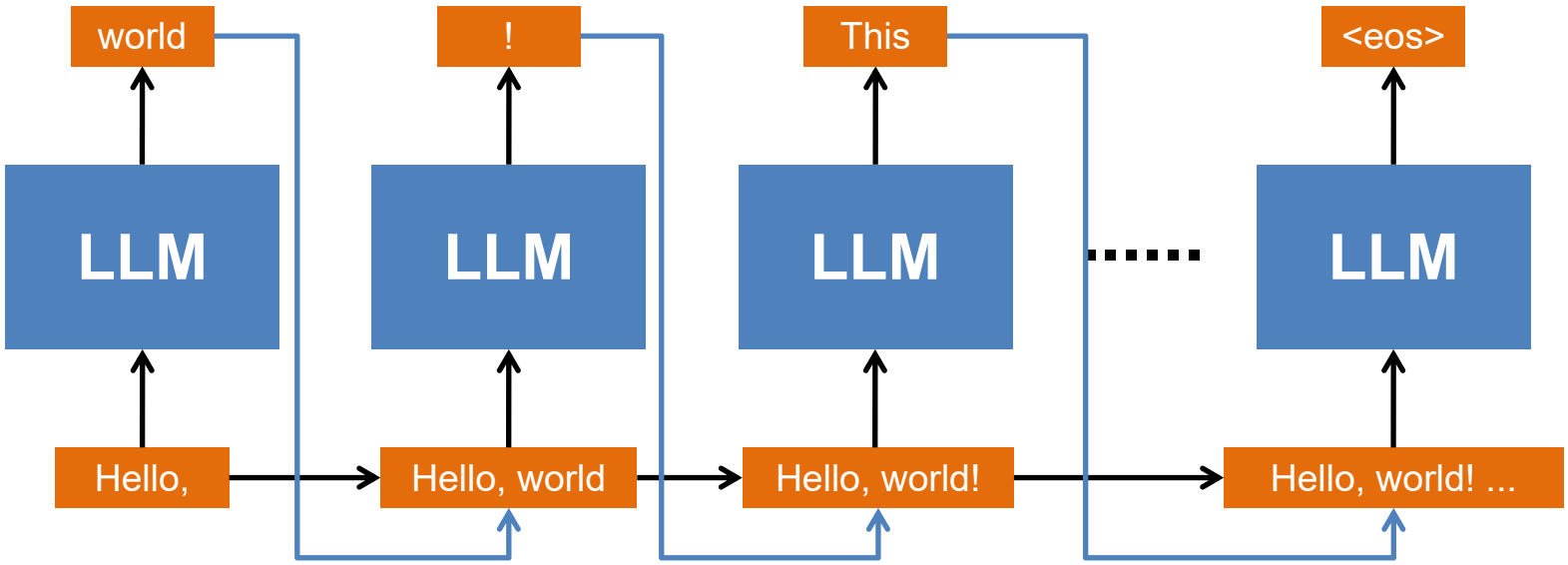
UC San Diego

NEURAL INFORMATION PROCESSING SYSTEMS

Efficient LLM Scheduling By Learning To Rank

Yichao Fu, Siqi Zhu, Runlong Su, Aurick Qiao, Ion Stoica, Hao Zhang

LLM Decoding Process

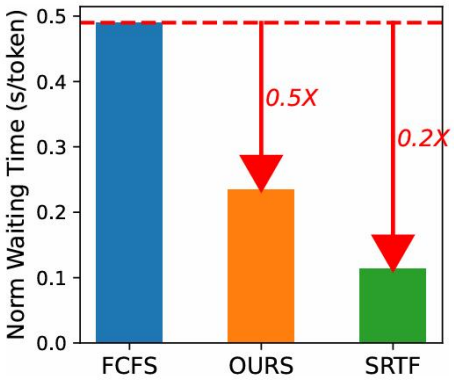


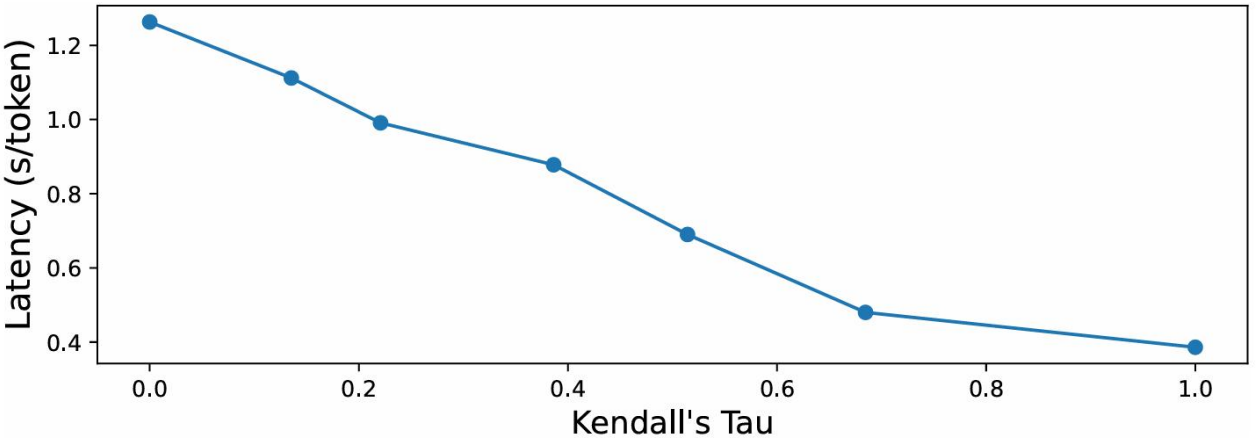
The output length of the LLM is unpredictable due to the autoregressive decoding process.

Head-of-Line Blocking in LLM Serving

LLM services that implement a first-come-first-serve (FCFS) scheduling inevitably face significant Head-Of-Line (HOL) blocking.

Ranking is All You Need (to appx. SJF)

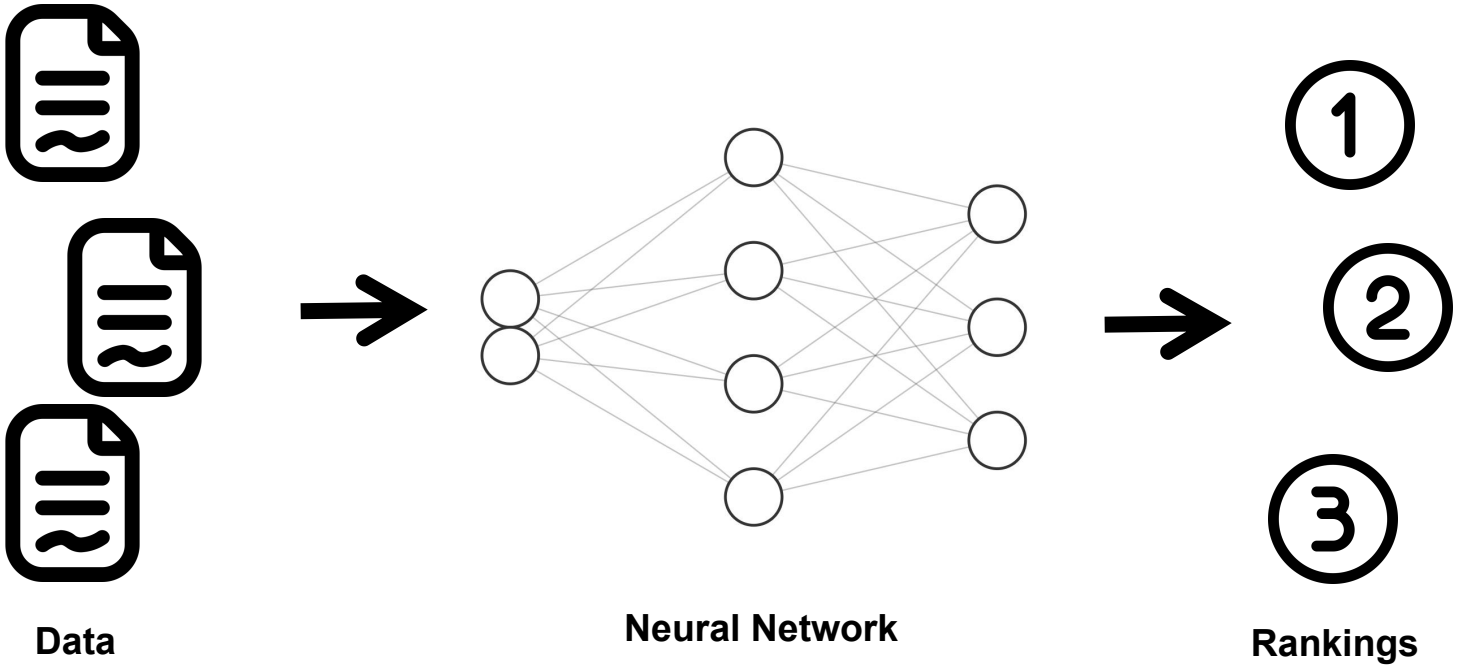




The precise generation length is not needed.

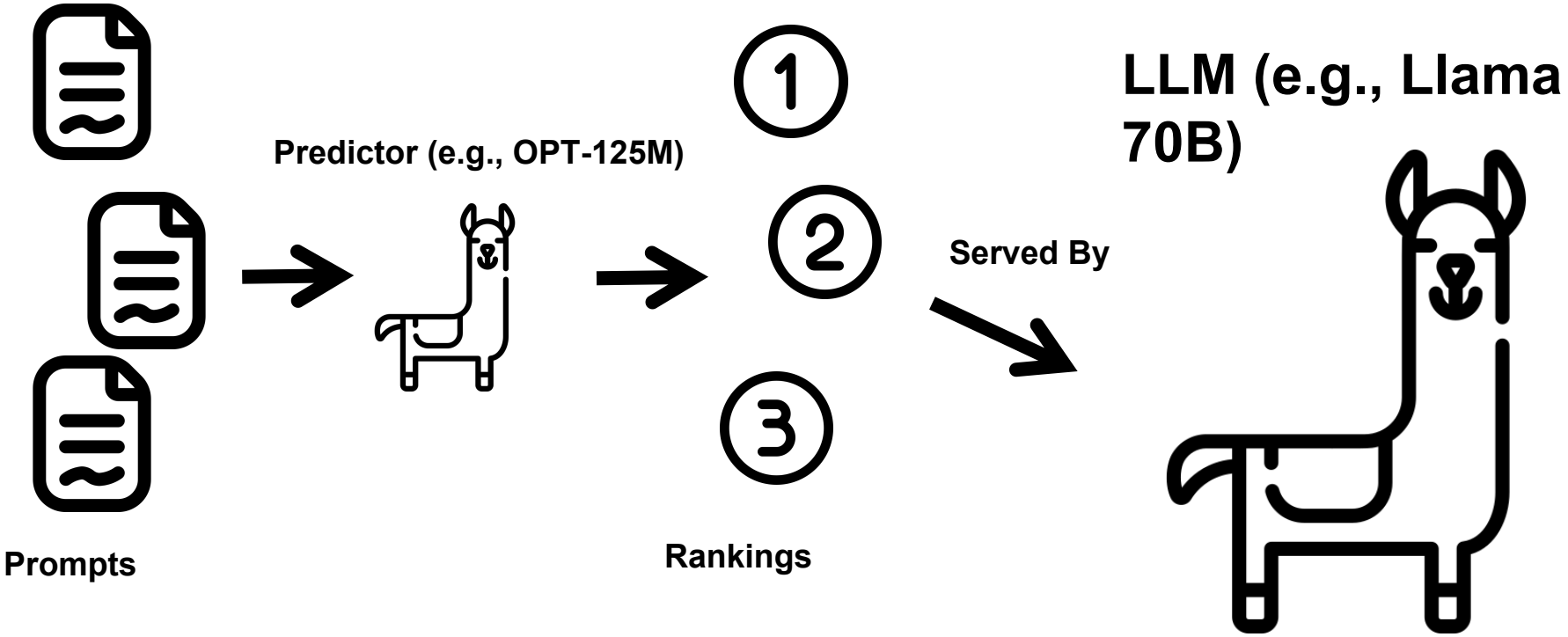
An accurate generation length ranking prediction is enough.

Learning To Rank



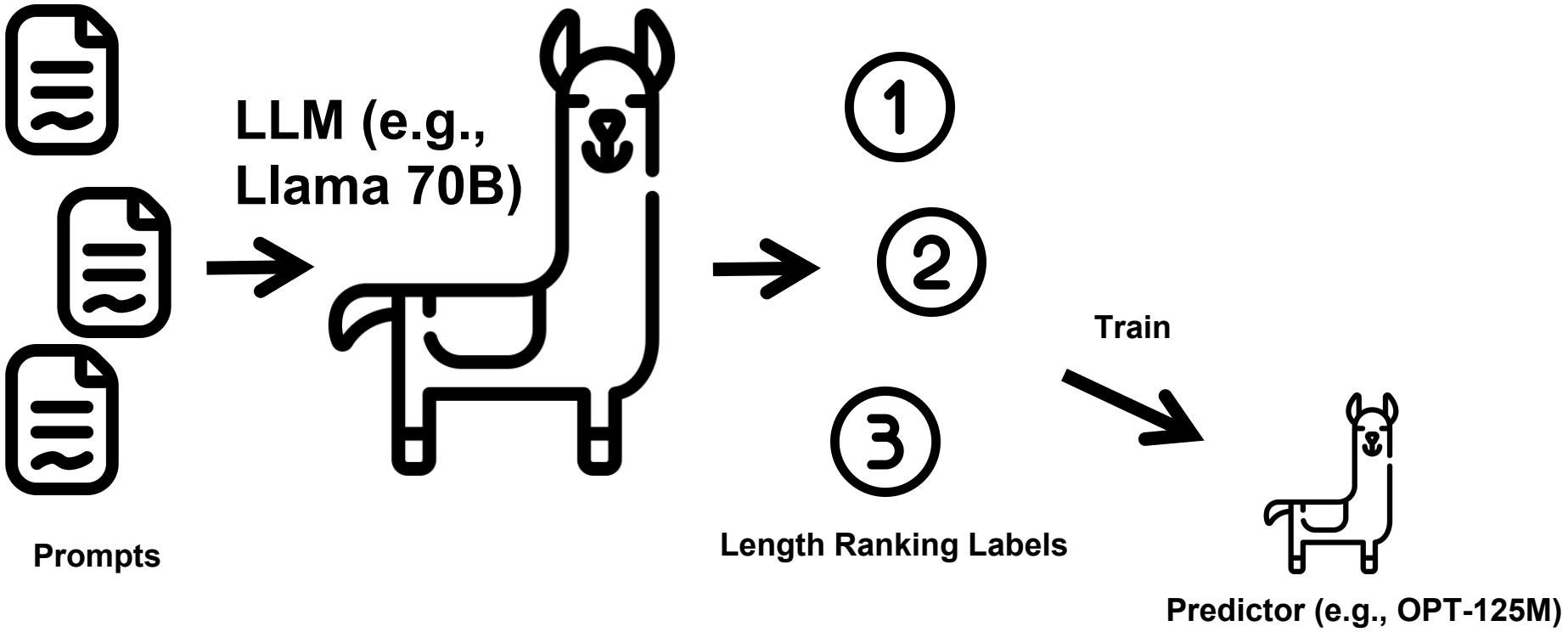
Learning to rank (LTR) applies machine learning methods to ranking supervised data.

Generation Length Ranking Predictor



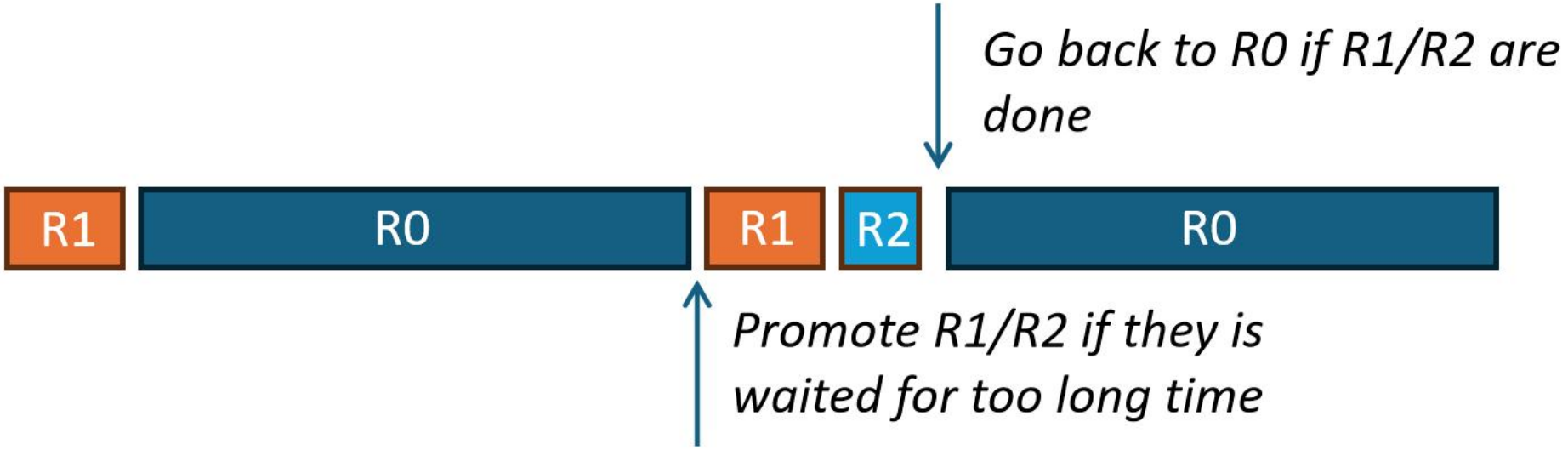
The predictor estimates the relative length of responses for incoming prompts, allowing them to be efficiently ordered before being processed by the target language model.

Train Length Ranking Predictor



The predictor can be rapidly trained on live production data within minutes (e.g., 5 minutes) during actual LLM deployment.

Starvation Prevention



$$max\_waiting\_time = \max(TTFT, \max(TPOT)).$$

Evaluation

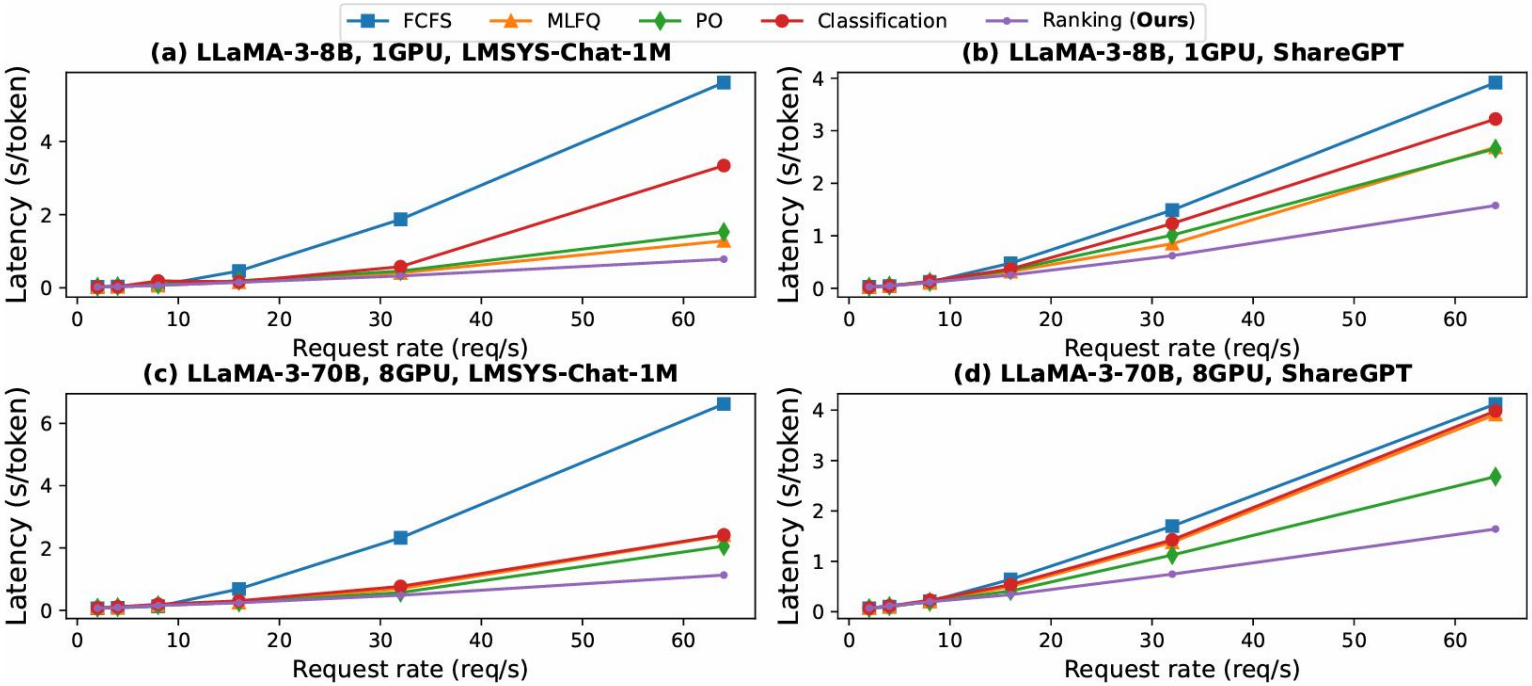


Figure 3: Mean latency of different schedulers with Llama-3 models on real workloads.

Our proposed method improve the mean latency by up to 6.9× compared with FCFS and from 1.5×–1.9× compared with PO in Chatbot Serving.

Thank You!