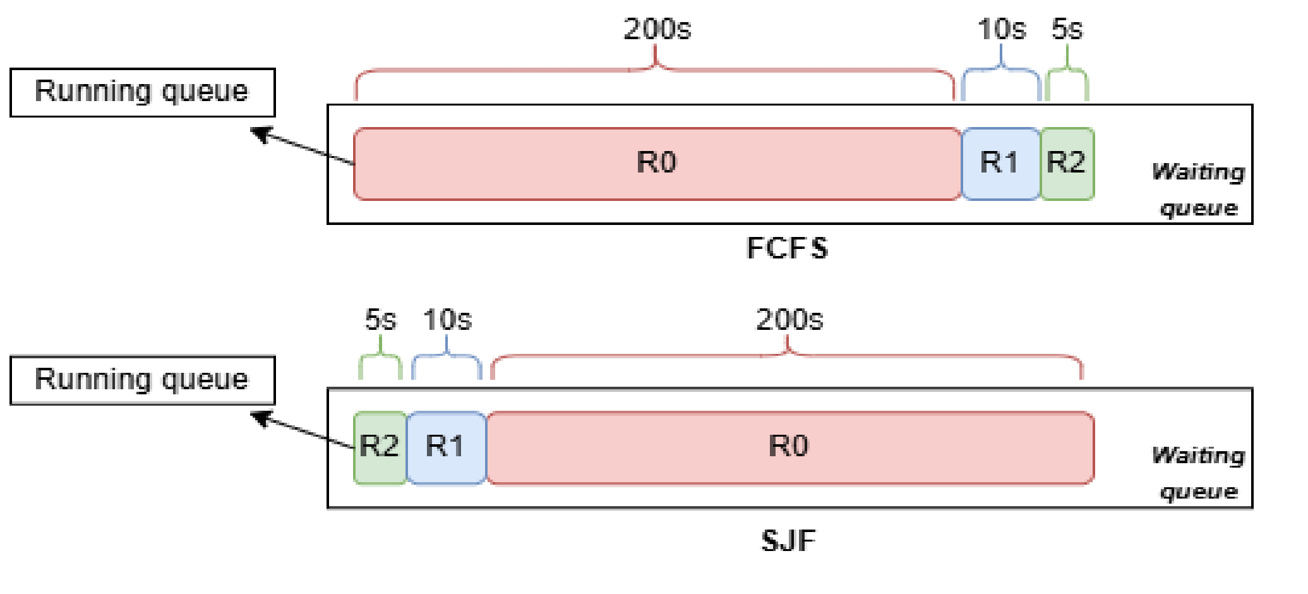
From Prompts to Priorities: Pairwise Learning-to-Rank Scheduling for Low-Latency LLM Serving

Yiheng Tao, Yihe Zhang, Matthew T. Dearing, Xin Wang, Zhiling Lan

Motivation

Large language model (LLM) requests vary widely: a short factual query may produce only a few dozen tokens, while multi-step reasoning or proof generation can run to tens of thousands. Because generation is autoregressive, inference time grows proportionally with output length. Current serving stacks such as vLLM and Orca schedule requests using a simple first-come-first-serve (FCFS) policy. While fair, FCFS suffers from a well-known issue in LLM serving called ***head-of-line blocking***, where a single long request delays all shorter ones behind it—leading to inflated tail latency and wasted throughput. A classical remedy is the Shortest-Job-First (SJF) policy, which improves efficiency by serving shorter jobs first. However, SJF requires knowledge of job length in advance—information that is unavailable in LLMs until generation completes, due to their autoregressive nature.

We introduce a prompt-sensitive LLM task scheduler based on pairwise learning method from learning-to-rank utilizing Margin Ranking Loss and show that our method significantly improves scheduling performance with minimal overhead.



SJF(Oracle) can reduce average latency by **94.5%** compared with FCFS

Figure 1: head-of-line blocking issues in LLM serving

Experimental Setup

Testbed

All experiments are conducted on a server equipped with two NVIDIA A100 GPUs (40GB each)and an Arm-based Neoverse-N1 CPU.

Models

We evaluate our scheduling method on two representative types of LLMs:

- **Llama:** A family of efficient, general-purpose autoregressive transformers designed for instruction following and language understanding, often used as a benchmark for standard generation tasks in LLM research.
- **DeepSeek R1**: A reasoning-oriented model that includes multi-step thinking traces in its output. For this model, we include the full reasoning process as part of the generation length, since these intermediate outputs contribute significantly to actual inference time in practical deployments.

Datasets

We conduct our evaluations on two real-world prompt datasets:

- **Alpaca**: A widely used instruction-tuning dataset containing diverse natural language prompts across a broad range of tasks.
- **LMSYS-Chat-1M**: A large-scale, multi-turn conversation dataset containing over one million real-world user interactions across multiple LLMs.

Our Design

Pairwise Learning-to-Rank: Instead of predicting absolute response lengths—a difficult task due to the variability of LLM outputs—we frame the problem as a pairwise ranking task. Given two prompts, the model learns to predict which one is likely to produce a longer response. This pairwise ranking formulation guides our training strategy: the dataset consists of prompt pairs, each annotated with a binary label $y \in \{-1, 1\}$, indicating which prompt is expected to generate the longer output. Our model architecture includes a BERT encoder followed by a fully connected (FC) layer.

Integrating the Predictor into LLM Requests Scheduling: During training, our ranking predictor learns to distinguish prompts through pairwise comparisons. However, applying pairwise comparisons to all pending prompts at runtime is computationally expensive. To address this, we leverage BERT's non-autoregressive architecture, which allows batch processing of prompts in parallel. All queued prompts are scored independently in a single forward pass, and the scores are sorted to produce the final ranking. This design ensures that our scheduler remains efficient and scalable under high system load.

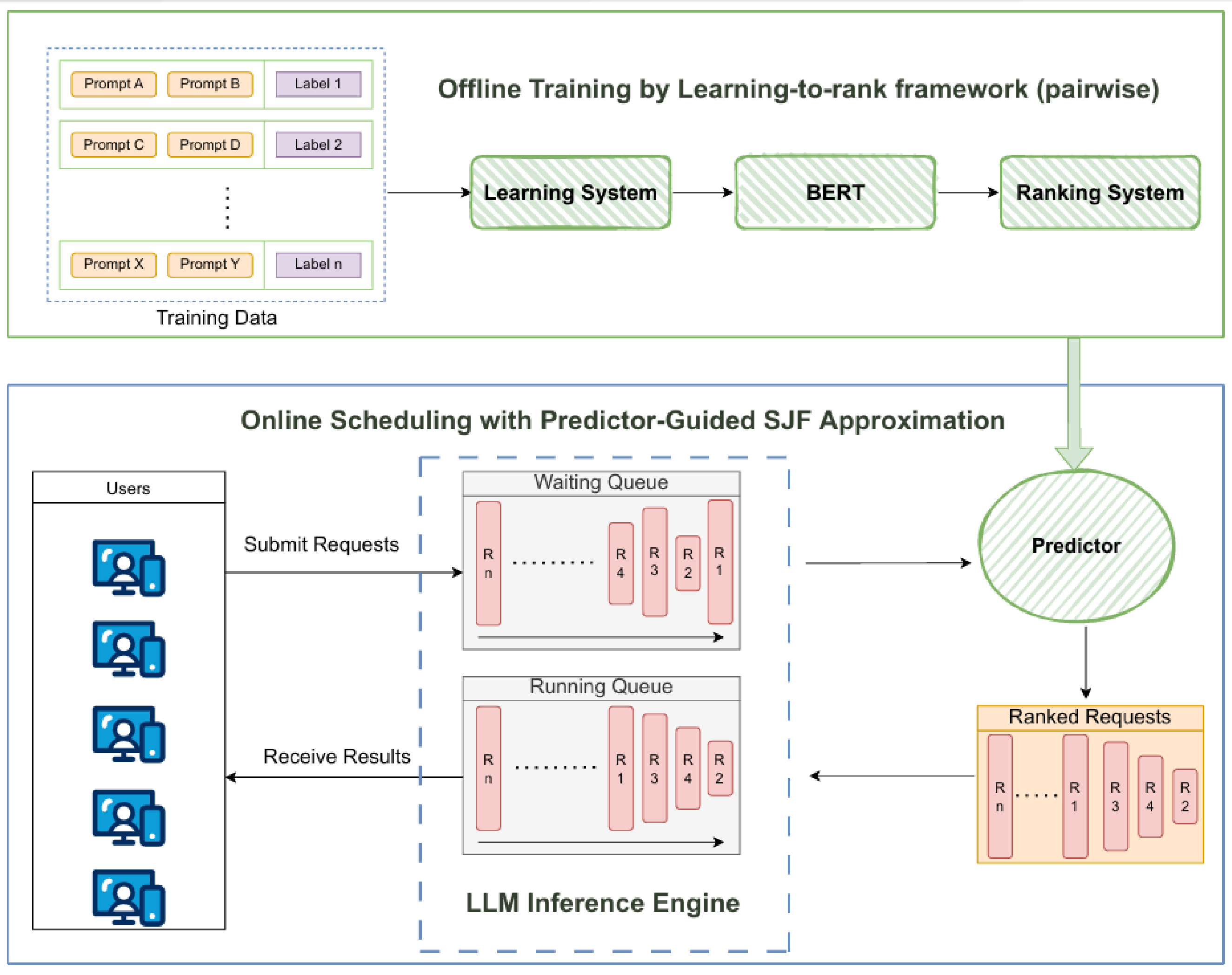


Figure 2: Learning-to-Ranking pairwise framework design

Prediction Accuracy

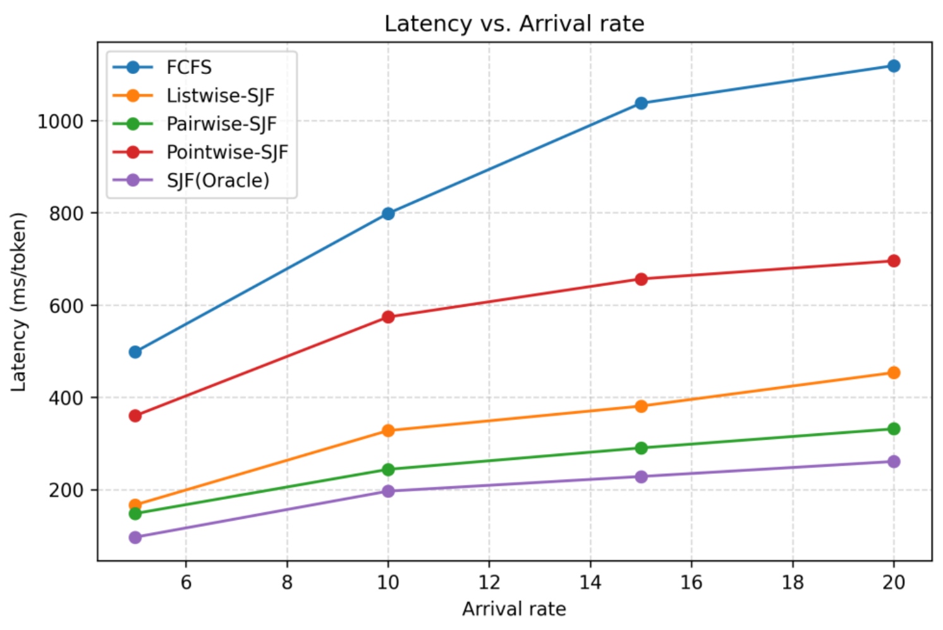
Dataset	Listwise	Pointwise	Pairwise
Alpaca (GPT-4)	0.69	0.70	**0.89**
Alpaca (Llama)	0.67	0.64	**0.72**
Alpaca (R1)	0.50	0.30	**0.59**
LMSYS-Chat-1M (GPT-4)	0.63	0.33	**0.70**
LMSYS-Chat-1M (Llama)	0.52	0.37	**0.65**
LMSYS-Chat-1M (R1)	0.54	0.43	**0.66**

Table 1: KENDALL'S TAU (TAU-B) COMPARISON ACROSS DATASETS ANDRANKING APPROACHES

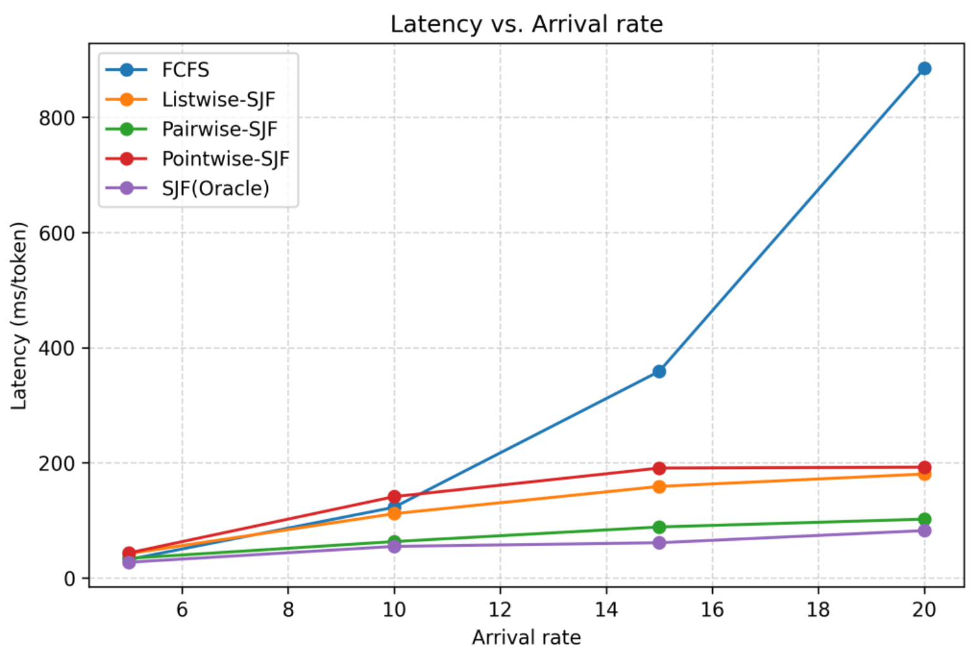
A **Kendall's Tau** value of 1 indicates perfect agreement with the ground truth ranking, 0 implies no correlation, and -1 represents complete reversal. Here, a higher Tau means the predicted ranking of prompts more closely mirrors the actual ordering based on true response lengths.

Scheduling Performance

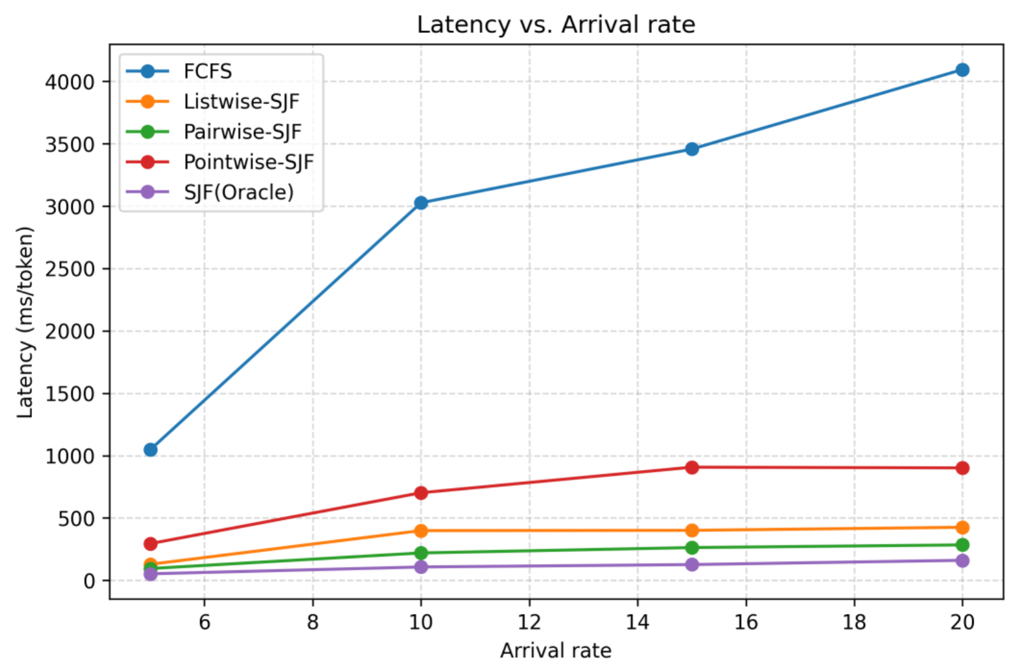
Average Latency using Different Strategies tested on Deepseek-R1 and Llama model using Alpaca and Lmsys dataset



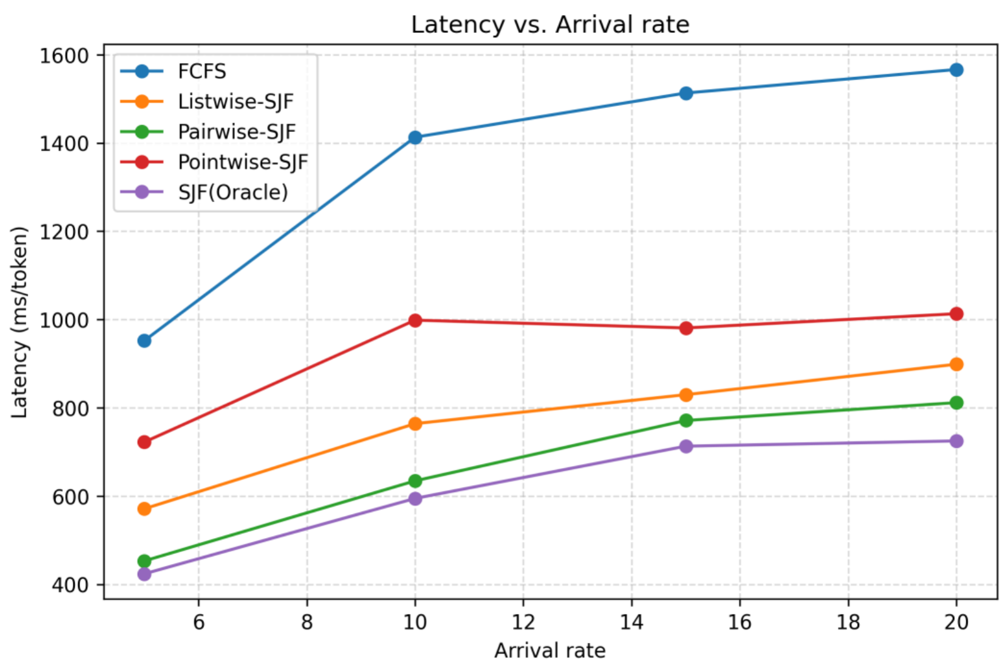
Deepseek-R1 using Alpaca



Llama using Alpaca



Llama using Lmsys



Deepseek-R1 using Lmsys

UNIVERSITY OF ILLINOIS CHICAGO