# RouterDC: Query-Based Router by Dual Contrastive Learning for Assembling Large Language Models

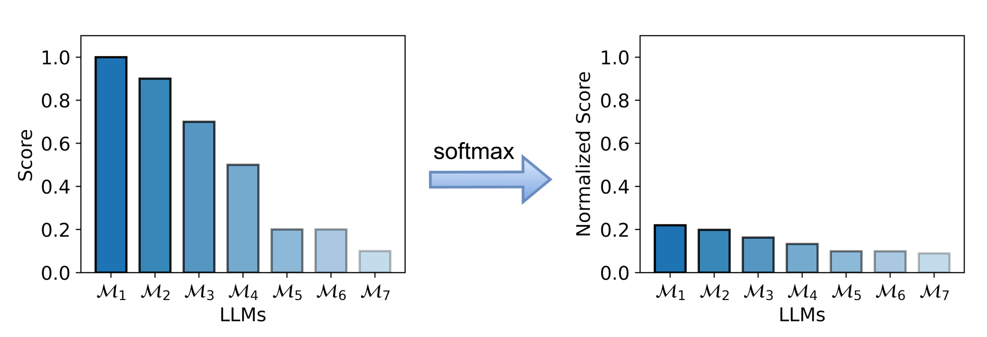
Shuhao Chen[1,*], Weisen Jiang[1,2,*], Baijiong Lin[3], James T. Kwok[2], Yu Zhang[1,†]

[1]Southern University of Science and Technology [2]The Hong Kong University of Science and Technology
[3]The Hong Kong University of Science and Technology (Guangzhou)

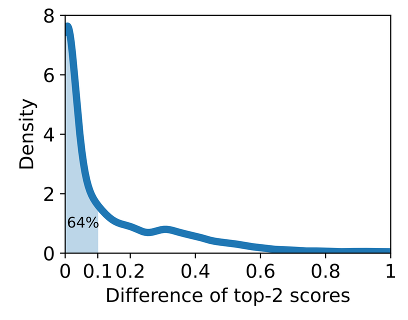
PAPER CODE

## Background

- Large language models (LLMs) have demonstrated proficient capabilities across various tasks. They typically exhibit **varying strengths and weaknesses** across different tasks. Assembling multiple off-the-shelf LLMs can harness their complementary abilities, resulting in better performance than relying on a single LLM.
- Routing is a promising assembling method which **learns a router to select a suitable LLM for each query**. Compared with LLM ensembling, routing is much more efficient as it only needs to perform inference on the selected LLM.
- ZOOTER (NAACL, 2024) scores LLMs for each query, then minimizes Kullback-Leibler divergence between selection probability from the router and the softmax normalized score. However, when multiple LLMs perform well for a query, the normalized score tends to be uniform, which is not a strong supervision signal for learning the router.




**(a)**: Score distributions of LLMs on an example query (w/ or w/o normalization).




**(b)**: Distribution of the score difference between the top two LLMs.

## Scoring

Consider a set of LLMs $\{\mathcal{M}_t : t = 1, \dots, T\}$ and a training set $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i) : i = 1, \dots, n\}$, where $\mathbf{x}_i$ is a query (i.e., question) and $y_i$ is its answer (i.e., ground truth). We design a scoring method to assess the performance of LLMs on queries.

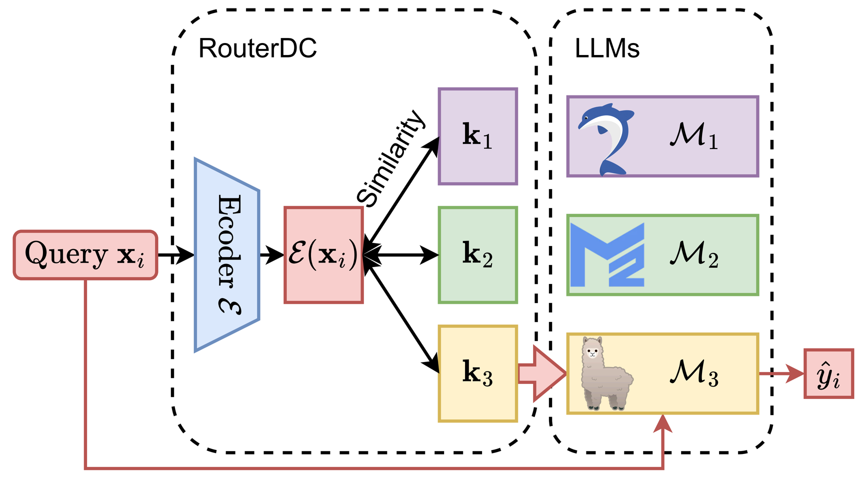
- For an *open-ended* generation query $\mathbf{x}_i$ (requiring a long answer, e.g., GSM8K), we feed it to LLM $M$ times to generate outputs $\{\hat{y}_{i,m}^{(t)} : m = 1, \dots, M\}$, then define the score of LLM $\mathcal{M}_t$ on the query $x_t$ as:

$$s_i^{(t)} = \frac{1}{M} \sum_{m=1}^{M} \text{evaluate}(\hat{y}_{i,m}^{(t)}, y_i)$$

- For a *multiple-choice question* $\mathbf{x}_i$ with an option set $\mathcal{A}_i$ (e.g., MMLU), we define the score based on the probability of options, i.e.,

$$s_i^{(t)} = \begin{cases} \frac{\mathbb{P}_{\mathcal{M}_t}(\hat{y}_i^{(t)}|\mathbf{x}_i)}{\sum_{a \in \mathcal{A}_i} \mathbb{P}_{\mathcal{M}_t}(a|\mathbf{x}_i)} & \text{if } \hat{y}_i^{(t)} = y_i \\ 0 & \text{otherwise} \end{cases}$$

## RouterDC Framework




The proposed RouterDC consists of

- An encoder $\mathcal{E}(\mathbf{x}; \mathbf{w})$ parameterized by $\mathbf{w}$ which maps $\mathbf{x}$ into an embedding in $\mathbb{R}^p$.
- $T$ learnable LLM embeddings $\{\mathbf{k}_t \in \mathbb{R}^p : t = 1, \dots, T\}$ for the $T$ LLMs.

For a query $\mathbf{x}_i$, RouterDC generates a selection probability distribution over $T$ LLMs as

$$R(\mathbf{x}_i; \boldsymbol{\theta}) = \text{softmax}\left[\text{sim}(\mathcal{E}(\mathbf{x}_i; \mathbf{w}), \mathbf{k}_1), \dots, \text{sim}(\mathcal{E}(\mathbf{x}_i; \mathbf{w}), \mathbf{k}_T)\right],$$

where $\boldsymbol{\theta} \equiv \{\mathbf{w}, \mathbf{k}_1, \mathbf{k}_2, \dots, \mathbf{k}_T\}$ is the learnable parameters in RouterDC and $\text{sim}(\cdot, \cdot)$ is the cosine similarity.

## Dual Contrastive Loss

### Sample-LLM Contrastive Loss

- Based on the score, we construct positive LLMs index set $\mathcal{I}_i^+$ and negative LLMs index set $\mathcal{I}_i^-$ as:
  1. $\mathcal{I}_i^+$ consists of the indices of LLMs corresponding to the top-$K_+$ scores.
  2. $\mathcal{I}_i^-$ consists of the indices of LLMs corresponding to the bottom-$K_-$ scores with $s_i^{(t)} < 0.5$.
- We expect the router to pull the query embedding $\mathcal{E}(\mathbf{x}_i; \mathbf{w})$ closer to the positive LLMs' embeddings $\{\mathbf{k}_{t_+} : t_+ \in \mathcal{I}_i^+\}$ while pushing apart from the negative LLMs' embeddings $\{\mathbf{k}_{t_-} : t_- \in \mathcal{I}_i^-\}$.

$$\mathcal{L}_{\text{sample-LLM}}(\mathbf{x}_i, y_i; \boldsymbol{\theta}) = \sum_{t_+ \in \mathcal{I}_i^+} -\log \frac{e^{\text{sim}(\mathcal{E}(\mathbf{x}_i;\mathbf{w}), \mathbf{k}_{t_+})}}{e^{\text{sim}(\mathcal{E}(\mathbf{x}_i;\mathbf{w}), \mathbf{k}_{t_+})} + \sum_{t_- \in \mathcal{I}_i^-} e^{\text{sim}(\mathcal{E}(\mathbf{x}_i;\mathbf{w}), \mathbf{k}_{t_-})}}$$

### Sample-Sample Contrastive Loss

- Minimizing the sample-LLM contrastive loss alone is not stable. Some similar queries can have dissimilar embeddings and may be routed to different LLMs.
- Training samples are grouped into $N$ groups $\{\mathcal{K}_1, \dots, \mathcal{K}_N\}$ by applying $k$-means algorithm on extracted t-SNE low-dimensional vectors. For a query $\mathbf{x}_i \in \mathcal{K}_j$, we randomly select an in-group query $\mathbf{x}_i^+ \in \mathcal{K}_j$ and an out-group set $\mathcal{X}_i^- \subset \{\cup_{j' \neq j} \mathcal{K}_{j'}\}$ of $H$ queries from the training mini-batch at each iteration.

$$\mathcal{L}_{\text{sample-sample}}(\mathbf{x}_i; \boldsymbol{\theta}) = -\log \frac{e^{\text{sim}(\mathcal{E}(\mathbf{x}_i;\mathbf{w}), \mathcal{E}(\mathbf{x}_i^+;\mathbf{w}))}}{e^{\text{sim}(\mathcal{E}(\mathbf{x}_i;\mathbf{w}), \mathcal{E}(\mathbf{x}_i^+;\mathbf{w}))} + \sum_{\mathbf{x}_i^- \in \mathcal{X}_i^-} e^{\text{sim}(\mathcal{E}(\mathbf{x}_i;\mathbf{w}), \mathcal{E}(\mathbf{x}_i^-;\mathbf{w}))}}.$$

### Training

- We learn a router $R(\mathbf{x}; \boldsymbol{\theta})$ by minimizing the final objective consisting of sample-LLM and sample-sample contrastive losses, i.e.,

$$\mathcal{L}(\mathcal{D}_{\text{train}}; \boldsymbol{\theta}) = \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{train}}} \mathcal{L}_{\text{sample-LLM}}(\mathbf{x}_i, y_i; \boldsymbol{\theta}) + \lambda\, \mathcal{L}_{\text{sample-sample}}(\mathbf{x}_i; \boldsymbol{\theta})$$

## Experiments

**Table 1**: Testing accuracy (%) on in-distribution tasks. "Time" denotes the total inference time in minutes.

| | | MMLU | GSM8K | CMMLU | ARC-C | HumanEval | Avg | Time (m) |
|---|---|---|---|---|---|---|---|---|
| *Candidate LLMs* | Mistral-7B | 62.14 | 36.71 | 43.83 | 49.43 | 28.98 | 44.22 | 6.94 |
| | MetaMath-Mistral-7B | 59.86 | 69.63 | 43.83 | 48.30 | 29.80 | 50.28 | 7.23 |
| | zephyr-7b-beta | 59.81 | 33.00 | 42.82 | 57.95 | 22.04 | 43.13 | 6.73 |
| | Chinese-Mistral-7B | 57.42 | 41.03 | 49.67 | 43.47 | 21.43 | 42.60 | 7.11 |
| | dolphin-2.6-mistral-7b | 60.53 | 52.38 | 43.71 | 52.56 | 45.10 | 50.86 | 6.91 |
| | Meta-Llama-3-8B | **64.59** | 47.76 | **51.77** | 49.43 | 26.73 | 48.06 | 6.33 |
| | dolphin-2.9-llama3-8b | 59.46 | 69.81 | 44.72 | 49.43 | 49.39 | 54.56 | 5.33 |
| | Voting | 63.30 | 67.39 | 47.48 | 50.85 | 42.85 | 54.37 | 46.59 |
| *Routing* | CosineClassifier | 59.72 | 69.03 | 45.47 | 50.57 | 46.33 | 54.22 | 8.30 |
| | ZOOTER | 60.48 | 66.69 | 45.27 | 53.13 | 44.29 | 53.97 | 8.01 |
| | LoraRetriever (clustering) | 63.33 | 66.63 | **51.77** | 57.10 | 40.00 | 55.77 | 7.86 |
| | RouterDC | 61.07 | **70.32** | **51.77** | **58.52** | **51.02** | **58.54** | 7.97 |

- RouterDC achieves the **highest** average accuracy, surpassing the best individual LLM (i.e., dolphin-2.9-llama3-8b)
- RouterDC is **better** than ZOOTER and CosineClassifier, demonstrating that the proposed dual contrastive losses can train a more effective router. RouterDC **outperforms** LoraRetriever, validating the usefulness of the sample-LLM contrastive loss.
- RouterDC is about $6\times$ **faster** in inference than voting.

**Table 2**: Testing accuracy (%) on out-of-distribution tasks. "Time" denotes the total inference time in minutes.

| | | PreAlgebra | MBPP | C-EVAL | Avg | Time (m) |
|---|---|---|---|---|---|---|
| *Candidate LLMs* | Mistral-7B | 24.80 | 37.90 | 46.43 | 36.38 | 4.31 |
| | MetaMath-Mistral-7B | 39.15 | 37.74 | 45.17 | 40.69 | 4.13 |
| | zephyr-7b-beta | 20.78 | 31.14 | 44.87 | 32.26 | 4.30 |
| | Chinese-Mistral-7B | 18.48 | 29.64 | 48.44 | 32.19 | 4.40 |
| | dolphin-2.6-mistral-7b | 29.28 | 44.86 | 45.10 | 39.75 | 3.20 |
| | Meta-Llama-3-8B | 27.67 | 43.02 | **52.01** | 40.90 | 3.95 |
| | dolphin-2.9-llama3-8b | **39.72** | **47.34** | 44.80 | 43.95 | 3.15 |
| | Voting | 39.03 | 41.60 | 48.50 | 43.04 | 27.43 |
| *Routing* | CosineClassifier | 36.97 | 38.48 | 47.77 | 41.07 | 4.43 |
| | ZOOTER | 34.44 | 41.10 | 44.95 | 40.16 | 4.28 |
| | LoraRetriever (clustering) | 35.36 | 43.12 | **52.01** | 43.50 | 4.22 |
| | RouterDC | 38.81 | 46.80 | 51.93 | **45.85** | 4.24 |

## Summary

- Problem: harness the complementary abilities of LLMs.
- Propose a novel routing method RouterDC and two contrastive losses to train the router.
- Experimental results show that RouterDC effectively assembles LLMs and outperforms individual top-performing LLMs as well as existing routing methods.