**8th Place Solution**

I knew little about LoRA and model quantization at the start. Initially, I thought I could avoid finetuning LLMs entirely, but that was naive. Thanks to everyone who shared their code and insights.

**Two key references:**

* [@sayoulala](https://www.kaggle.com/sayoulala) : [https://www.kaggle.com/competitions/eedi-mining-misconceptions-in-mathematics/discussion/543519](https://www.kaggle.com/competitions/eedi-mining-misconceptions-in-mathematics/discussion/url)
  + Enlightened me on LoRA finetuning, hard negative mining, and recall+rerank training scheme.
* [@cdeotte](https://www.kaggle.com/cdeotte) : [https://www.kaggle.com/competitions/lmsys-chatbot-arena/discussion/521294](https://www.kaggle.com/competitions/eedi-mining-misconceptions-in-mathematics/discussion/url)
  + Inspiring ideas on model quantization, wise-FT LoRA adaptation, and VLLM inference.

**Inference pipeline:**

* Use 4 Qwen2.5-14b models as retrievers, ensemble their results, then feed the retrieved misconceptions to a single Qwen2.5-32b model as a reranker.

**Training details:**

* Retriever (Qwen2.5-14b): Finetuned with bit4 + LoRA for FEATURE\_EXTRACTION.
* Reranker (Qwen2.5-32b): Finetuned with bit4 + LoRA for CASUAL\_ML, outputting a “Yes” logit for each query–misconception pair.
* 5 CV-folds used for training retrievers. full dataset used for training reranker without validation.
* Use few-shots prompt with chatgpt to generate extra training data:
  + Added 983 synthetic samples to fill missing misconceptions.
  + For each fold, regenerated extra samples to cover unique misconceptions in that fold for training.

**Inference details:**

* Retrieving:
  + 4 retrievers (chosen from 5-fold CV based on LB scores) retrieve 4 times, then ensembled via weighted sum.
  + infer in bnb bit4+LoRA (merge unnecessary), so that each single retriever can fit in one T4 gpu, 4 retrievers only take 2 T4s running twice.
* Reranking:
  + model first merged with lora adaptor, with wise-FT hyper parameter alpha=0.8
  + model then quantized to AWQ and infer in vllm.
  + Each query is paired with 40 misconceptions retrieved, needs to turn on "enable\_prefix\_caching" to accerlerate inference.

**Key improvements (Public LB):**

* Add Synthetic data: single retriever public LB score from 0.3~ → 0.49
* Add reranker: 0.49(single retriever only) → 0.57
* Apply Wise-FT on reranker's LoRA merging: 0.57 → 0.59
* Ensemble retrievers with tuned weights: 0.59 → 0.61

**Personal note 1**

The entire pipeline—from LoRA finetuning to quantization—was very new to me, and I nearly gave up more than once. I started small with Qwen2.5-0.5B to establish a workable process, but the AWQ-quantized model kept producing random letters during inference. After much frustration, I discovered that autoawq didn’t export the im-head-embed layer parameters for Qwen2.5-0.5B, forcing me to add them manually. Without [@cdeotte](https://www.kaggle.com/cdeotte)‘s paper proving it could be done, I might have quit altogether. Thanks to that reassurance, I stuck with it and eventually got the pipeline running smoothly.

**Personal note 2**

This is a quick summary, I will add more details and takeaways when I have time for my learning purpose.