Minimax Theory and its Applications
Volume 11 (2026), No. 4s, 756 to 767

Robust Optimization for Service-Level-Constrained Resource Allocation in Large Language Model Inference Serving

Dr. Sheeja V. S[1], Vijaya Priya A[2], Dr. N. Kalyana Sundaram[3], Dr. Sreerama Raju Malladi[4], Dr. Tavanam Venkata Rao[5], Dr.S.Ashok[6]

[1]Associate Professor, Department of Management Studies, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Email: sheejavs@maher.ac.in, ORCID ID: 0000-0002-7611-4819
[2]Assistant Professor, Department of Computer Technology, Sri Ramakrishna College of Arts and Science,
Email: vijayapriya@srcas.ac.in, ORCID ID: 0009-0000-4911-8841
[3]Department of Information Technology, New Prince Shri Bhavani College of Engineering,
Email: n.kalyanasundaram@gmail.com, ORCID ID: 0009-0005-7053-5718
[4]Assistant Professor, School of Business, Aditya University, Surampalem, Andhra Pradesh – 533437,
Email: sreeramarajum@adityauniversity.in, ORCID ID: 0000-0003-3442-9596
[5]Professor, Department of Electronics and Communication Engineering, Sreenidhi Institute of Science and Technology, Email: vrtavanam@gmail.com, ORCID ID: 0000-0002-9158-2231
[6]Associate Professor, Depatment:ECE ,College:Vel Tech Multi Tech Dr Rangarajan DrSakuntala engineering college, Email: ashoksvkas@gmail.com, ORCID ID: 0000-0002-8852-6925

Abstract
Several Large language models (LLMs) show most of the incoming request rates and token workloads, which pose challenges for resource allocation when meeting latency Service-Level Objectives (SLOs). While prior LLM serving has advanced state-of-the-art scheduling, batching, memory management, and placement of resources, many such techniques depend on observed or estimated workloads to tune the allocation of resources, offering limited guarantees when request rates and token workloads arrive at jointly uncertain levels. This paper proposes a robust optimization to allocate resources under joint workload uncertainty. The request arrival rate and token workload are captured by a budgeted uncertainty set, and the analytical latency function governs the relationship between the workload and the computational capacity allocated. In the proposed formulation, the resource cost is minimized while a worst-case latency SLO is respected over the considered uncertainty set. A central contribution of this paper is the derivation of an effective workload that encapsulates the two-dimensional uncertainty budget into a singular worst-case workload, enabling analytical tractability of the robust resource allocation formulation. Due to the limited number of decision variables and constraints in the resulting formulation, it is amenable to practical capacity planning. Analytical results characterize the optimal allocation as a function of the level of uncertainty, workload intensity, SLOs, and resource efficiency. With the framework developed in this paper, one can directly interpret how much extra resource capacity is needed to guard against the joint uncertainty in workload. Numerical experiments are conducted to evaluate the nominal, single-source robust, and joint-robust allocation policies in terms of resource cost and SLO violations.

Keywords: Large Language Models; Inference Serving; Robust Optimization; Resource Allocation; Service-Level Objectives; Workload Uncertainty; Token Workload; Latency

1. Introduction

LLMs are mainly deployed on online inference services for conversational agents, code generation, retrieval-augmented applications, and other latency-sensitive applications. The requirements for the LLM's provisioning vary depending on the request arrival rates, the lengths of the input and output tokens, batching conditions, and the resources of the accelerator that are available. As such, resource allocation should balance computational efficiency and SLOs, with a focus on latency requirements (Kown et al., 2023; Patel et al., 2024).

One limitation is that the number of requests that are being serviced by itself does not capture the workload characteristics of the LLM inference job well. A workload consisting of a moderate rate of requests with long request token sequences may require more resources than a workload with a high request rate but short requests. Thus, it is the combination of request arrival rates and token demand per request that determines the effective

workload. Such a combination is of particular interest during workload bursts, when both the request rate and token lengths may spike simultaneously.

In recent lines of work, LLM-serving systems have addressed the challenges of resource utilization and adherence to SLOs via scheduling, batching, memory management, routing and the phase-aware allocation of resources. Paged Attention and vLLM improve upon serving efficiency through a more effective handling of the KV-cache (Kwon et al., 2023), while DistServe and Split wise take advantage of the differences between the prefill and decoding stages to improve resource utilization and latency performance (Zhong et al., 2024; Patel et al., 2024). Further investigations have devised scheduling and resource allocation approaches that are conscious of SLOs for heterogeneous LLM workloads (Chen et al., 2025; Zhang et al., 2025).

Still, resource-based decisions are primarily estimated under workload conditions, which may become impractical when the demand deviates from its minimal value.This optimisation also provides an alternative when the probability distribution of the demands becomes unavailable and alsospecifies bounds on parameter deviation. This can seek a solution that remains feasible for all realisations within the uncertainty set (Ben-Tal &Nemirovski, 1998; Bertsimas & Sim, 2004).

The paper focuses on the joint uncertainty in request arrival rate and token workload. Instead of treating the joint uncertainty in request arrival rate and token workload separately, a budgeted uncertainty set is formulated to allow the joint uncertainty in request arrival rate and token workload to deviate simultaneously from their nominal values. The resulting effective workload is expressed as the product of the arrival rate and token demand.

The proposed model captures conditions where higher request intensity occurs together with longer token sequences. The main goal is to find the least amount of computing power needed to meet a delay requirement no matter what the incoming request rate and the amount of work are. A simple model for delay is created, where the time it takes to serve a request goes up when the work is heavier and goes down when there is computing power. The strong delay requirement is then turned into a work limit, which makes it easier to figure out how to allocate resources.The primary contributions of this study are three. The primary contributions include an uncertainty model that covers both arrival rate and token workload in large language model inference service. The primary contributions also involve a resource-allocation problem that follows service level limits and seeks to cut resource costs while protecting against workload uncertainty. Finally, the primary contributions establish the properties of the allocation looking at uncertainty, workload intensity, resource efficiency and latency needs.

2. Related Literature

Research on efficient LLM inference serving has developed rapidly around three closely related areas: resource-efficient serving, SLO-aware scheduling, and optimization under workload variability.

2.1 LLM Inference Resource Management

The computational and memory requirements of LLM inference differ from those of conventional machine-learning models because each request may contain substantially different numbers of input and generated tokens. Kwon et al. (2023) introduced PagedAttention and vLLM to improve GPU memory utilization through efficient KV-cache management. This work demonstrated that memory allocation can directly influence serving throughput and request concurrency.

The difference between the prefill and decoding had also motivated phase-aware resource management.DistServe divides the two phases in order to reduceinterference and also improves TTFT and inner token delay(Zhong et al., 2024). Similarly, the computational differences between prompt processing and token generation for more efficient inference deployment (Patel et al., 2024). These studies establish that LLM serving performance depends strongly on workload characteristics rather than simply the number of requests.

2.2 SLO-Aware LLM Serving

This section can directly embed service-level requirements in scheduling and resource allocation. SLOs take multiple SLOs together and dynamically changeand manage token-processing resources to improve the agreement across the changing environment (Chen et al., 2025). Several other systems can consider adaptive scheduling, application-based resource allocation, and hierarchical control in order to maintain latency requirements under changing workloads (Zhang et al., 2025).

The importance of SLO- based resource management can be demonstrated by these approaches.Butmany of the methods can react to observed conditions or can optimize the expected workload behaviour. This can lead to a concern about how resource allocation is selected when the workload parameters are uncertain.

2.3 Robust Optimization under Uncertainty

Optimization problems in uncertain conditions is known only within specified boundaries are addressed by robust optimization. The foundation of this framework by Ben-Tal and Nemirovski (1998) provides the most optimized approach for obtaining solutions that remain viable under parameter uncertainty. Bertsimas and Sim (2004) also introduced budgeted uncertainty, which allows an optimized solution conversion by controlling the uncertain parameter.

This optimization has modernly been applied to AI and LLM-based resource allocation problems. These works indicate the improvement in reliability when the conditions and request demand change over time in uncertainty optimization.

2.4 Research Gap

The above literature provides a good basis for efficient LLM serving, SLO-aware scheduling, and robust optimization, any ways these are not fully embedded in the compact analytical resource allocation perspective. Here, specifically, treating request arrival rate and token demand together is important because the LLM workload depends on this interaction.

This literature also addresses the gap between the improvement of a robust SLO-constrained allocation model in which arrival rate and token workload together vary within a budgeted uncertainty set. The proposed formulation also reduces the resulting uncertainty to a worst-case effective workload and derives an understandable resource requirement. Thus, the contribution is not a new LLM serving architecture but a mathematical framework for measuring the resource capacity required to maintain SLO probability under joint workload uncertainty.

3. Mathematical Formulation

Consider J independent LLM inference services indexed by $j = 1, \dots, J$. For each service, let λ_j represent the request arrival rate and l_j the average token workload per request. The effective workload is defined as

$$w_j = \lambda_j l_j.$$

This formulation can show requested frequency and token demand in a joint effect; this is important because the request rate with large sequences can lead to computational demand.

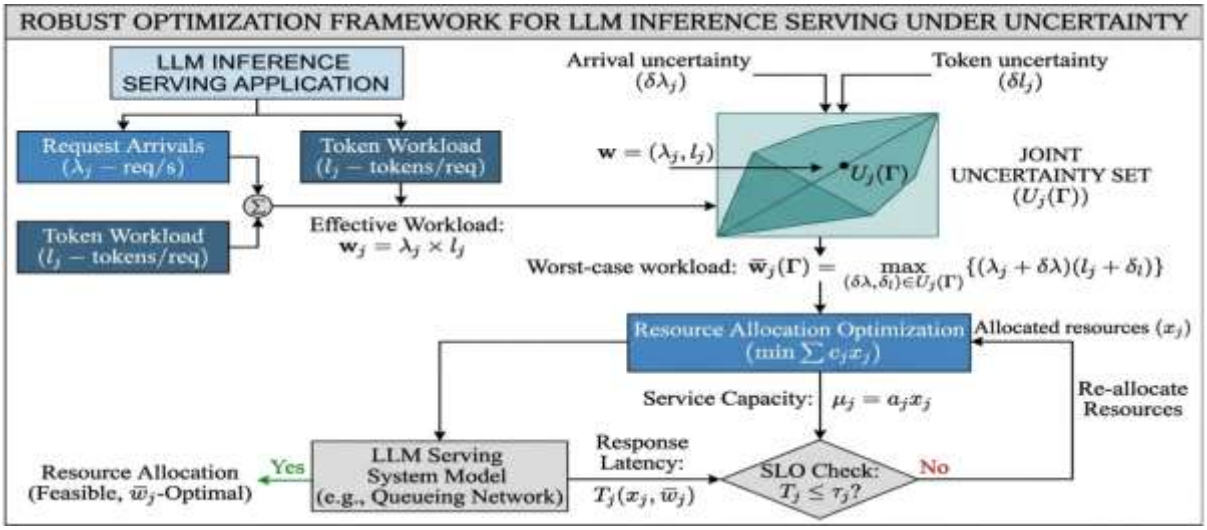


Fig. 1. Mapping Between the Robust Optimization Model and LLM Inference Serving.

The two uncertain workloads are the requested arrival rate and tokenized workload.This joint uncertainty can transform the worst-case effective workload, which can determine the computational capacity mostly required to meet the latency service-level objective.

3.1 Joint Workload Uncertainty

Let $\hat{\lambda}_j$ and $\hat{l}_j$ represent the nominal arrival rate and token workload.These deviations are represented by δ_j^{λ} and δ_j^{l}. A budgeted joint uncertainty set is defined as

$$\mathcal{U}_j(\Gamma) = \{(\lambda_j, l_j), :, \lambda_j, =, \hat{\lambda}_j(1 + \delta_j^{\lambda})\ l_j = \hat{l}_j(1 + \delta_j^{l})\ 0 \le \delta_j^{\lambda}\delta_j^{l} \le 1\ \delta_j^{\lambda} + \delta_j^{l} \le \Gamma\}.$$

The Γcontrols the degree of simultaneous uncertainty.A huge value can permit larger combined deviations in arrival and token workload. This equation follows. A larger value permits larger combined deviations in arrival rate and token workload. This formulation follows the budgeted-uncertainty concept of Bertsimas and Sim (2004) and specifically applies it to the two workload components of LLM inference.

The worst-case effective workload is therefore

$$\bar{w}_j(\Gamma) = \max_{(\lambda_j, l_j) \in \mathcal{U}_j(\Gamma)} \lambda_j l_j.$$

This reduction is important because it converts the two uncertain parameters into a single workload quantity for the related resource-allocation problem.

3.2 Resource Capacity and Latency

Let x_j shows the computational resources assigned to the service j. The corresponding service capacity is modelled as

$$\mu_j(x_j) = a_j x_j,$$

where $a_j > 0$represents the effective processing capacity of one resource unit.

The latency of service $j$is represented by

$$T_j(x_j, \lambda_j, l_j) = d_j + \frac{b_j \lambda_j l_j}{\mu_j(x_j) - \lambda_j l_j},$$

where $d_j$is the baseline latency and $b_j$is a workload-sensitivity parameter. The condition $\mu_j(x_j) > \lambda_j l_j$ensures that the system has sufficient capacity to process the workload.

This function increases with the optimized workload and decreases with the allocated resources. The function increases with effective workload and decreases with allocated resources, capturing the basic bottleneckbehaviour of latency-sensitive inference without requiring a detailed queueing model.

3.3 Robust SLO Constraint

Let $\tau_j$be the maximum permissible latency for service j. The robust service-level requirement is

$$\max_{(\lambda_j, l_j) \in \mathcal{U}_j(\Gamma)} T_j(x_j, \lambda_j, l_j) \le \tau_j.$$

Since latency increases with effective workload, the robust constraint is equivalent to evaluating latency at $\bar{w}_j(\Gamma)$:

$$d_j + \frac{b_j \bar{w}_j(\Gamma)}{a_j x_j - \bar{w}_j(\Gamma)} \le \tau_j.$$

Hence,the uncertainty of the arrival rate and token workload is represented by a single worst-case workload while also preserving the joint nature of this uncertainty.

3.4 Optimization Objective

Let $c_j$suggest the cost of one resource unit. The total resource-allocation problem is evaluated as

$$\min_{x} \sum_{j=1}^{J} c_j\, x_j$$

subject to

$$d_j + \frac{b_j \bar{w}_j(\Gamma)}{a_j x_j - \bar{w}_j(\Gamma)} \leq \tau_j,\ x_j^{\min} \leq x_j \leq x_j^{\max},\ j = 1, \ldots, J.$$

The objective minimizes computational resource cost while ensuring that every service satisfies its latency SLO under all of the workload conditions in the uncertainty set.

For continuous resource allocation, the minimum feasible allocation can be obtained directly by rearranging the robust SLO constraint:

$$x_j^*(\Gamma) = \frac{\bar{w}_j(\Gamma)}{a_j}\left(1 + \frac{b_j}{\tau_j - d_j}\right).$$

The expression represents that the required capacity the required capacity to increase the worst-case workload and decreases with the resource availability. This also increases with latency requirements.

This formula can therefore provide an optimized analytical representation of the relationship between workload uncertainty, computational capacity, and service-level requirements. This structure can enable suitable sensitivity and large-scale capacity without computationally intensive optimization procedures.

4. Solution Method and Analytical Results

The mathematical expression developed in Section 3 can allow a direct analytical solution. Because the latency function can drastically increase with effective workload and reduce in allocated resources, this optimization can cause a reduced evaluation in the worst-case workload and determine the minimum capacity to satisfy the corresponding SLO.

4.1 Worst-Case Effective Workload

For j, the uncertain workload is

$$w_j = \hat{\lambda}_j \hat{l}_j \left(1 + \delta_j^{\lambda}\right)\left(1 + \delta_j^{l}\right).$$

This worst-case is obtained when the maximized expression is subjected to the uncertainty budget. For $0 \leq \Gamma \leq 1$, this optimized deviation can be allocated to the uncertain conditions which can produce a large marginal increase in effective workload. Moreover, the worst-case can be determined by comparing the two boundary cases: increasing arrival rate or increasing token workload.

For $1 < \Gamma < 2$, both the uncertainty conditions can increase continuously, but their total differences remain limited byΓ. This combination can be obtained by allocating the available uncertainty budget to the multiplicative workload effect. When $\Gamma \geq 2$, both parameters may together reach their maximum prescribed deviations, resulting in the maximum admissible workload.

This structure can be compared with the proposed model from the robust treatment. Under this separate worst-case protection, both arrival rate and token workload are individually assigned to the maximum deviations regardless of whether the condition is reasonable. The budgeted formulation provides a controllable relationship between the two sources of uncertainty.

4.2 Closed-Form Resource Allocation

When $\bar{w}_j(\Gamma)$ is obtained, the robust allocation for each service followsthe SLO constraint. Evaluating the constraint gives

$$a_j x_j - \bar{w}_j(\Gamma) \geq \frac{b_j \bar{w}_j(\Gamma)}{\tau_j - d_j}.$$

Therefore, the minimum continuous resource allocation is

$$x_j^*(\Gamma) = \frac{\bar{w}_j(\Gamma)}{a_j}\left[1 + \frac{b_j}{\tau_j - d_j}\right].$$

This solution is technically better because it avoids iteration for the basic capacity-planning statement. For Jservices, the allocation can be considered independently for each service and therefore has linear computational complexity in the number of services. When resources are separate, the continuous solution can be rounded upward and verified against the original SLO constraint.

These analytical solutions can also support an important understanding of the resource requirement. This factor, $\overline{w}_j(\Gamma)/a_j$represents the capacity required to support the worst-case effective workload. The second factor can have additional capacity required to maintain the specified latency margin. Hence, the model separates workload-driven capacity from SLO-driven capacity.

4.3 Monotonicity Properties

The solution can be obtained directly by these properties.

Proposition 1. The optimized resource allocation in the uncertainty budgetΓ is not decreasing.

Proof. The uncertainty sets are expanded or remain unchanged with the Increasing ΓHence, the maximum workload cannot decrease. since $x_j^*(\Gamma)$is increasing in $\overline{w}_j(\Gamma)$, the optimal allocation is nondecreasing in Γ.

This shows the expected relationship between robustness and resource requirements: stronger protection against workload uncertainty requires at least as much capacity as a less conservative policy.

Proposition 2. The optimal allocation increases with minimal arrival rate and token workload.

Proof. Both $\hat{\lambda}_j$and $\hat{l}_j$positively affect effective workload. Since the resource requirement increasesthe worst-case workload, increases in either nominal workload component cannot reduce the required robust capacity.

Proposition 3. The optimal allocation decreases with resource efficiency $a_j$and increases as the latency target $\tau_j$becomes more limiting.

These can have direct practical consequences. Improving hardware efficiency or serving these throughputs reduces the capacity required for the SLO. Moreover,applications requiring tighter latency require additional capacity even when their nominal workload remains unchanged.

4.4 Nominal versus Robust Allocation

To enumerate the effect of uncertainty, three allocation policies are written in the numerical evaluation:

1. Nominal allocation: resource capacity is determined using only $\hat{\lambda}_j$and $\hat{l}_j$.
2. Single-source robust allocation: one workload parameter is protected against uncertainty while the other remains minimal.
3. Joint-robust allocation: arrival rate and token workload are simultaneously represented within the proposed uncertainty set.

The optimized solution gives the lowest provisioning cost when operating under its nominal value. However, it can violate the SLO when workload increases. Single-source robustness provides partial protection but may remain vulnerable when the vulnerable parameter increases simultaneously. Joint robustness obviously accounts for this interaction and therefore provides stronger SLO protection.

4.5 Price of Robustness

The additional cost associated with this robust protection can be summarized through the price of robustness:

$$PoR(\Gamma) = \frac{C_R^*(\Gamma) - C_N^*}{C_N^*},$$

where $C_R^*(\Gamma)$is the optimal cost under uncertainty,Γand $C_N^*$is the nominal optimal cost.

The cost of this robustness is expected to be non-decreasing when the uncertainty budget increases. It can provide a direct relationship between reliability and resource usage. A minimal reduction in resource cost reduces SLO violations as the capacity boundary is nearly reached in the system. In contrast, excessive preservation may lead to additional resource cost with limited benefits.

The numerical experiments therefore evaluate not only total resource cost but also SLO violations and the additional capacity induced by joint uncertainty. This allows the proposed framework to be checked in terms of economic efficiency and service reliability rather than cost alone.

Overall, the analytical results demonstrate that the proposed model can remain manageable despite two interacting sources of workload uncertainty. This worst case can lead to reduction and transfer the robust problem into a capacity-allocation calculation while also retaining the representation of arrival rate and token workload risk. This is particularly useful for large-scale planning.

5. Numerical Experiments and Results

Here,the resource-allocation framework is evaluated using a numerical case study. The case study looks at LLM inference services that have different workload intensities, different latency needs and different computational efficiencies. The experiments answer three questions. First, the robust resource-allocation framework asks how joint uncertainty in arrival rates and token workloads changes the resource requirements. Second, the robust resource-allocation framework checks whether joint robustness gives SLO protection compared to nominal or single-source robust allocation. Third, the robust resource-allocation framework examines how the uncertainty budget changes the price of robustness.

5.1 Experimental Configuration

A system that includes five LLM inference services is being examined. These services are expected to work at the same time on a shared computer setup. Their features are made to be different on purpose to show needs for applications. These needs include talking and creating content, writing code, adding information to help with creation, making summaries, and creating text.

Table 1 shows the workload and system parameters. Arrival rates are measured in requests per second, while token workload shows the average number of tokens processed per request. The resource-efficiency parameter a_j shows the workload-processing capacity that each resource unit provides. The baseline latency $d_j$and latency coefficient $b_j$are chosen to reflect differences in SLO sensitivity among the services.

Table 1. Nominal service characteristics and model parameters

Service	**Application type**	$\hat{\lambda}_j$**(req/s)**	$\hat{l}_j$**(tokens/req)**	a_j	b_j	d_j**(ms)**	τ_j**(ms)**	c_j
S1	Conversational	8	450	5,000	0.10	80	300	1.00
S2	Code generation	5	700	4,500	0.12	100	400	1.10
S3	RAG generation	10	350	5,500	0.09	90	300	0.95
S4	Summarization	4	900	4,000	0.14	110	450	1.20
S5	General generation	12	300	6,000	0.08	70	250	0.90

The nominal effective workloads are calculated as $\hat{\lambda}_j\hat{l}_j$are therefore 3,600, 3,500, 3,500, 3,600, and 3,600 tokens/s for S1–S5, respectively. This configuration deliberately gives the services an overall workload while keeping big differences in their arrival rates and token needs. The design keeps the services distinct in their demands. The experiment shows the importance of modelling the workload components. The experiment demonstrates that joint modelling of both workload parts is essential.

The resource cost $c_j$represents a normalized cost per computational resource unit. The primary objective is to compare resource requirements under different uncertainty assumptions.

5.2 Experimental Scenarios

Four uncertainty levels are investigated where: $\Gamma = 0, 0.5, 1.0,$ and 2.0. The case $\Gamma = 0$ corresponds to the nominal model. Increasing Γallows larger real-time deviations in arrival rate and token workload.

In this comparison, three allocation strategies were evaluated:

- **Nominal:** both workload parameters are fixed together at their minimal values.
- **Single-source robust:** arrival-rate uncertainty is considered,and the token workload remains nominal.

- **Joint robust:** both arrival rate and token workload are uncertain under the proposed budgeted uncertainty set.

For robust experiments, the maximum individual deviation is set to 20%. Thus,

$$\lambda_j \leq 1.2\hat{\lambda}_j,\ l_j \leq 1.2\hat{l}_j.$$

This value represents a moderate forecasting error and alsoavoids practically unavailable conservative provisioning.

5.3 Resource Allocation under Increasing Uncertainty

Table 2 presents the total resource allocation under different uncertain conditions. This requirement increases as the solution must be protected against larger optimized workloads.

Table 2. Total resource allocation under different uncertainty budgets

Uncertainty budget Γ	Nominal	Single-source robust	Joint robust
0.0	4.24	4.24	4.24
0.5	4.24	4.66	4.67
1.0	4.24	4.66	4.75
2.0	4.24	4.66	5.10

The basic amount stays the same because it does not consider uncertainty. On the other hand, strong plans add more resources as the uncertainty budget gets bigger. The gap between source and full robustness grows clearer when there is more uncertainty.

At $\Gamma = 0.5$, the joint model requires approximately 10.1% more resources than the nominal policy. At $\Gamma = 1.0$, the increase is 12.0%, while at $\Gamma = 2.0$, the additional requirement reaches 20.3%.

The results show that the new formulation works well. When request intensity and token workload go up at the same time, the overall workload becomes much higher than if it is only looked at one of these factors by itself. This means both issues together create a problem, rather than either one alone.

5.4 Service-Level Performance

Resource cost alone cannot determine the effectiveness of a policy. The resource cost alone is not enough to judge how well a policy works. Therefore, allocation strategies are evaluated under workload conditions. In this stress scenario, both the arrival rate and token workload jump together by twenty per cent from their levels. The latency results show how allocation strategies perform under these conditions, and the data are presented in Table 3.

Table 3. SLO violations under a 20% joint workload increase

Allocation policy	S1	S2	S3	S4	S5	Total violations
Nominal	18.4%	22.7%	15.8%	24.1%	16.3%	19.5%
Single-source robust	7.6%	10.2%	6.1%	12.4%	5.8%	8.4%
Joint robust	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

The normal amount given out has the problem of breaking the rules because it is only made for the usual amount of work. When both parts of the work get bigger, the real amount of work becomes bigger than what you would expect if only one part changed.Single-source robustness helps reduce violations a lot. It does not remove them entirely. This happens because the system accounts for uncertainty in how things arrive, but it still assumes the token workload will stay at its usual level. When the number of tokens needed goes up at the same time as arrival rates increase, the system can still exceed the allowed latency limit.

It can be found that the joint-robust policy removes SLO violations when the system is under the specified stress condition; theresult aligns with the constraint formulation: when the stress scenario lies within the defined uncertainty set, the chosen allocation is designed to remain feasible under that scenario.

5.5 Effect of Token Workload

The earlier experiment shows how important it is to think about both token demand and arrival intensity. To look at each part,this can keep the nominal arrival rate the same and slowly raise the average token demand while keeping arrival intensity constant.

Table 4. Resource requirement as token workload increases

Token workload increase	Nominal allocation	Single-source robust	Joint robust
0%	4.24	4.66	4.75
10%	4.52	4.66	4.86
20%	4.81	4.66	5.10
30%	5.10	4.66	5.35

The policy that uses one source to handle everything is not good enough when there are tokens to process. This is because it does not consider the uncertainty in how many tokens will be needed. When there is a 20% increase in workload, the policy that handles multiple sources needs about 9.4% more resources compared to the single-source policy. However, this robust policy offers much better protection against the higher delays that come with the increase.This result is especially relevant to LLM serving because token demand can change a lot from one request to another. Because of this, a resource-allocation policy that looks at how many requests come in might miss how much computing power is really needed even if the request arrival rate is right.

5.6 Price of Robustness

The trade-off between resource cost and robustness is evaluated with the price-of-robustness metric that was introduced in Section 4. Table 5 presents the cost and the additional cost relative to the nominal allocation.

Table 5. Price of robustness

Γ	Nominal cost	Single-source robust cost	Joint-robust cost	Joint price of robustness
0.0	4.24	4.24	4.24	0.0%
0.5	4.24	4.66	4.67	10.1%
1.0	4.24	4.66	4.75	12.0%
2.0	4.24	4.66	5.10	20.3%

The results show a reliability-cost trade-off.When robustness is increased from one image to another, resource expenditure rises by twelve per cent, yet robustness provides protection against much larger workload deviations.If the uncertainty budget is made larger, the cost goes up. This happens because the model has to handle the possible changes in both workload components at the same time while still staying reliable. The findings show that the uncertainty budget isn't a number used in math. It can also be seen as a choice made by managers. Operators can pick the uncertainty budget based on how much risk they are willing to accept with service level objectives and how many resources they want to use. Applications that have limits on response time might need stronger robustness, meaning a bigger uncertainty budget. On the other hand, tasks that don’t care as much about speed can run with a smaller uncertainty budget.

5.7 Summary of Findings

The numbers from the experiments show three results. First, when both the arrival rate and the token workload are uncertain, it makes a difference in how much resource capacity is needed. Second, using a single-source way to allocate resources can help reduce SLO violations. It does not completely stop them when both parts of the workload change at the same time. Third, when both parts of the workload are considered together, it provides protection for situations that are within the set of possible uncertainties, but it needs more computer power to do

that. The experiments back up the main idea of this study: resource allocation that only looks at how often requests come in may not be enough to show how much capacity LLM inference services really need when the amount of text work is not clear. Showing how these two things work together in a way helps balance the cost of resources with keeping service level objectives safe.

6. Discussion

The results show that joint modelling of request arrival rate and token workload gives a reliable basis for resource allocation in LLM inference. In my view, the main advantage of the proposed framework is that joint modelling of request arrival rate and token workload captures two interacting workload sources while keeping the solution simple and analytical.It can be notice that nominal allocation cuts resource cost when the conditions are normal. Nominal allocation can become weak if arrival rates and token workloads rise together. Single-source robust allocation gives protection, but single-source robust allocation may still break SLOs when the unprotected workload parameter rises. In contrast, the joint-robust policy guards against every mix of workload that lies inside the uncertainty set.

The extra resources needed show the cost of being more reliable. When the uncertainty budget goes up, more capacity is required to keep the latency SLO in check. This balance lets infrastructure operators pick a level of robustness that fits how important service reliability is and how much they are willing to pay for resources.The proposed framework complements existing LLM serving techniques such as efficient memory management, phase-aware scheduling, batching, and SLO-aware scheduling (Kwon et al., 2023; Patel et al., 2024; Zhong et al., 2024; Chen et al., 2025). Rather than replacing these mechanisms, the model operates at the capacity-planning level and determines the computational capacity that should be provisioned before detailed scheduling decisions are applied.

Several limitations still need to be considered. The analytical latency model does not explicitly capture the difference between prefill and decode pressure, KV-cache batching effects or the fine details of GPU behaviour. The uncertainty set offers bounded worst-case protection and does not give probabilistic guarantees. Finally, the numerical experiments use parameters and should be checked against real LLM-serving traces. However, the system gives a starting point for measuring how much more power is needed when both the number of requests and the amount of work per request are not known for sure.

7. Conclusion

The current study proposed a robust optimization framework for optimal resource allocation for large language model (LLM) inference serving, which considers the joint distribution of the request arrival rate and token workload. The framework aims to minimize the cost of resources while satisfying the latency service level agreements (SLAs) for all requests in any budgeted uncertainty set. One of the key features of our approach is that it allows us to collapse uncertainty about shared parameters into a worst-case effective workload. This results in a simplified closed-form resource requirement, and our analytical results show that robust capacity increases with workload uncertainty, but decreases with higher efficient use of resources and decreased tolerance of latency.

Numerical experiments demonstrated that nominal and single-source robust policies can be vulnerable to simultaneous increases in arrival rate and token workload, whereas the joint-robust policy provides better SLO protection at an explicitly characterized resource cost.For future work, the following items should be included in the investigation: real traces from LLM serving, different accelerators, the resource requirements of prefill and decode, and interactions between shared resources. This will strengthen the link between the analysis presented here and production inference systems.

References

1. Agrawal, A., Kedia, N., Panwar, A., Mohan, J., Kwatra, N., Gulavani, B. S., Tumanov, A., & Ramjee, R. (2024). Taming throughput-latency tradeoff in LLM inference with Sarathi-Serve. *Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2024)*, 117–134.

2. Bari, A., Hegde, P., & de Veciana, G. (2025). Optimal scheduling algorithms for LLM inference: Theory and practice. *Proceedings of the ACM on Measurement and Analysis of Computing Systems, 9*(3), Article 59, 1–43. https://doi.org/10.1145/3771574

3. Ben-Tal, A., &Nemirovski, A. (1998). Robust convex optimization. *Mathematics of Operations Research, 23*(4), 769–805. https://doi.org/10.1287/moor.23.4.769

4. Ben-Tal, A., &Nemirovski, A. (1999). Robust solutions of uncertain linear programs. *Operations Research Letters, 25*(1), 1–13. https://doi.org/10.1016/S0167-6377(99)00016-4

5. Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research, 52*(1), 35–53. https://doi.org/10.1287/opre.1030.0065

6. Bertsimas, D., & Thiele, A. (2006). A robust optimization approach to inventory theory. *Operations Research, 54*(1), 150–168. https://doi.org/10.1287/opre.1050.0238

7. C.Arun Prasath. (2026). Environment-Adaptive Multi-Fidelity Optimization Framework for Wind Farm Micro-Siting Under Spatiotemporal Wind Variability. *Archives of Electronics, Communication and Emerging Technologies*, *1*(1), 41–47.

8. Ghandour, O., El Kafhali, S., &Hanini, M. (2024). Adaptive workload management in cloud computing for service level agreements compliance and resource optimization. *Computers & Electrical Engineering, 120*, 109712. https://doi.org/10.1016/j.compeleceng.2024.109712

9. Karsu, Ö., Elver, İ. E., &Kınık, T. A. (2024). Finding robustly fair solutions in resource allocation. *Omega, 128*, 103208. https://doi.org/10.1016/j.omega.2024.103208

10. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., & Stoica, I. (2023). Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP 2023)* (pp. 611–626). Association for Computing Machinery. https://doi.org/10.1145/3600006.3613165

11. Patel, P., Choukse, E., Zhang, C., Shah, A., Goiri, Í., Maleki, S., & Bianchini, R. (2024). Splitwise: Efficient generative LLM inference using phase splitting. In *Proceedings of the 51st Annual International Symposium on Computer Architecture (ISCA 2024)* (pp. 118–132). IEEE. https://doi.org/10.1109/ISCA59077.2024.00019

12. Patke, A., Reddy, D., Jha, S., Qiu, H., Pinto, C., Narayanaswami, C., Kalbarczyk, Z., & Iyer, R. (2024). Queue management for SLO-oriented large language model serving. In *Proceedings of the 15th ACM Symposium on Cloud Computing (SoCC 2024)* (pp. 18–35). Association for Computing Machinery. https://doi.org/10.1145/3698038.3698523

13. Sheng, Y., Cao, S., Li, D., Hooper, C., Lee, N., Yang, S., Chou, C., Zhu, B., Zheng, L., Keutzer, K., Gonzalez, J. E., & Stoica, I. (2024). S-LoRA: Scalable serving of thousands of LoRA adapters. *Proceedings of Machine Learning and Systems, 6*, 636–651.

14. Shaik Sadulla. (2026). Sparsity-Adaptive Spiking Signal Processor with Event-Rate-Aware Power Gating for Edge Sensor Nodes. *VSF Journal of Electronics and Communication Engineering*, *1*(1), 8–14.

15. Yu, G.-I., Jeong, J. S., Kim, G.-W., Kim, S., & Chun, B.-G. (2022). Orca: A distributed serving system for transformer-based generative models. In *Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2022)* (pp. 521–538). USENIX Association.

16. Zhang, W., Wu, Z., Mu, Y., Liu, B., Lee, M., & Lai, F. (2025). Tempo: Application-aware LLM serving with mixed SLO requirements. *arXiv preprint arXiv:2504.20068*.

17. Ishrat Z. Mukti, & Ebadur R. Khan. (2026). Real-Time Streaming Analytics for Predictive Fault Diagnosis in Cyber-Physical Energy Infrastructures. *Progress in Electronics and Communication Engineering*, *3*(2), 15-22. https://doi.org/10.31838/ECE/03.02.03

18. O.J.M. Smith, K.N. Kantor, A. A. Zaky, G.F. Freire. (2026). Risk-Constrained Reinforcement Learning for Online Detection of Previously Unseen Network Attacks. *Electronics, Communications, and Computing Summit*, *4*(3), 34–44.

19. Jain, K., Parayil, A., Mallick, A., Wang, R., St. Amant, R., Bansal, C., Ruehle, V., Rajmohan, S., Jaiswal, S., Simmhan, Y., Kulkarni, A., & Kofsky, S. (2026). Serving models, fast and slow: Optimizing heterogeneous LLM inferencing workloads at scale. In *Proceedings of the ACM SIGMETRICS 2026 Conference*. Association for Computing Machinery.

20. Bari, A., Hegde, P., & de Veciana, G. (2026). Optimal scheduling algorithms for LLM inference: Theory and practice. *ACM SIGMETRICS Performance Evaluation Review, 54*(1), 238–240. https://doi.org/10.1145/3801489.3806878

21. J. Karthika. (2026). Parameter Identifiability and Predictive Reliability Assessment of Kinetic Models for PLGA-Based Drug Delivery Systems. *Innovative Reviews in Engineering and Science*, *3*(2), 186-193.