Win Fast or Lose Slow: Balancing Speed and Accuracy in Latency-Sensitive Decisions of LLMs

Hao Kang
Georgia Institute of Technology
hkang342@gatech.edu

Qingru Zhang
Georgia Institute of Technology
qzhang441@gatech.edu

Han Cai
NVIDIA Corporation
hcai.hm@gmail.com

Weiyuan Xu
University of California, Berkeley

Tushar Krishna
Georgia Institute of Technology
tushar@ece.gatech.edu

Yilun Du
Harvard University
yilundu@gmail.com

Tsachy Weissman
Stanford University
tsachy@stanford.edu

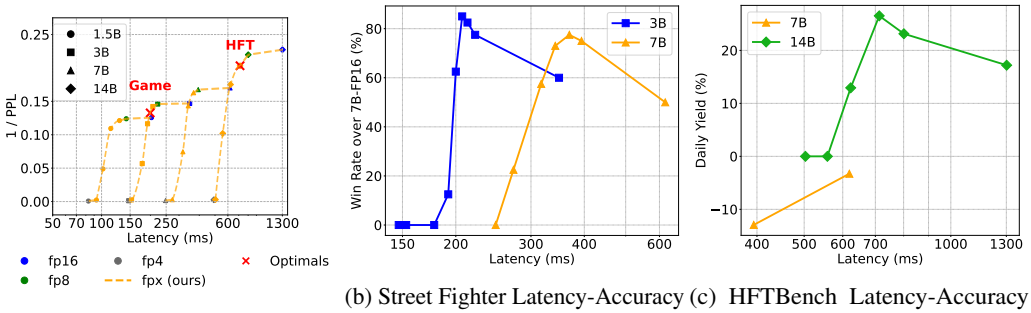
Abstract

Large language models (LLMs) have shown remarkable performance across diverse reasoning and generation tasks, and are increasingly deployed as agents in dynamic environments such as code generation and recommendation systems. However, many real-world applications, such as high-frequency trading and real-time competitive gaming, require decisions under strict latency constraints, where faster responses directly translate into higher rewards. Despite the importance of this latency–quality trade-off, it remains underexplored in the context of LLM-based agents. In this work, we present the first systematic study of this trade-off in real-time decision-making tasks. To support our investigation, we introduce two new benchmarks: **HFTBench**, a high-frequency trading simulation, and **StreetFighter**, a competitive gaming platform. Our analysis reveals that optimal latency–quality balance varies by task, and that sacrificing quality for lower latency can significantly enhance downstream performance. To address this, we propose **FPX**, an adaptive framework that dynamically selects model size and quantization level based on real-time demands. Our method achieves the best performance on both benchmarks, improving win rate by up to **80%** in Street Fighter and boosting daily yield by up to **26.52%** in trading, underscoring the need for latency-aware evaluation and deployment strategies for LLM-based agents. These results demonstrate the critical importance of latency-aware evaluation and deployment strategies for real-world LLM-based agents. Our benchmarks are available at Latency Sensitive Benchmarks.

1 Introduction

Large language models (LLMs) exhibit remarkable performance across various natural language processing (NLP) tasks and artificial intelligence (AI) applications, ranging from text generation to complex reasoning [OpenAI, 2023, Abdin et al., 2024, Team et al., 2025]. Beyond their standalone use, LLMs can be integrated into agent frameworks, enabling more sophisticated behaviors such as decision-making, multi-step reasoning, and planning [Yao et al., 2023, Shinn et al., 2023, Li et al., 2023, Du et al., 2023]. In these settings, a LLM acts as a decision-making agent, generating its actions or responses and then receiving feedback or rewards from environment. Many of these

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) Latency-Accuracy trade off

(b) Street Fighter Latency-Accuracy pareto curve

(c) HFTBench Latency-Accuracy pareto curve

Figure 1: Latency–accuracy trade-offs across different model configurations and tasks. (a) FPX enables a smooth and continuous trade-off between latency and accuracy, allowing models to meet diverse task-specific requirements. (b) In the Street Fighter benchmark, win rate first increases as latency decreases, peaking at a Pareto-optimal point, before dropping due to excessive accuracy loss. (c) Observation in HFTBench: daily yield improves with moderate latency reduction, but degrades when model accuracy is overly compromised.

agent tasks exhibit a high tolerance for inference latency, where slow responses are acceptable as long as the output quality remains high. Examples include code generation [Zhuo et al., 2024], mathematical problem solving [Xiao et al., 2023], and product recommendation [Wang et al., 2023], where correctness and completeness are prioritized over speed.

However, there is a different large class of real-time tasks that are highly sensitive to response latency and remains largely unexplored. These tasks often take place in dynamic environments that evolve continuously over time and are influenced by the agent's actions. In such settings, response latency becomes a critical factor in an agent's overall performance. Fast and well-timed actions are essential for obtaining positive rewards, while delays often leads to missed opportunities or suboptimal outcomes. One prominent example is gaming. Competitive games, such as Street Fighter [Su, 2010] and StarCraft [Samvelyan et al., 2019], take place in real-time environments where agents must perform multiple actions in a timely manner to win. Faster agents are more likely to stay synchronized with environmental changes and maintain an advantage, while slower agents may fall behind while processing outdated observations. Similarly, robotic control in dynamic physical environments demands rapid perception–action loops. Delayed responses in such settings, especially in high-stakes applications like autonomous driving, can result in unsafe or incorrect behavior.

Another important example is using LLM agents for high-frequency financial trading [He and Lin, 2022], where both low latency and high response quality are crucial. Stock exchanges match transactions based on real-time order flow, and faster trading agents can exploit arbitrage opportunities by acting before competitors respond. Prior research [Baron et al., 2019, He and Lin, 2022] in finance shows that trading latency directly impacts earning yields, motivating investment institutions to heavily invest in low-latency methods.

Across all these examples, both inference latency and response quality are critical for LLM agents to achieve strong performance. Either delayed or low-quality actions can lead to performance degradation or outright task failure. However, a fundamental trade-off exists between latency and quality when choosing models of different sizes or compressing them to low precisions. As illustrated by Figure 1a, larger models typically generate higher-quality outputs but suffer from longer inference time, while smaller or highly compressed models offer faster inference speed at the expense of reduced output quality. Therefore, as shown in Figures 1b and 1c, there exists an optimal solution for this trade-off and effectively balancing this trade-off is essential to optimize model performance in this type of real-world tasks.

In this paper, we are the first to systematically formulate and investigate the latency–quality trade-off in the context of real-time decision-making by LLM agents. We define this class of tasks as *latency-sensitive agent decision tasks*, where both output quality and response latency jointly determine the agent's overall performance. To evaluate model performance in such latency-sensitive setting, we develop two novel real-time evaluation benchmarks: (i) HFTBench: a high-frequency trading system tailored to assess real-time trading decisions of LLMs; (ii) StreetFighter: a competitive gaming platform that evaluate real-time gaming decisions of LLMs.

Based on our benchmarks, we observe that different tasks exhibit varying sensitivities to inference latency and output quality. As shown in Figure 1b and 1c, StreetFighter is more latency-sensitive and less quality-sensitive – timely, even if suboptimal, actions often lead to winning outcomes due to the game's simple yet rapidly evolving dynamics. In contrast, trading tasks demand both high quality and low latency. Inaccurate decisions can result in significant financial losses, making response accuracy as crucial as speed. Confronting such diverse task requirements, it is inherently challenging to identify the optimal point along the latency–quality trade-off. Existing approaches, such as selecting among fixed model sizes or applying static low-precision quantization, typically offer only a limited set of discrete options, which fail to capture the fine-grained trade-offs required across real-world tasks. To enable fine-grained searching, we introduce FPX, an adaptive mixed-precision inference framework that enables flexible control over inference latency while minimizing quality degradation. FPX jointly adjusts model size and dynamically mixes inference bitwidths across model layers to meet any specified latency target. Specifically, it hybridizes FP8 and FP4 inference kernels by selectively applying lower precision (FP4) to compression-tolerant layers while preserving FP8 for more sensitive components. This progressive and targeted quantization approach allows FPX to achieve continuous, fine-grained control across the latency–quality trade-off, effectively minimizing performance loss while satisfying diverse latency requirements. Our contributions are as follows:

- **Latency–Quality Trade-off.** We are the first to systematically formulate and investigate the latency–quality trade-off in the context of *latency-sensitive agent decision tasks*.
- **Latency-Sensitive Evaluation Benchmarks:** We introduce two novel benchmarks for evaluating LLM performance in the latency-sensitive settings: (1) a high-frequency trading (HFT) system specifically tailored to LLMs, and (2) a competitive fighting game environment based on *Street Fighter* from the DIAMBRA platform [Palmas, 2022].
- **Adaptive Mixed Precision Inference Framework.** We propose an adaptive mixed-precision inference framework that that enables flexible control over inference latency while minimizing quality degradation.

2 Background

2.1 Low Precision Inference to Reduce Latency

Recent advancements in hardware-supported low-precision inference, such as FP8 and FP4 [Micikevicius et al., 2022, Li et al., 2025], offer significant improvements in both throughput and latency over standard full-precision inference (FP16). These methods employ floating point quantization (FP Quant) to map high-precision tensors to low-precision ones, reducing memory footprint of both model weights and activations [Li et al., 2025]. Given a tensor X, FP quantization rescales its entries and rounds them to values within a bounded range determined by bitwidth b:

$$Q(X) = \text{round}\left(\frac{X}{\text{scale}_X}\right), \quad \text{scale}_X = \begin{cases} \frac{\max(|X|)}{\text{range}_b} & \text{if } \max(|X|) > \text{range}_b \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

Here, $Q(X)$ is the quantized matrix and range_b is determined by the bitwidth b, specifically 240 for FP8 [Micikevicius et al., 2022] and 6 for FP4. During inference, the forward pass in linear layers can be approximated as:

$$XW \approx \text{scale}_X \cdot \text{scale}_W \cdot Q(X)Q(W) \tag{2}$$

As supported by hardware, low-precision inference benefits from faster floating-point operations, improved memory bandwidth, and efficient datatype conversion. With substantially reduced memory footprint, low-precision inference can significantly lower end-to-end inference latency compared to FP16. For instance, FP8 typically provides up to $2\times$ latency speedup while maintaining near-lossless output quality, making it widely adopted. FP4, on the other hand, can yield up to $4\times$ latency reduction, but often causes severe degradation in model performance, limiting its standalone application. Recent work such as SVDQuant [Li et al., 2025] attempts to mitigate the accuracy loss by combining it with low-rank corrections and smoothing. However, such approaches remain static and do not offer adaptive control over the latency–quality trade-off in real-time, latency-sensitive tasks.

2.2 Additional related work on throughput optimization

Another line of related work focuses on conventional serving scenarios, whose primary goal is to improve serving throughput while maintaining near-lossless performance. For example, systems such

as vLLM [Kwon et al., 2023] and SGLang [Zheng et al., 2024] achieve around $6.4\times$ throughput improvements without compromising output quality. While such systems may reduce latency in specific conditions (e.g., shared prefill structures in SGLang), they are generally not designed to optimize latency in a task-specific manner. Other efforts, such as AI Metropolis [Xie et al., 2024], build distributed cluster systems to accelerate agentic simulations through speculative execution of multiple agents. These approaches aim to maximize simulation throughput but are not tailored for latency-sensitive, real-world agent deployments. Separately, a substantial body of work explores integer quantization to improve serving throughput [Lee et al., 2024, Frantar et al., 2023, Kang et al., 2024b, Zirui Liu et al., 2023]. Unlike hardware-supported FP quantization, integer quantization typically requires highly costly dequantization operations during inference. While it enables larger batch sizes and improves overall throughput, the dequantization overhead significantly limits its effectiveness in reducing latency [Lin et al., 2024, Zhao et al., 2024, Kang et al., 2024a]. Other works [Tang et al., 2023, Pandey et al., 2023] propose mixed-precision schemes combining integer and floating-point formats to balance throughput and accuracy. However, these methods remain static and lack the ability to provide fine-grained, dynamic control over LLM inference latency.

3 Latency-Sensitive Agent Decision Tasks

In this section, we formally define the *latency-sensitive agent decision tasks*, formulate its *latency-quality trade-off*, and introduce two real-time evaluation benchmarks: (i) *HFTBench* – a high-frequency trading system tailored to evaluate real-time trading decisions of LLMs, and (ii) *StreetFighter* – a competitive gaming environment that assess real-time gaming decision of LLMs.

3.1 Formulating Latency-Sensitive Agent Decision Tasks

Consider a general setup of in which an LLM agent interacts with an environment $\mathcal{E}$ to solve a task. At time step t, the agent receives an environmental observation $o_t \in \mathcal{O}$. After spending Δ_t time conducting inference, the agent responds an action $a_{t+\Delta_t} \in \mathcal{A}$ following its decision policy π_θ:

$$a_{t+\Delta_t} \sim \pi_\theta(c_t) \quad \text{where} \quad c_t = \{(o_0, a_{0+\Delta_0}), (o_1, a_{1+\Delta_1}), \ldots, o_t\}. \tag{3}$$

Here c_t is the context to the agent, for example, a conversation between a user and the agent.

As introduced in Section 1, conventional agent tasks exhibit high tolerance to LLM inference latency Δ_t. A simple case is single-step tasks such as one-hop question answering [Kwiatkowski et al., 2019] or document summarization [Shaham et al., 2022], where the agent generates a single action given an initial input prompt o_0, and the outcome is evaluated purely based on the output quality: $r = \mathcal{R}(a|o_0)$, where $\mathcal{R}$ denotes a task-specific evaluation or reward function. A more complex case involves multi-step task-solving, such as multi-step mathematical reasoning or code generation, where the agent produces a sequence of actions over time. In such tasks, the overall performance depends on the cumulative quality of all outputs:

$$r = \sum_t \mathcal{R}(a_{t+\Delta_t}|c_t) \tag{4}$$

In both cases above, the environment is relatively static, and delayed responses are acceptable as long as the agent maintains high response quality. Correctness is prioritized over speed.

However, many real-world tasks, such as gaming, robotic control, and high-frequency trading, take place in dynamic environments $\mathcal{E}_t$ that evolve rapidly over time. This setting remains large unexplored and we name it as *latency-sensitive agent decision tasks*. In these tasks, a delayed actions $a_{t+\Delta_t}$ is often rendered ineffective or obsolete by the time it is executed under the updated environment state $\mathcal{E}_{t+\Delta_t}$, leading to missed opportunities or degraded outcomes. In such setting, the agent is evaluated not only by *what* it decides, but also by *how long* it decides. To succeed, it must produce actions that are both high-quality and timely. The reward thus becomes a function of both the decision and its latency, evaluated under the evolved environment:

$$r = \sum_t \mathcal{R}(a_{t+\Delta_t}|\mathcal{E}_{t+\Delta_t}). \tag{5}$$

This formulation captures the core challenge of latency-sensitive tasks: enabling LLM agents to make fast and accurate decisions in environments where speed is as critical as accuracy.

3.2 HFTBench: High-Frequency Trading Benchmark

Latency and Quality in Financial Trading. High-frequency trading (HFT) involves rapidly submitting buy and sell orders to centralized exchanges, where transactions are strictly matched based on arrival time. In this setting, even millisecond-level differences in reaction latency can significantly impact profitability. Temporary arbitrage opportunities often arise when short-term imbalances cause the bid–ask spread to widen. Agents that respond quickly can capitalize on these brief windows by buying at temporarily depressed prices or selling at elevated ones—before the market rebalances.

However, latency alone is insufficient. High-quality trading decisions rely on correctly interpreting market conditions, which often require processing multi-step patterns in historical prices, order book dynamics, and occasionally external signals such as policy announcements or financial news. While smaller LLMs benefit from lower latency, our experiments show that they often fail to capture such complex financial patterns, resulting in poor decisions that negate their speed advantage.

Benchmark Design. We construct a realistic backtesting simulation using historical per-second trading data from Polygon.io [Polygon.io, 2024]. Each agent receives synchronized market observations at 1-second intervals and must decide whether to take action. To isolate the effect of decision latency, all agents have access to the same information and observation windows.

When an arbitrage opportunity is detected, agents initiate inference. The simulated exchange ranks agents by their response time and assigns execution prices accordingly: faster agents secure more favorable prices. We implement a linearly decaying price model of time and price, where trading advantage diminishes with slower responses—mimicking real-world queue-based order execution.

Evaluation Protocol. Each agent observes a compact state containing prior execution prices, current bid–ask margins, available capital, and time remaining in the trading session. To avoid unnecessary LLM calls, inference is only triggered when the bid–ask margin exceeds a preset threshold b. Agents are evaluated by their cumulative daily yield, and a configurable cooling window t is applied between evaluations to improve simulation efficiency.

3.3 Gaming Benchmark: Street Fighter

Latency Sensitivity in Competitive Games. In real-time competitive games such as *Street Fighter* and *StarCraft*, delayed actions can result in immediate penalties, positional disadvantages, or even round losses. Unlike financial trading, where both decision quality and latency play important roles, these games are overwhelmingly latency-sensitive. In our experiments(Figure 1b), agents with just a 20% reduction in response time consistently outperform their slower counterparts. Interestingly, the strategic depth of *Street Fighter* is relatively limited, and well-prompted small LLMs can produce effective actions, provided they respond quickly enough.

Benchmark Design. We build on top of DIAMBRA's simulation platform [Palmas, 2022] to support real-time *Street Fighter* matches with local model inference. To improve performance for compact models (e.g., <7B parameters), we augment the prompt with tailored few-shot examples specific to each character and scenario. This enhancement helps mitigate performance degradation from reduced model capacity.

Evaluation Protocol. Agents receive a concise game state that includes character-specific move sets, recent action history, and a contextual prompt. We evaluate performance using the ELO rating system [Elo, 1967], where agents compete across multiple matches against a diverse set of opponents. ELO scores are updated dynamically to reflect win–loss outcomes, providing a stable and interpretable metric for real-time decision quality under latency constraints.

3.4 Discussion

LLM Agents in Finance and Gaming. Recent works have explored the application of LLM-based agents in both financial trading and competitive gaming. In finance, FinMem [Yu et al., 2023] and FinAgents [Zhang et al., 2024] demonstrate that LLM agents outperform traditional reinforcement learning and rule-based strategies. This performance gain is attributed to the robustness of LLMs against overfitting and their unique ability to process unstructured inputs, such as policy updates or financial news, through in-context learning. However, these approaches are evaluated on static historical datasets and ignore the role of response timing, which is crucial in real-time trading. In contrast, our high-frequency trading benchmark captures not only the agent's decision quality, but

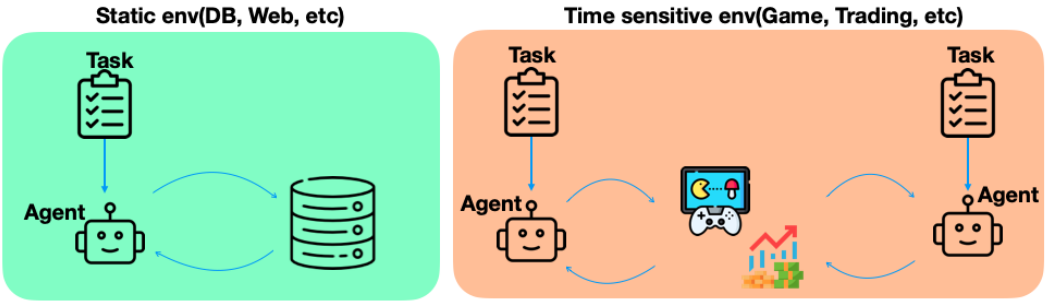


Figure 2: Comparison of agentic LLM for Static environments like code generateion or research and time sensitive environments like trading and gaming. Environment is constantly changing with time and other agent's interaction. For such tasks, reward is related to both quality and latency of agents.

also its response speed and the pricing gap it can exploit —offering a more faithful simulation of real-world trading dynamics.

In the gaming domain, prior work has applied LLM agents to real-time strategy and fighting games such as *StarCraft* and *Street Fighter* [Ma et al., 2024, Palmas, 2022]. These studies primarily focus on improving action quality and designing robust inference pipelines. However, they do not consider the inherent trade-off between latency and decision quality that governs real-time decision performance. Our benchmarks specifically emphasize this trade-off, providing a clearer understanding of how timing impacts success in latency-sensitive environments.

4 FPX: Adaptive Mixed Precision Inference Framework

In this section, we introduce FPX, our adaptive mixed-precision inference algorithm designed for *latency-sensitive agent decision tasks*. As motivated in section 1, FPX dynamically adjusts precision at the operator level, switching between the matrix multiplication kernels of FP8 and FP4, to enable continuous fine-grained control over the latency–quality trade-off.

4.1 Adaptive Mixed-Precision Algorithm Design

The core goal of FPX is to balance latency and accuracy by selectively lowering the precision of only the most compression-tolerant components in a model. Instead of modifying full models or entire layers, we adopt a more granular precision control scheme that applies FP4 only to linear layers that can tolerate aggressive quantization, while preserving FP8 for more sensitive parts.

To ensure compatibility with a wide range of transformer architectures, we focus exclusively on optimizing matrix multiplication operators, which dominate inference latency in LLMs. These include query/key/value (QKV) projections, output projections, and feedforward layers. Other components, such as normalization and attention mechanics, are left untouched to maintain functional correctness and deployment simplicity.

Importantly, because transformer linear layers exhibit similar structural and computational properties, latency gain from replacing FP8 with FP4 is approximately uniform across layers. This decouples precision assignment from latency impact and shifts the optimization focus entirely toward minimizing quality loss. To quantify the robustness of each linear layer to quantization, we compute a relative error metric ε_l based on activation outputs under FP16 and FP4 execution:

$$\varepsilon_l = \frac{\|A_l^{\text{fp16}} - A_l^{\text{fp4}}\|_2}{\|A_l^{\text{fp16}}\|_2} \tag{6}$$

Here, A_l^{fp16} is the output of layer l under FP16 execution, and A_l^{fp4} is the output when the same input is processed using an FP4 kernel. The normalized error ε_l captures the fidelity loss introduced by low-precision inference and serves as the basis for selecting compression candidates.

Given a user-specified compression ratio $\gamma \in [0, 1]$, we define a precision assignment function $\delta(l) \in \{4, 8\}$ for each linear layer l:

$$\delta(l) = \begin{cases} 4 & \text{if } l \in \mathcal{S}_\gamma \\ 8 & \text{otherwise} \end{cases}, \quad \text{where } \mathcal{S}_\gamma = \operatorname{argmin}_{\substack{S \subset \mathcal{L} \\ |S| = \gamma L}} \sum_{l \in S} \varepsilon_l \tag{7}$$

Algorithm 1 Adaptive FP4/FP8 Precision Assignment for Transformer Layers

Require: Transformer model $\mathcal{M}$ with L linear layers $\mathcal{L} = \{l_1, \ldots, l_L\}$, calibration dataset $\mathcal{D}$, compression ratio $\gamma \in [0, 1]$

Ensure: Precision assignment function $\delta(l) \in \{4, 8\}$ for all $l \in \mathcal{L}$

for all layer $l \in \mathcal{L}$ **do**

  Run FP16 inference on $\mathcal{D}$ to collect outputs A_l^{fp16}

  Simulate FP4 output A_l^{fp4} using the same inputs

  Compute relative quantization error:

$$\varepsilon_l = \frac{\|A_l^{\text{fp16}} - A_l^{\text{fp4}}\|_2}{\|A_l^{\text{fp16}}\|_2}$$

end for

Sort layers in ascending order of ε_l

Select $\mathcal{S}_\gamma$ as the γL layers with the smallest ε_l

for all layer $l \in \mathcal{L}$ **do**

  if $l \in \mathcal{S}_\gamma$ **then**

    $\delta(l) \leftarrow 4$ ▷ Assign FP4 to quantization-tolerant layers

  else

    $\delta(l) \leftarrow 8$ ▷ Preserve FP8 for sensitive layers

  end if

end for

return δ

Here, $\mathcal{S}_\gamma$ denotes the subset of γL layers with the lowest quantization error. This design ensures that FP4 is selectively applied to the most robust layers, enabling substantial latency gains while minimizing quality degradation.

4.2 Offline Calibration

To compute the layer-wise quantization error ε_l, we perform a one-time offline calibration using a held-out language modeling dataset. Following standard practice in quantization research [Xiao et al., 2024, Hooper et al., 2024], this calibration phase estimates typical activation distributions observed during inference. Concretely, we run full-precision (FP16) inference on the Wikitext-2 dataset [Merity et al., 2016], capturing both the input and output activations for each linear layer. Then, we simulate FP4 execution by running each layer individually, replacing its FP16 kernel with an FP4 kernel, while keeping all other layers unchanged. This isolates the quantization impact at the layer level and yields a reliable estimate of ε_l for each candidate.

The complete precision assignment pipeline is summarized in Algorithm 1.

5 Experiments

We evaluate our method on two time-sensitive task benchmarks introduced in Section 3. We then perform an ablation study to analyze the lantecy-quality trade-off brought by FPX.

5.1 Experimental Setup

Models. To ensure a fair comparison and reduce the complexity of the search space, we conduct our experiments on a family of models pretrained on similar datasets. Evaluating across heterogeneous model families could introduce biases due to differences in pretraining quality, architecture, or tokenizer design. Therefore, we focus on the Qwen2.5 model suite [Qwen et al., 2025], ranging from 1.5B to 14B parameters.

Benchmark Configurations. For the high-frequency trading (HFT) benchmark, we evaluate on stock data from Nvidia and Amazon on August 5th, 2024. We follow the configuration introduced in Section 3.2, setting the profit threshold b to 2% and the time window t to 1 minute. The initial cash for agent is 10,000 dollars. For the gaming benchmark, we conduct 40 matches between model pairs and compute win rates to derive ElO ratings.

Method Configurations. We discretize the compression ratio γ of FPX into steps of 0.1 to explore the trade-off between latency and accuracy across different benchmarks. We only report the **best-performing setting** for each model in each task. In our experiments, fine-grained changes in γ

Table 1: Evaluation results on latency-sensitive benchmarks. Our method achieves the best latency–reward trade-off across both tasks. Only shows top-6 results. More results are shown in appendix.

HFTBench			
Model Parameter Size	**Bitwidth Avg**	**Latency (ms)↓**	**Daily Yield (%)↑**
14B (ours)	7.2	713	**26.52**
14B	8	801	23.14
14B	16	1302	17.20
7B	16	619	-3.28
7B (ours)	7.6	386	-7.25
7B	8	394	-12.94
Street Fighter			
Model Parameter Size	**Bitwidth Avg**	**Latency (ms)↓**	**ELO Score↑**
3B (ours)	6.8	195	**5.99**
7B (ours)	7.2	354	2.33
3B	8	222	2.19
3B	16	349	0.25
7B	8	394	-0.44
1.5B	8	142	-1.25

generally have minimal effect, indicating that our selected settings are near-optimal. All experiments are run on an RTX 5090 GPU unless otherwise specified. 14B models are served across multiple GPUs using model parallelism.

5.2 Baseline Techniques

Time-sensitive benchmarks are sensitive to both model quality and inference latency. Any quantization method that results in slower inference than FP8 is excluded from consideration. We evaluate the following baselines:

• *FP16*: A standard dense model with both activations and weights in 16-bit floating point. This serves as the upper baseline in quality but incurs the highest latency.

• *FP8*: A widely adopted low-precision format for Hopper and newer GPU architectures, representing both activations and weights as 8-bit floating point. It typically offers near-lossless accuracy with significantly better efficiency than FP16.

• *FP4*: A highly compressed representation where both activations and weights are quantized to 4 bits and packed as 8-bit integers. This setting drastically improves efficiency but at the huge cost of model response quality. It is only available on blackwell architecture GPUs.

5.3 Evaluation Result

Table 1 demonstrates that FPX, by dynamically trading off latency and quality through adaptive model size and bitwidth selection, achieves the highest daily yield on HFTBench and the best overall reward across both benchmarks.

High-Frequency Trading (HFTBench). This benchmark requires a careful balance between latency and response quality. We observe that larger models, such as 14B, outperform smaller alternatives due to their stronger ability to recognize profitable opportunities. In contrast, smaller models often fail to detect high-reward patterns or generate outputs that are too unreliable to be translated into effective trading decisions. FPX improves the latency of the 14B model by compressing 20% of its linear layers into FP4, while preserving FP8 for the rest. This enables a favorable speed–quality trade-off, allowing 14B+FPX to achieve the highest daily yield among all candidates. Interestingly, we find that further reducing the latency of weaker models like 7B actually harms performance. Faster response does not help if the decisions themselves are poor, and can even increase the rate of loss.

Table 2: Performance under different compression levels on Qwen2.5 models for HFTBench and Street Fighter."–" means model performance is complete destroyed.

HFTBench – Qwen2.5-14B			
Gamma (γ)	**Latency (ms)↓**	**PPL↓**	**Daily Yield (%)↑**
0.0 (FP8)	801	4.55	23.14
0.2	713	4.92	**26.52**
0.4	623	6.71	12.93
0.6	558	–	0.00
0.8	503	–	0.00
1.0 (FP4)	489	–	0.00
Street Fighter – Qwen2.5-3B+FPX versus Qwen2.5-3B-FP16			
Gamma (γ)	**Latency (ms)↓**	**PPL↓**	**Winrate (%)↑**
0.0 (FP8)	222	6.85	72.5
0.2	207	7.03	77.5
0.3	200	9.02	**80.0**
0.4	192	11.59	62.5
0.6	178	17.42	12.5
0.8	153	–	0.0
1.0 (FP4)	147	–	0.0

StreetFighter. This task is highly latency-sensitive, yet quality still matters. Our method achieves the best performance with a 3B model configured with 30% of layers in FP4 and 70% in FP8. Notably, although the fastest candidate, the 1.5B model with full FP8 inference, has the lowest latency, it performs poorly due to its limited decision-making capability. Moreover, the environment itself imposes an upper bound on effective response rate. In StreetFighter, each character action takes a fixed amount of in-game time to complete, with an effective frame rate of around 5 actions per second (i.e., 200ms per action). Any optimization that reduces model latency beyond this threshold yields no further benefit, as the game cannot process actions faster than this limit.

5.4 Ablation Study

Latency–Quality Trade-off of FPX We evaluate the Pareto frontier of the latency–quality trade-off induced by FPX across both benchmarks and bitwidth configurations. Specifically, we apply FPX to the Qwen2.5 model family and compare against standard FP16 inference. Our results show that FPX effectively adapts each model's inference path between FP8 and FP4 regimes, dynamically balancing latency and accuracy. Notably, the optimal trade-off point varies by task and model: for instance, on HFTBench with the 14B model, the best performance is achieved at $\gamma = 0.2$, while on Street Fighter with the 3B model, the optimal setting is $\gamma = 0.3$. These findings highlight that latency-sensitive decision-making tasks require task-specific latency–quality configurations, and FPX enables LLM agents to navigate this trade-off effectively.

6 Limitations and Conclusion

In this work, we present the first systematic study of the latency–quality trade-off for LLM-based agents in *latency-sensitive agent decision tasks*. To support this investigation, we introduce two real-time evaluation benchmarks: **HFTBench**, a high-frequency trading simulator, and **StreetFighter**, a competitive gaming environment. In both settings, rapid yet accurate decisions are essential to achieving high downstream rewards.

To meet the heterogeneous demands of these tasks, we propose FPX, an adaptive mixed-precision inference framework that dynamically adjusts model precision to optimize for task-specific latency–quality trade-offs. By selectively applying FP4 quantization to compression-tolerant layers while retaining FP8 for sensitive components, FPX enables fine-grained latency control with minimal performance degradation.

Extensive experiments on Qwen2.5 model variants demonstrate that FPX consistently discovers favorable operating points that outperform fixed-precision baselines across both domains. Our ablation results further reveal that the optimal compression configuration varies significantly by task and model, underscoring the importance of latency-aware deployment strategies for LLM agents.

While FPX demonstrates strong empirical gains, it has limitations. Our current precision assignment operates at the layer level for simplicity and compatibility. More fine-grained schemes, such as token-level precision control, may unlock better trade-offs, but require significantly more complex implementation and kernel support. We left this optimization for future works.

We hope that our benchmarks and findings encourage future research toward building efficient, adaptive LLM systems and algorithms that prioritize latency-awareness in real-world applications, rather than focusing solely on maximizing accuracy or model performance.

7 Acknowledgement

We are deeply grateful to Professor Tsachy Weissman for his guidance. We also thank Jinyan Su(PhD student from Cornell University) for valuable suggestions on refining the paper.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. URL https://arxiv.org/abs/2412.08905. 1

Matthew Baron, Jonathan Brogaard, Björn Hagströmer, and Andrei Kirilenko. Risk and return in high-frequency trading. *Journal of Financial and Quantitative Analysis*, 54(3):993–1024, 2019. doi: 10.1017/S0022109018001096. 2

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate, 2023. URL https://arxiv.org/abs/2305.14325. 1

Arpad E. Elo. The proposed uscf rating system: Its development, theory, and applications. *Chess Life*, pages 21–28, August 1967. URL https://uscf1-nyc1.aodhosting.com/CL-AND-CR-ALL/CL-ALL/1967/1967_08.pdf#page=26. Accessed: 2025-05-05. 5

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2023. URL https://arxiv.org/abs/2210.17323. 4

Xue-Zhong He and Shen Lin. Reinforcement Learning Equilibrium in Limit Order Markets. *Journal of Economic Dynamics and Control*, 144(C), 2022. doi: 10.1016/j.jedc.2022.104497. URL https://ideas.repec.org/a/eee/dyncon/v144y2022ics0165188922002019.html. 2

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization, 2024. URL https://arxiv.org/abs/2401.18079. 7

Hao Kang, Srikant Bharadwaj, James Hensman, Tushar Krishna, Victor Ruhle, and Saravan Rajmohan. Turboattention: Efficient attention approximation for high throughputs llms, 2024a. URL https://arxiv.org/abs/2412.08585. 4

Hao Kang, Qingru Zhang, Souvik Kundu, Geonhwa Jeong, Zaoxing Liu, Tushar Krishna, and Tuo Zhao. Gear: An efficient kv cache compression recipe for near-lossless generative inference of llm, 2024b. URL https://arxiv.org/abs/2403.05527. 4

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026. 4

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180. 4

Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. Owq: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models, 2024. URL https://arxiv.org/abs/2306.02272. 4

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008, 2023. 1

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models, 2025. URL https://arxiv.org/abs/2411.05007. 3

Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving, 2024. URL https://arxiv.org/abs/2405.04532. 4

Weiyu Ma, Qirui Mi, Yongcheng Zeng, Xue Yan, Yuqiao Wu, Runji Lin, Haifeng Zhang, and Jun Wang. Large language models play starcraft ii: Benchmarks and a chain of summarization approach, 2024. URL https://arxiv.org/abs/2312.11865. 6

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. 7

Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, Naveen Mellempudi, Stuart Oberman, Mohammad Shoeybi, Michael Siu, and Hao Wu. Fp8 formats for deep learning, 2022. URL https://arxiv.org/abs/2209.05433. 3

OpenAI. Gpt-4 technical report, 2023. 1

Alessandro Palmas. Diambra arena: a new reinforcement learning platform for research and experimentation, 2022. URL https://arxiv.org/abs/2210.10595. 3, 5, 6

Nilesh Prasad Pandey, Markus Nagel, Mart van Baalen, Yin Huang, Chirag Patel, and Tijmen Blankevoort. A practical mixed precision algorithm for post-training quantization, 2023. URL https://arxiv.org/abs/2302.05397. 4

Polygon.io. Polygon - real-time financial market apis. https://polygon.io, 2024. Accessed: 2025-04-29. 5

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115. 7

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019. 2

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, et al. Scrolls: Standardized comparison over long language sequences. *arXiv preprint arXiv:2201.03533*, 2022. 4

Noah Shinn, Beck Labash, and Ashwin Gopinath. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*, 2(5):9, 2023. 1

Norman Makoto Su. Street fighter iv: braggadocio off and on-line. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pages 361–370, 2010. 2

Chen Tang, Kai Ouyang, Zhi Wang, Yifei Zhu, Yaowei Wang, Wen Ji, and Wenwu Zhu. Mixed-precision neural network quantization via learned layer-wise importance, 2023. URL https://arxiv.org/abs/2203.08368. 4

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa

Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786. 1

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. Recmind: Large language model powered agent for recommendation. *arXiv preprint arXiv:2308.14296*, 2023. 2

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models, 2024. URL https://arxiv.org/abs/2211.10438. 7

Ziyang Xiao, Dongxiang Zhang, Yangjun Wu, Lilin Xu, Yuan Jessica Wang, Xiongwei Han, Xiaojin Fu, Tao Zhong, Jia Zeng, Mingli Song, et al. Chain-of-experts: When llms meet complex operations research problems. In *The twelfth international conference on learning representations*, 2023. 2

Zhiqiang Xie, Hao Kang, Ying Sheng, Tushar Krishna, Kayvon Fatahalian, and Christos Kozyrakis. Ai metropolis: Scaling large language model-based multi-agent simulation with out-of-order execution, 2024. URL https://arxiv.org/abs/2411.03519. 4

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X. 1

Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Denghui Zhang, Rong Liu, Jordan W. Suchow, and Khaldoun Khashanah. Finmem: A performance-enhanced llm trading agent with layered memory and character design, 2023. URL https://arxiv.org/abs/2311.13743. 5

Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, Longtao Zheng, Xinrun Wang, and Bo An. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist, 2024. URL https://arxiv.org/abs/2402.18485. 5

Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. Atom: Low-bit quantization for efficient and accurate llm serving, 2024. URL https://arxiv.org/abs/2310.19102. 4

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. Sglang: Efficient execution of structured language model programs, 2024. URL https://arxiv.org/abs/2312.07104. 4

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. *arXiv preprint arXiv:2406.15877*, 2024. 2

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi : Plug-and-play 2bit kv cache quantization with streaming asymmetric quantization. 2023. doi: 10.13140/RG.2.2.28167.37282. URL https://rgdoi.net/10.13140/RG.2.2.28167.37282. 4

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes it matches the contribution. We are the first the evaluate latency-quality trade-off in such latency-sensitive tasks.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: This paper analyzes the trade-off of latency-sensitive agent decision tasks along with a method. We evaluate the limitations of our method and leave this for future works.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide detailed illustration and assumption for our algorithms.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes we will and we will keeping optimize our benchmark and method.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Ye we will opensource our benchmark and evaluation script of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we have specify the setting of our experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes we do.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes we have specificly pointed out what kind of hardware and platform we use for experiments in experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Yes we make sure to preserve anonymity.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes we discuss the positive societal and impacts of the work in the conclusion section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper pose no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes we introduce two benchmarks and one adaptive quantization algorithm.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This project does not involve research with human nor crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Visualization of HFTBench Data

Here we provide the high-low price per second for the data we have used for HFTBench tests in Figure 3. Red rectangle points out the buy-sell price gap in short time, which provide trading opportunity for agents. Such opportunity only happens in short time. Buying and selling decisions of other agents will decrease the gap quickly in miliseconds.

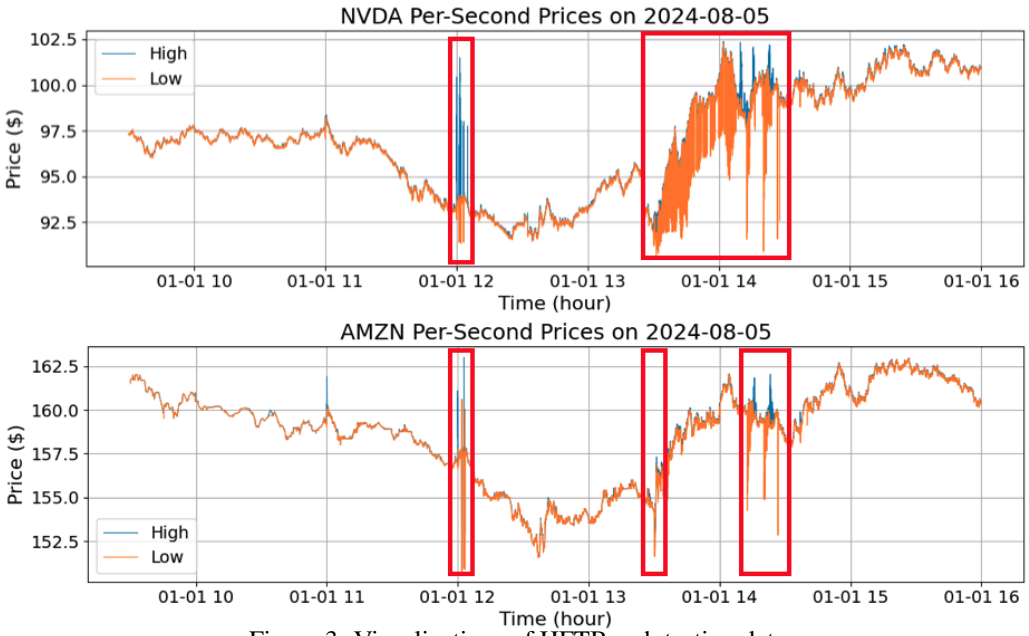


Figure 3: Visualizations of HFTBench testing data.

B More Experiment Results for StreetFighter

Here we provide more results of StreetFighter. Competitors are run for 40 round and calculate the ELO scores.

Table 3: Latency and yield comparison on **StreetFighter**.

Model Parameter Size	Bitwidth Avg	ELO Score(%)↑
3B	6.8	**5.65**
3B	7.2	3.57
7B	6.8	2.33
7B	7.2	2.33
3B	8	2.18
3B	16	0.26
7B	8	-0.45
1.5B	16	-1.25
1.5B	8	-2.66
7B	16	-2.89
14B	8	-3.14
14B	16	-5.94

C Latency Profiling of Quantization method

We conduct a detailed latency profiling of various quantization methods on RTX 5090 GPUs. For the 14B model, we employ model parallelism across two GPUs. The results are summarized in Table 4. Our findings show that both FP8 and FP4 kernels yield substantial latency reductions compared to

the FP16 baseline. However, for the W4A16 configuration, where model weights are stored as 4-bit integers, the latency benefits are less pronounced, except in large models such as Qwen2.5-14B. This is likely due to the overhead introduced by data type conversion and dequantization. These results suggest that hybrid usage of FP8 and FP4 kernels is a promising strategy for improving inference efficiency, particularly on large-scale models.

Table 4: Latency (ms) Comparison Across Quantization Schemes

Model	**FP16**	**FP8**	**W4A16(int)**	**FP4**
Qwen-1.5B	203	142	254	83
Qwen-3B	349	222	323	147
Qwen-7B	619	394	537	248
Qwen-14B (2×5090)	1302	801	792	492