
AVENGERS	71.9	49.4	86.6	27.5	58.9
LLMRANK	73.8	40.8	79.2	34.5	57.1
Query-only Routers					
SMOOTHIE (TOP-1/3)	74.8 / 78.2	43.6 / 48.1	76.1 / 79.6	31.6 / 31.6	56.5 / 59.4
CASCAL (TOP-1/3)	74.6 / 78.7	42.6 / 49.2	81.6 / 84.1	32.1 / 32.3	57.7 / 61.1
			Pool Small		
ORACLE	84.4	67.8	91.2	56.7	75.0
TOP-1 / TOP-3	66.2 / 65.7	37.3 / 35.7	71.0 / 70.2	25.1 / 25.0	49.9 / 49.2
Query-answer Routers					
CASCAL-GT (TOP-1/3)	52.9 / 58.9	25.6 / 30.8	56.8 / 65.3	21.4 / 23.4	39.2 / 44.6
AVENGERS	54.7	25.9	59.8	22.4	40.7
LLMRANK	53.9	28.8	57.0	24.4	41.0
Query-only Routers					
SMOOTHIE (TOP-1/3)	54.3 / 57.8	25.7 / 27.6	59.7 / 65.3	21.7 / 22.7	40.4 / 43.4
CASCAL (TOP-1/3)	57.1 / 61.7	32.1 / 31.2	60.8 / 65.3	20.6 / 22.8	42.7 / 45.3

Table 10: Performance of different routers when trained using **Exaone-3.5-7.8B-Instruct** generated queries across model pools. Results shown as Top-1 / Top-3 where applicable, where top-3 represents consensus-voting with the top 3 ranked models in CASCAL-variants, and majority voting in SMOOTHIE-TRAIN.

MODELS	MMLU-PRO	SUPERGPQA	MEDMCQA	BBEH	AVG.
Large Models					
QWEN3-32B	78.4	54.4	78.9	27.5	59.8
GPT-OSS-120B	79.4	52.0	81.4	34.0	61.7
GLM-4-32B-0414	74.6	43.3	78.6	31.1	56.9
LLAMA-3.3-70B-INSTRUCT	70.6	36.3	92.7	33.3	58.3
GEMMA-3-27B	69.0	36.2	73.4	31.7	52.6
EXAONE-4-32B	75.1	41.5	71.7	30.2	54.6
Small Models					
QWEN3-8B	66.2	37.3	71.0	25.1	49.9
EXAONE-3.5-7.8B-INSTRUCT	55.3	25.8	56.9	24.1	40.5
GEMMA-2-9B-IT	52.4	24.6	64.6	21.1	40.7
GLM-4-9B-CHAT	46.1	21.5	59.8	20.1	36.9
YI-1.5-9B-CHAT-16K	43.7	19.4	52.8	18.3	33.6
DEEPSEEK-MATH-7B-INSTRUCT	28.5	17.4	36.2	15.7	24.5

Table 11: Single model accuracy on the test split of all evaluated benchmarks.

	RGD Scenario				
	MMLU-Pro	**SuperGPQA**	**MedMCQA**	**BBEH**	**Average**
Gemini-3-Flash	0.86	0.68	0.97	0.49	0.74
Generator Model					
Exaone-3.5-7.8B-Instruct	65.6	75.8	75.4	44.1	65.2
Qwen3-32B	75.1	84.0	79.0	61.4	74.9

Table 12: Top row (gray): Gemini-3-Flash accuracy on test sets, included for reference. Bottom three rows: Generator label quality measured as agreement with Gemini-3-Flash answers. The stronger generator (Qwen3-32B) has consistently higher agreement.

GENERATOR	DESCRIPTION	SUPERGPQA	BBEH
Pool Large			
GEMINI-GENERATED	DEFAULT	42.9 / 52.3	30.5 / 33.9
	SHORT	45.7 / 52.4	32.7 / 33.1
	SHORTEST	53.4 / 53.2	33.7 / 32.4
QWEN-GENERATED	DEFAULT	43.9 / 49.3	32.6 / 32.1
	SHORT	45.5 / 53.0	32.1 / 34.3
	SHORTEST	43.5 / 50.2	31.1 / 32.1
EXAONE-GENERATED	DEFAULT	42.6 / 49.2	32.1 / 32.3
	SHORT	42.3 / 45.5	31.0 / 31.5
	SHORTEST	42.9 / 46.9	31.2 / 32.8
Pool Small			
GEMINI-2.5-FLASH	DEFAULT	36.2 / 31.7	23.8 / 24.1
	SHORT	36.2 / 31.6	26.9 / 24.6
	SHORTEST	36.1 / 30.6	26.6 / 23.5
QWEN3-32B	DEFAULT	35.4 / 30.8	24.1 / 23.3
	SHORT	32.9 / 31.7	26.5 / 25.1
	SHORTEST	33.4 / 31.4	24.1 / 23.1
EXAONE-3.5-7.8B-INSTRUCT	DEFAULT	32.1 / 31.2	20.6 / 22.8
	SHORT	30.1 / 30.4	25.4 / 25.0
	SHORTEST	32.0 / 31.4	23.0 / 25.1

Table 13: Pipeline robustness to initial task descriptions. Results are shown as Top-1 / Top-3 for CASCAL using routing data generated from task descriptions of varying granularity. Across generators and datasets, performance is relatively stable under changes in description length.

F Prompts

This section includes all prompts used. Section F.2 outlines the task descriptions given to the generator.

F.1 LLMRank Implementation Prompts

F.2 Task Descriptions

We show the task descriptions used for MMLU-Pro (Table 14), SuperGPQA (Table 15), MedMCQA (Table 16, Table 17), and BBEH (Table 18, Table 19).

F.3 Model Response Generation

Figure 7 shows the prompt used to extract model responses for all queries. Figure 9 displays the prompt given to generators to generate RGD queries. Figure 8 shows the prompt used to generate task descriptions using Gemini-2.5-Flash given 5 seed example questions.

User Prompt

Question: <question>
What are the core knowledge, subjects or skills needed to solve this problem? List 2-5 keywords separated in comma. Example keywords: psychology, virology, behavioral theory, microbiology, diplomacy, political science, property law, finance, business. Give ONLY the keywords, no other words or explanation. Follow this format: Keywords: <keyword1>, <keyword2>...

Figure 5: Prompt used to extract skills from questions.

User Prompt

Question: <question>
Answer: <answer>
Do not answer the question directly. Write a brief section describing the high-level ideas of the problem and the skills or knowledge required to solve it. Remember to keep your response concise. It should touch on the main topics needed to answer the question, but not go into too much detail.

Figure 6: Prompt used to generate step-back prompts.

MMLU-Pro Task Descriptions

Biology	These questions describe conceptual and explanatory biology problems—often open-ended—that require understanding and articulating core mechanisms in genetics, evolution, plant and animal physiology, and endocrinology.
History	Analytical reasoning tasks requiring interpretation of historical documents and scientific evidence to infer causes, processes, and timelines.
Chemistry	Quantitative physical chemistry and thermodynamics problems involving gas laws, electrochemical cell equations, and entropy calculations.
Psychology	Behavioral science and applied statistics combining perception, learning theory, and organizational behavior concepts.
Economics	Problems testing macroeconomic measurement, short-run fluctuations, price determination, and profit-maximizing behavior.
Math	Quantitative exercises ranging from basic arithmetic to calculus and linear algebra requiring explicit numerical calculations.
Engineering	Advanced calculation-based problems applying core physical laws and analytical formulas to compute precise results.
Physics	Quantitative problems using fundamental principles from optics, mechanics, electromagnetism, and nuclear physics.
Law	Hypotheticals requiring the application of legal doctrines from constitutional, property, and criminal law to fact patterns.
Computer Science	Foundational concepts spanning algorithms, operating systems, and machine learning, requiring conceptual evaluation of system properties.

Table 14: MMLU-Pro Task Descriptions

SUPER-GPQA Task Descriptions

Management	These questions describe conceptual and policy-oriented social science problems focused on public administration, health policy, human capital theory, social welfare systems, and international marketing strategy, requiring classification, principle identification, and applied theoretical understanding rather than numerical computation.
Literature and Arts	These questions describe arts and humanities knowledge queries focused on music theory, literature symbolism, composers and albums, classical drama, and modern art exhibitions, requiring factual recall and interpretive understanding of artistic works and creators.
Military Science	These questions describe fact-based military science and defense studies inquiries focused on the history of military thought, senior command appointments, key revolutionary military doctrines, organizational terminology, and leadership succession in national defense systems.
Agronomy	These questions describe fact-based and applied problems in animal science, botany, forestry, and veterinary medicine that require biological classification, ecological data recall, anatomical knowledge, and clinical diagnostic reasoning for livestock diseases.
Science	These questions describe advanced quantitative science and mathematics problems—spanning astrophysics, physical chemistry equilibrium, differential equations/curve construction, real analysis of function sequences, and planetary dynamics—that require applying formal models and formulas to idealized systems in order to derive specific numerical values or rigorously characterize mathematical behavior.
Economics	These questions describe quantitative and analytical economics/business problems that combine logical argument evaluation, macroeconomic multipliers, Marxist political economy concepts, cost/benefit evaluation, and discounted cash-flow investment appraisal.
Philosophy	These questions describe comparative and interpretive philosophy, religion, and intellectual history inquiries that analyze major Western and Chinese thinkers' views on ethics, causality, personhood, aesthetics, and religious movements, requiring conceptual understanding of philosophical doctrines and their historical contexts.
Law	These questions describe legal and political knowledge problems that require understanding of military law, civil procedure and jurisdiction, conflict-of-laws principles, and twentieth-century East Asian political history and strategy, combining doctrinal legal analysis with historical–political interpretation.
Sociology	These questions describe fact-based and conceptual sociology and social history inquiries that combine cultural anthropology, social theory, civic movements, and institutional history, requiring precise recall of dates, organizations, and definitions of core sociological concepts.
History	These questions describe fact-based historical and political knowledge queries that require precise recall of dates, official appointments, territorial definitions, foundational events, and ideological interpretations from modern and premodern history.
Education	These questions describe education and sports science theory problems that focus on student character development, exercise physiology, physical education teaching methodology, training control principles, and learner-centered instructional design, requiring conceptual understanding of how physiological factors and pedagogical strategies influence student learning and development.
Medicine	These questions describe applied medical and biomedical science problems that require integrating physiology, pathology, pharmacokinetics, and immunohematology to identify disease mechanisms, interpret clinical signs, and explain diagnostic or treatment-related phenomena.
Engineering	These questions describe technical and applied science problems spanning marine instrumentation, mechanical engineering materials, medical aesthetics technology, information theory coding efficiency, and strategic geophysical sensing, requiring domain-specific technical knowledge and practical reasoning.

Table 15: SuperGPQA Task Descriptions

MedMCQA Task Descriptions (Part 1)

Pathology	These questions describe applied medical pathology and genetic risk assessment problems that require understanding of endocrine dysfunction in systemic disease, characteristic histopathological tissue injury patterns, premalignant lesion progression, Mendelian inheritance probabilities, and the fundamental pathological changes associated with chronic metabolic disorders.
Ophthalmology	These questions describe applied ophthalmology and orbital anatomy knowledge checks that test identification of extraocular muscle anatomy, patterns of orbital trauma, infectious causes of pediatric eye disease, vascular causes of proptosis, and precise anatomical relationships within the lacrimal drainage system.
Gynaecology & Obstetrics	These questions describe applied obstetrics and reproductive physiology knowledge checks that focus on abnormal pregnancy types, statistical laws of multiple gestation, placental disorders, timing of ovulation, and the appropriate gestational age for prenatal diagnostic procedures.
Anaesthesia	These questions describe applied anesthesiology and perioperative medicine problems that test understanding of anesthesia physiology, emergency induction drug selection, anesthetic equipment function, neuromuscular pharmacology, and respiratory support principles such as CPAP.
Forensic Medicine	These questions describe applied community medicine, forensic medicine, and clinical toxicology problems that require understanding of safe substance use limits, postmortem changes, principles of enhanced drug elimination, identification of common environmental poisons, and recognition of characteristic clinical signs of heavy-metal toxicity.
Psychiatry	These questions describe clinical psychiatry, neurology, pharmacology, and behavioral psychology problems that assess recognition of substance-related emergencies, rational drug selection in children, principles of learning and reinforcement, EEG interpretation, and mood disorder diagnosis.
Unknown	These questions describe integrated medical science knowledge checks spanning endocrinology, infectious disease diagnostics, pediatric emergency medicine, neuroanatomy of sensory pathways, and biostatistics, requiring applied understanding of disease mechanisms, clinical presentation and management, physiological pathways, and research methodology.
Microbiology	These questions describe core medical microbiology and immunology knowledge checks that focus on immune protein synthesis, host-parasite relationships, bacterial virulence factors, laboratory identification of pathogens based on culture characteristics, and fundamental biological classification of fungi.
Medicine	These questions describe integrated clinical medicine knowledge checks that require applying anatomy, cardiology diagnostics, immunology, hematology, and electrocardiography principles to identify disease mechanisms, diagnostic tests, and characteristic clinical findings.
Radiology	These questions describe diagnostic radiology and medical imaging knowledge problems that assess understanding of MRI and CT principles, radiographic anatomy and pathology correlations, systemic disease indicators on imaging, and safe imaging practices in pregnancy.

Table 16: MedMCQA Task Descriptions (Part 1)

MedMCQA Task Descriptions (Part 2)

Biochemistry	These questions describe core biochemistry and molecular biology knowledge checks that require understanding of lipid metabolism, RNA processing, ketone body synthesis, metabolic pathway integration, and hormone storage physiology.
Dental	These questions describe foundational dental and oral pathology knowledge checks that assess understanding of oral histology, mucocutaneous disease identification, pediatric orthodontic appliances, diagnostic criteria, and periodontal tissue healing mechanisms.
Pediatrics	These questions describe core pediatrics and neonatal medicine knowledge checks that focus on infant and child physiology, developmental milestones, neonatal pathology, congenital cardiovascular adaptation, toxic ingestion emergencies, and bilirubin metabolism-related risks, requiring applied clinical understanding of normal development and common pediatric disorders.
Physiology	These questions describe core human physiology and biomedical science knowledge checks that focus on hormone secretion, cardiac electrophysiology, intracellular second-messenger systems, membrane biophysics, and gastrointestinal motility patterns, requiring integrated understanding of cellular signaling and organ system function.
Skin	These questions describe clinical dermatology multiple-choice problems that assess recognition of skin disease patterns, drug-related complications, characteristic pathological phenomena, pregnancy-related treatment considerations, and the mechanisms and management of pigmentary and pustular dermatoses.
Pharmacology	These questions describe applied clinical pharmacology and emergency medicine knowledge checks that focus on drug mechanisms of action, receptor physiology, antidote use in poisoning, obstetric therapeutics, dermatologic treatments, and antibiotic modes of action for rational clinical decision-making.
Social & Preventive Medicine	These questions describe applied epidemiology, infectious disease control, and biostatistics problems that focus on disease surveillance, transmission dynamics, risk measurement, standardized treatment regimens, and interpretation of temporal patterns in public health data.
ENT	These questions describe applied ENT and neuroanatomy knowledge problems that require understanding of cranial nerve injury effects, auditory and visual sensory physiology, tympanic membrane histology, cholesteatoma pathology, and characteristic fracture patterns of the facial skeleton.
Surgery	These questions describe applied clinical medicine and surgery knowledge checks that focus on cancer patterns, anatomical risk during operative procedures, characteristic diagnostic signs, common causes of acute surgical emergencies, and evidence-based hormonal therapy selection.
Orthopaedics	These questions describe orthopedic clinical knowledge problems that test diagnosis, deformity identification, fracture management principles, bone infection etiology, and classic fracture-dislocation patterns through applied medical reasoning.
Anatomy	These questions describe foundational human anatomy knowledge checks that test structural identification, embryological origin of muscles, composition of superficial tissues, and key anatomical relationships between organs and neurovascular structures.

Table 17: MedMCQA Task Descriptions (Part 2)

BBEH Task Descriptions (Part 1)

Geometric Shapes	This question type tests planar and computational geometry reasoning from low-level graphical data: the student is given an SVG path consisting of M/L coordinate pairs that define multiple line segments in the plane, must mentally reconstruct the composite drawing formed by those segments, account for collinearity where multiple segments form a single side, and then identify which polygonal shapes are present from a fixed taxonomy (triangles, rectangles, squares, trapezoids, parallelograms, and various classes of pentagons and hexagons), using geometric properties such as number of sides, parallelism, right angles, regular vs irregular structure, and convexity vs concavity.
Movie Recommendation	These problems are all the same general type: for each question, you are (implicitly) given some candidate options, where each option is a set or pair/list of movies, and your task is to decide which option contains movies that are most similar with respect to how a group of people will respond to them—that is, which option groups together movies that are likely to be liked or disliked by (roughly) the same subset of people; solving each item requires comparing, across the options, the predicted preference patterns for the movies in that option (e.g., genre overlap, tone, target audience, prior ratings or response profiles) and then selecting the option whose movies have the highest expected alignment in group-level like/dislike outcomes.
Boolean Expressions	These problems are classic self-referential logic puzzles of the "exactly one statement is true" form: for each question you are given five candidate expressions (not shown here), and you must determine which single one can consistently be true under the global constraint that the other four are false; solving them requires systematically exploring the logical relations among the five expressions in each set (often involving mutual reference, counting claims like "exactly two of these are true/false," or structural dependencies), identifying contradictions that arise if you assume each candidate is the true one in turn, and finding the unique assignment of truth values that satisfies the constraint for each question independently.
Shuffled Objects	This question tests long-horizon state tracking and update reasoning in a dynamic system: the student is given an initial configuration of entities and their associated states (such as positions, partners, gifts, or objects), followed by an extremely long sequence of pairwise swap actions interspersed with null events and irrelevant discussions, and must deterministically simulate the effects of these operations to determine the final state of a queried variable. Correctly solving it requires precise bookkeeping of swaps, recognizing that repeated swaps undo previous ones, ignoring non-state-changing actions, maintaining consistency across hundreds of transitions, and extracting the final ownership, location, or partner of a specified individual.

Table 18: BBEH Task Descriptions (Part 1)

User Prompt

<Question Text>
Choices:
(A) <Option A>
(B) <Option B>
(C) <Option C>
(D) <Option D>
Think through the problem and provide your step-by-step reasoning. After that, if the question is a multiple choice problem, print 'The answer is (X)', where X is the answer choice (one capital letter), at the end of your response. If the question is a calculation question that has a numerical output, print 'The answer is (X.X)', where X.X is a float representing the result of the calculation. If the question requires you to output a list of objects, print 'The answer is (X Y Z ...)' where X, Y, and Z represent an object in the list, delimited by a single space.

Figure 7: User prompt used to generate model responses to all queries.

BBEH Task Descriptions (Part 2)

NYCC	This question tests pragmatic and stylistic judgment in evaluating humor: given short descriptions of New Yorker-style cartoons and multiple candidate captions (not shown here), the student must imagine the scene, infer the likely social context and character attitudes, and then choose the caption whose wording, tone, and perspective produce the strongest comic effect, taking into account incongruity, timing, voice, and how well the caption fits the visual details and implied narrative.
Causal Understanding	These questions explore causal attribution and normative judgment, specifically investigating how human intuition differentiates between physical necessity, logical sufficiency, and moral responsibility. The task requires the student to determine whether an agent is perceived as the cause of an outcome based on whether their actions violated a formal rule or moral norm, even in cases of causal overdetermination. While the second task category focuses on the mechanical bookkeeping of state tracking through long sequences of swaps, this category focuses on the psychological and logical evaluation of accountability within social or physical systems.
Hyperbaton	Question tests understanding and induction of adjective ordering constraints in a specific artificial variant of English: the student is given a large list of noun phrases whose adjective order is stipulated to be well-formed in that variant, must infer the underlying hierarchy or template governing semantic classes of modifiers (e.g., evaluative, size, age, color, material, nationality, activity-related/participial, shape, etc.), and then apply the inferred ordering rules to a new set of candidate noun phrases (Options A-J) to decide which have a grammatical adjective sequence in that system, including cases with multiple adjectives, mixed semantic types, and interaction of lexicalized participles with more canonical adjectives.
Boardgame QA	This question tests rule-based nonmonotonic logical reasoning in a toy narrative "boardgame" domain: the student is given an initial set of facts about many agents (animals) plus a large collection of conditional rules, some of which are defeasible and explicitly ordered by preferences, and must compute whether a target query (e.g., "does X do Y?") is provably true, provably false, or remains undecided. Solving it requires forward-chaining through the rules, tracking derived literals, handling multi-premise antecedents, reasoning about existence statements ("there exists an animal that..."), and crucially resolving conflicts when different rules support a proposition and its negation by respecting the given priority relation over rules, thereby determining whether the queried statement is classified as proved, disproved, or unknown.
Disambiguation QA	This question tests discourse-level pronoun resolution and coreference reasoning in natural language: the student is given short multi-clause narratives containing multiple pronouns whose antecedents may be explicit, implicit, or genuinely ambiguous, and must determine which entities each pronoun refers to by using grammatical cues (gender, number, syntactic role), semantic plausibility, discourse coherence, and pragmatic context, including cases with conflicting gender stereotypes, shifting speakers, plural references, and long-distance dependencies across sentences.

Table 19: BBEH Task Descriptions (Part 2)

User Prompt

<Question 1 text>
<Question 2 text>
<Question 3 text>
<Question 4 text>
<Question 5 text>
I am working on a project that involves categorizing different types of questions. I will give you 5 sample questions, and then your task is to briefly describe the type/subject of the question or the task. The idea is that I can give this to another instructor, who can then use your description to come up with additional questions in the same domain.

Figure 8: User prompt used to generate task description using Gemini-2.5-Flash.

User Prompt

You are an intelligent teaching assistant. Your current task is to generate questions for a multiple-choice exam. You will be given a description of the question category. Based on the description, generate one detailed **advanced graduate-level** question on similar topics.
Constraints: (1) Regardless of the number of choices in the input, ensure every generated question has exactly 4 choices (A, B, C, D). (2) Do NOT use JSON. Use the custom "Tagged Block" format defined below.
The Tagged Block Format: Use the following tags to structure your response. Content can span multiple lines.

[QUESTION] <Write the question text here. Use LaTeX $...$ for math.>
[OPTION A] <Text for Option A>
[OPTION B] <Text for Option B>
[OPTION C] <Text for Option C>
[OPTION D] <Text for Option D>
[ANSWER] <Single letter A, B, C, or D>

Example Output:

[QUESTION] Calculate the limit of $f(x)$ as $x \to \infty$.
[OPTION A] 0
[OPTION B] 1
[OPTION C] infinity
[OPTION D] Undefined
[ANSWER] B

Figure 9: User prompt used to prompt generators.