JITServe: SLO-aware LLM Serving with Imprecise Request Information

Wei Zhang*, **Zhiyu Wu***, Yi Mu, Rui Ning, Banruo Liu,
Nikhil Sarda, Myungjin Lee, Fan Lai

LLMs Power Diverse Real-World Applications

CHATBOT

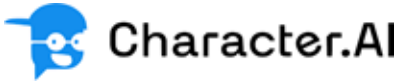
Gemini

PRODUCTIVITY

CODE

ENTERTAINMENT

Seedance 2.0



SLO Requirements are Heterogeneous

- Users exhibit ***diverse SLO*** needs both across and within applications.
- No single latency metric can capture ***application-level SLO*** needs.

LLM Applications	Real-Time	Direct Use	Content-Based
Code generation	**38.1%**	30.5%	31.4%
Report generation	**39.1%**	36.2%	24.7%
Deep research	38.6%	**47.1%**	14.3%
Real-time translation	36.2%	**39.9%**	23.9%
Batch data processing	15.6%	**49.6%**	34.8%
Reasoning task	28.9%	**47.4%**	23.7%

Characterizing LLM Serving Requests

What is *latency-sensitive* request?

TTFT and TBT matter

Each token must arrive in time to match the user's reading pace.

ChatGPT

What is *deadline-sensitive* request?

TTLT matters

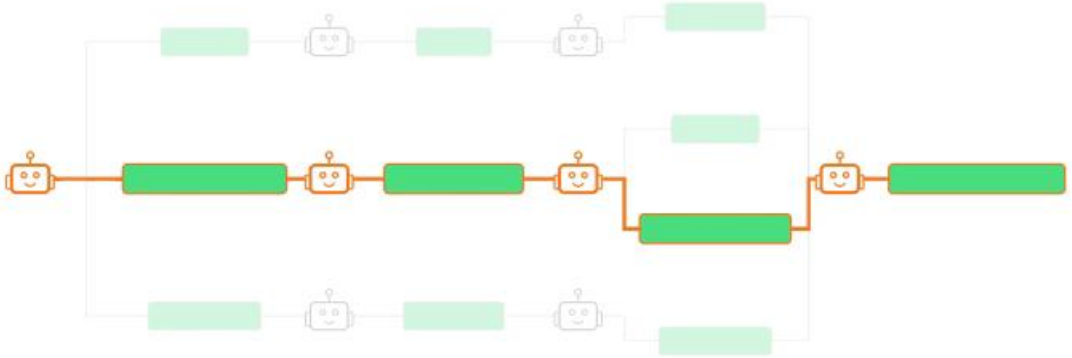
The full response must arrive before the downstream tool times out.

Characterizing LLM Serving Requests

What is ***compound/agentic*** request?

Critical path

End-to-end latency matters



OPENCLAW

TTFT/TBT
TTLT
E2E Latency

A single serving system must satisfy all three!

Define SLO/Goodput Across Request Types

CATEGORIZATION

- **Latency-sensitive:** The *i*-th token counts only if delivered before $TTFT + i \cdot TBT$.
- **Deadline-sensitive:** All tokens count only if the full response finishes before deadline.
- **Compound/Agentic:** All tokens across sub-requests count only if end-to-end DAG finishes before deadline.

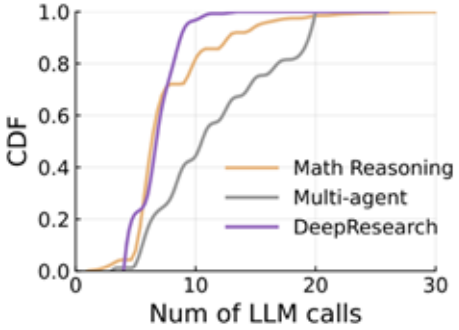
One rule: a token contributes to goodput only when its SLO is met.

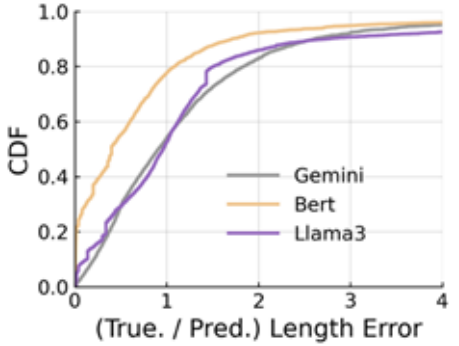
How to Handle Request Uncertainties?

PROBLEMS

- **Structural uncertainty:** Execution DAG & LLM call count unknown until runtime.
- **Length uncertainty:** LLM-based predictors are highly unreliable.

New ways to capture request uncertainties!

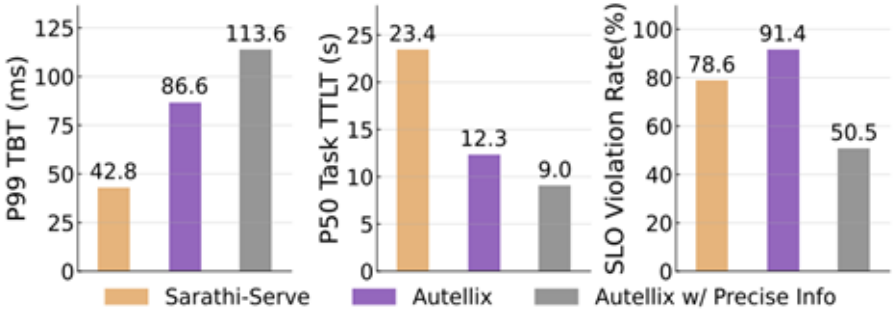




How to Align Efficiency and Service Goodput?

The Misalignment

- Latency-optimized schedulers miss SLOs (78.6–91.4%).
- Even with precise length info, miss rate stays at 50.5%.
- *Lower TTFT/TBT ≠ Lower SLO violation rate.* Policy matters more than prediction.

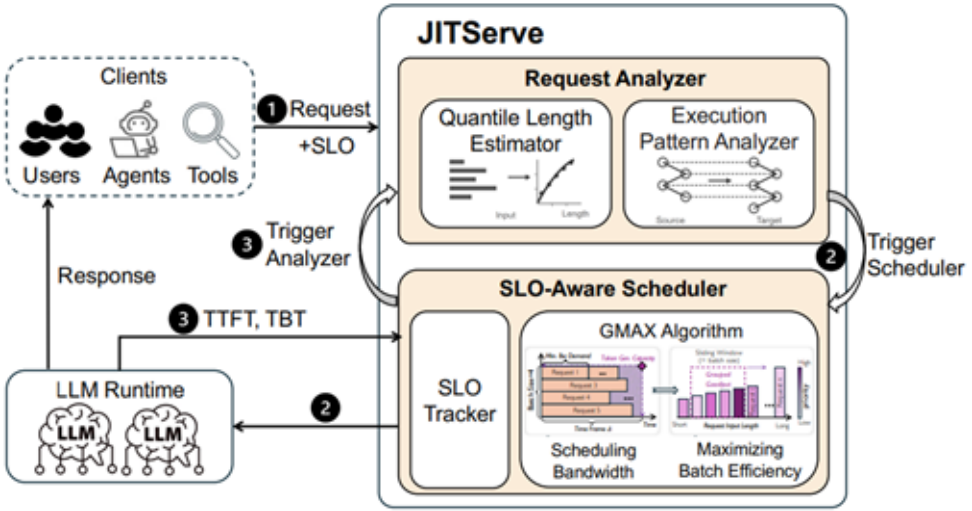


Just-In-Time Scheduling: allocate just enough bandwidth per SLO.

JITServe Overview

COMPONENTS

- **Request Analyzer:** Refine length & Dependency estimates.
- **SLO-aware Scheduler:** Allocate minimum bandwidth per frame.
- **SLO Tracker:** Monitors generation speed and feedbacks.



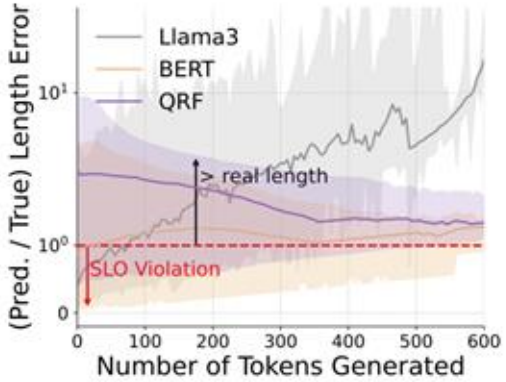
Philosophy: Schedule conservatively, relax as estimates refine online.

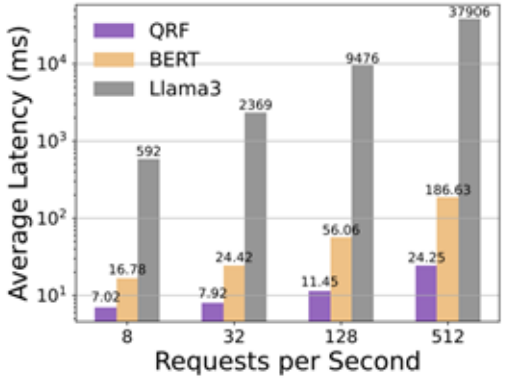
Length Uncertainty: QRF Upper-Bound Estimation

LENGTH ESTIMATOR

- **Request-adaptive**: Customizes quantile intervals per request.
- **Online Refinement:** Continuously sharpens predictions as more tokens are generated, improving accuracy over time.

Lightweight & Supports online refinement

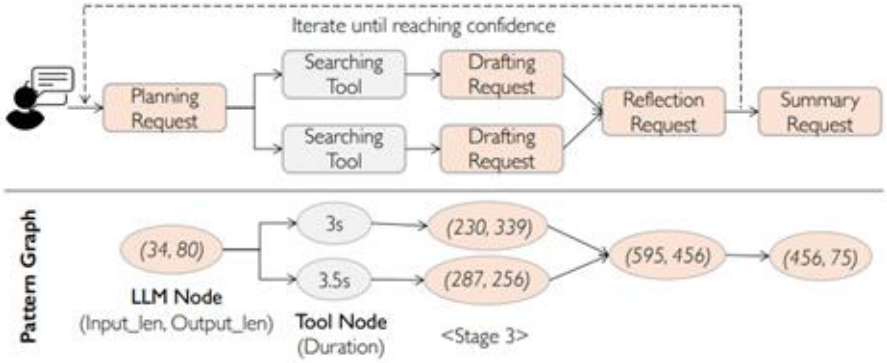




Dependency Uncertainty: Pattern-Graph Matching

PATTERN ANALYZER

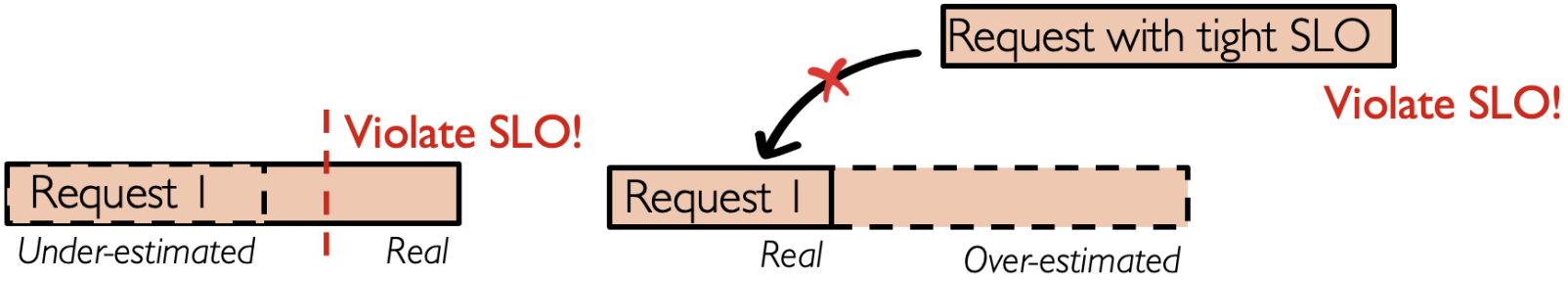
- **Idea**: Pattern graph Representation & Similarity matching.
- **Efficiency**: K-medoids mechanism & Low reuse frequency eviction.



Continuously refine matching as more tokens/stages appear.

GMAX: JIT-Scheduling Under Uncertainty

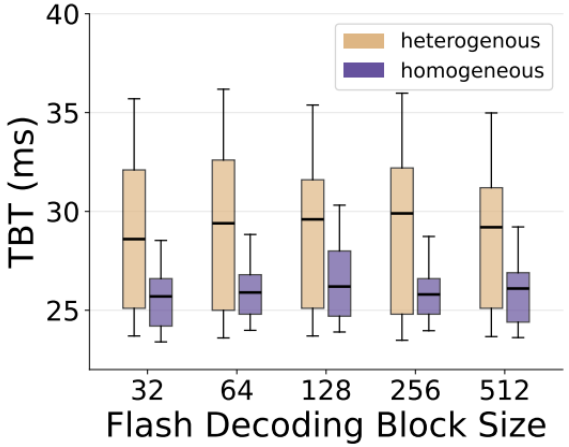
- **Challenge 1:** **Scheduler only knows the imprecise information.**



Static scheduling decisions are 'too early' !

GMAX: JIT-Scheduling Under Uncertainty

- Challenge 1: Scheduler only knows the imprecise information.
- **Challenge 2: Imbalanced batch length will influence per-step SLO(TBT).**



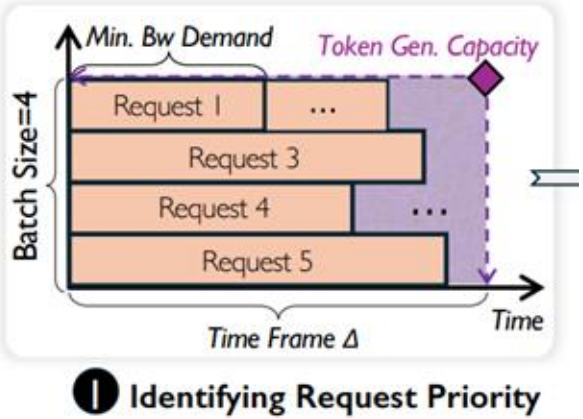
Stragglers leads to TBT violations for batches requests.

GMAX: JIT-Scheduling Under Uncertainty

SLO-aware Scheduler

- **SLO-Centric:** **Minimum-bandwidth for SLO guarantees.**

Conservative Serving!



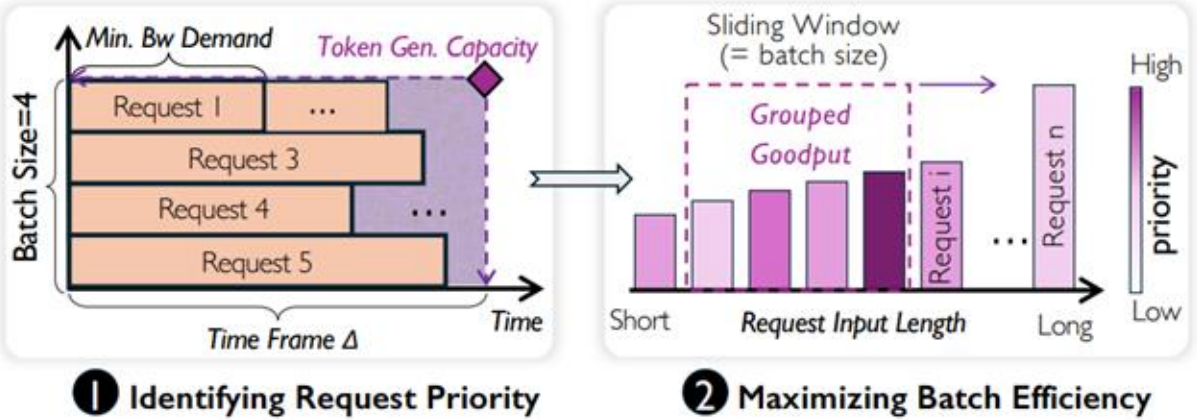
❶ Identifying Request Priority

Priority = Goodput Benefit / Generation Time

GMAX: JIT-Scheduling Under Uncertainty

SLO-aware Scheduler

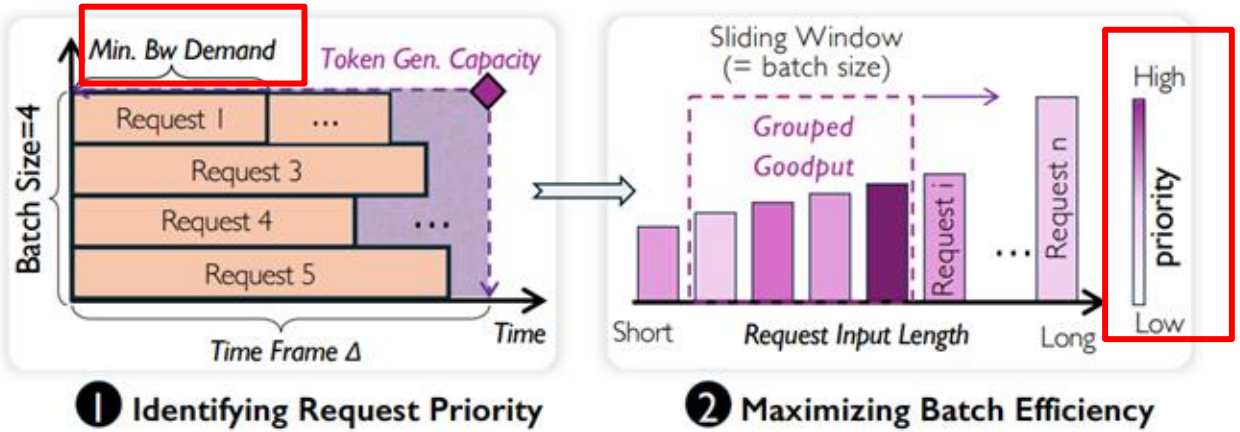
- SLO-Centric: Minimum-bandwidth for SLO guarantees.
- **JIT-Scheduling: Maximizing Grouped marginal goodput.**



GMAX: JIT-Scheduling Under Uncertainty

SLO-aware Scheduler

- SLO-Centric: Minimum-bandwidth for SLO guarantees.
- JIT-Scheduling: Maximizing Grouped marginal goodput.
- **Online-Refinement: Dynamically correct of length estimates.**

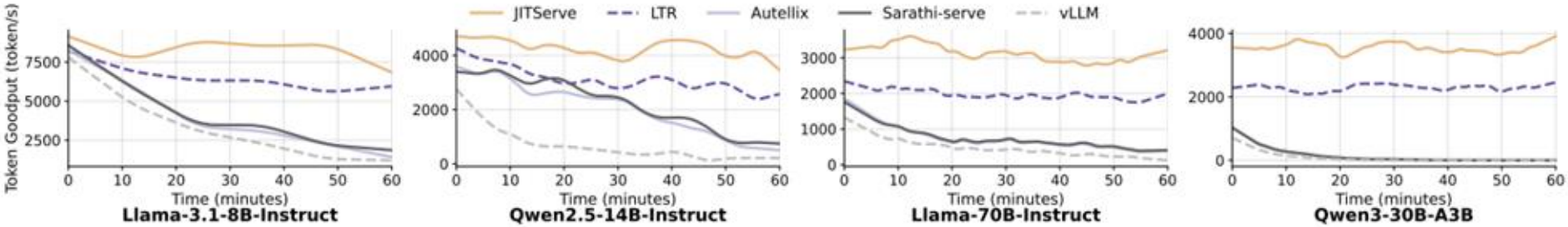


Provable guarantee

EVALUATION SETUP

- **Workloads:** Diverse real-world traces
  - ***Single Requests:*** Alpaca, LMsys-chat
  - ***Agentic Requests:*** Deep Research, Math Reasoning, Code Generation
- **Hardware:** 16x NVIDIA A100 GPUs
- **Metrics:**
  - ***SLO Metrics:*** Token goodput, Request Goodput
  - ***Traditional Metrics:*** TTFT, TBT, Throughput
- **Baseline:**
  - ***Standard:*** vLLM, Sarathi-Serve(Chunked-Prefill),
  - ***Policy-based:*** Least-Attained-Service, Learning-to-Rank(Shortest Job First),

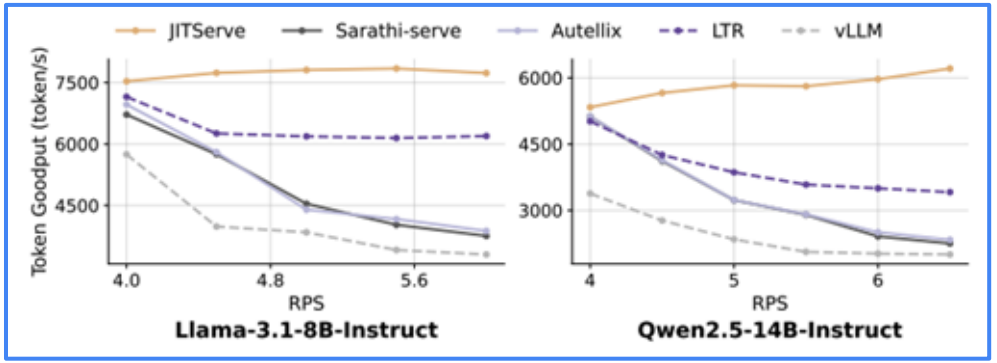
MAIN RESULT

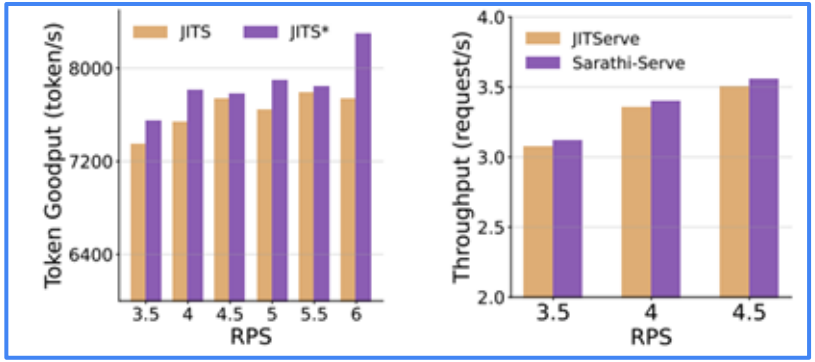


JITServe achieves 1.4×-6.3× goodput and 28.5%–83.2% resource savings.

- **Superior Scalability:** Consistently outperforms baselines across various model scales *(from 8B to 70B)*.
- **Sustained Performance:** Effectively mitigates performance decay during prolonged high-load periods *(60-min trace)*.

MAIN RESULT





JITServe achieves 1.4×-6.3× goodput and 28.5%–83.2% resource savings.

- **High-Pressure Resilience:** Performance gap widens as RPS increases, dominating in congested scenarios.
- **Near-Oracle Performance:** Effectively bridges the gap caused by imprecise request information.

JITServe

SCALABILITY

Maintains 1.3–2.4x gains as replica scales.

ADAPTABILITY

Resolves execution uncertainty in agentic requests.

COST-SAVING

Achieves up to 83.2% resource savings.

Conservative  **JIT Refinement**

Thank you!