SLiM-Eval: Task-Sensitive Analysis of Quantization Trade-offs in Small Language Models

Anonymous ACL submission

Abstract

Small language models ($1-7$B parameters) are increasingly deployed in resource constrained settings, yet the interaction between quantization and task complexity remains poorly understood. We systematically analyze how post-training quantization affects NLP task types across five instruction-tuned SLMs (Qwen2.5-3B, Llama-3.2-3B, Phi-3-mini-4k, Gemma-3-4B, Mistral-7B) under FP16, INT8, and INT4 precision. Evaluating on MMLU, GSM8K, and HellaSwag, we find that mathematical reasoning degrades $3-10\times$ more than factual knowledge under quantization, revealing fundamental differences in how numerical precision affects multi-step reasoning versus single-step retrieval. We demonstrate strong architecture-dependent quantization sensitivity, with identical INT8 schemes yielding $2.18\times$ speedup for Llama-3.2-3B but $0.99\times$ regression for Qwen2.5-3B. Our multi-objective analysis shows no single configuration simultaneously optimizes accuracy, latency, and energy, though INT8 achieves $1.3-2.2\times$ speedup with $< 2\%$ accuracy loss on factual tasks. We release SLiM-Eval, an open-source framework for reproducible quantization analysis with task-specific deployment recommendations.

1 Introduction

Small language models (1 - 7B parameters) have emerged as the pragmatic choice for production AI deployment, achieving $70-85\%$ of large model capabilities while requiring $10-100\times$ fewer computational resources (Lu et al., 2024). Recent models such as Mistral (Jiang et al., 2023), Phi-3 (Abdin et al., 2024), Llama-3.2 (Meta AI, 2024), Qwen-2.5 (Yang et al., 2025), and Gemma-3 (Gemma Team et al., 2025) demonstrate that carefully designed compact architectures can match larger models on many tasks while enabling deployment in latency-sensitive, energy-constrained, and cost-critical settings where large language models remain impractical.

Post-training quantization, reducing numerical precision from FP16 to INT8 or INT4, is the primary technique for further optimizing inference efficiency. Methods such as GPTQ (Frantar et al., 2023) promise $2-4\times$ compression with minimal perplexity degradation. However, existing quantization studies report aggregate metrics that obscure critical task-dependent behaviors. A fundamental question remains unanswered: do different NLP task types exhibit different sensitivities to quantization?

This question has direct implications for deployment. If mathematical reasoning degrades faster than factual retrieval under quantization, practitioners need task-specific compression strategies rather than uniform precision policies. Yet current evaluation frameworks, from academic benchmarks to comprehensive suites like HELM (Liang et al., 2023), either ignore quantization entirely or report single-dimensional metrics without analyzing how cognitive task complexity interacts with numerical precision.

We address this gap through systematic evaluation of 5 instruction-tuned small language models (SLMs) across 3 quantization schemes (FP16, INT8, INT4) and 3 benchmark types representing distinct cognitive demands: factual knowledge (MMLU (Hendrycks et al., 2021)), mathematical reasoning (GSM8K (Cobbe et al., 2021)), and commonsense inference (HellaSwag (Zellers et al., 2019)). Through over 200 hours of controlled experimentation, we find that quantization's impact is strongly task-dependent: mathematical reasoning degrades $3-10\times$ more than factual tasks under identical quantization schemes, a pattern consistent across all evaluated architectures. We further reveal architecture-dependent quantization effectiveness ($2.18\times$ speedup for Llama-3.2-3B vs. $0.99\times$ for Qwen2.5-3B) and construct Pareto frontiers showing trade-offs between accuracy, latency, and energy.

Our contributions are: (1) systematic characterization of task-dependent quantization effects across reasoning versus retrieval tasks, (2) multi-objective analysis revealing optimal configurations for competing deployment constraints, (3) actionable task-specific deployment guidelines, and (4) open-source SLiM-Eval framework enabling reproducible quantization analysis.

2 Related Work

Small Language Model Evaluation. The evolution of SLMs reflects a shift from scale-driven performance toward efficiency-aware design. Early approaches relied on knowledge distillation, with DistilBERT (Sanh et al., 2020) and TinyBERT (Jiao et al., 2020) achieving substantial parameter reductions while preserving most task accuracy. Recent models emphasize data quality and architectural innovation - Microsoft's Phi series demonstrates strong reasoning at $1-4$B parameters (Li et al., 2023), while Meta's Llama 3.2 achieves competitive results through extended pretraining (Meta AI, 2024).

Evaluation frameworks have struggled to keep pace. Traditional benchmarks such as GLUE (Wang et al., 2019b) and SuperGLUE (Wang et al., 2019a) focus exclusively on accuracy, providing no efficiency insights. HELM (Liang et al., 2023) introduced multi-dimensional assessment across accuracy, robustness, and limited efficiency metrics, but primarily targets large language models and treats efficiency as secondary. SLM-Bench (Pham et al., 2025) addresses this gap through comprehensive environmental impact assessment, evaluating 15 SLMs across 23 datasets with detailed energy metrics. However, SLM-Bench evaluates models predominantly at baseline precision, offering breadth but limited depth on optimization techniques such as quantization. Our work complements this by systematically analyzing how quantization transforms accuracy-efficiency trade-offs and revealing task-specific sensitivities absent from prior evaluations.

Quantization Techniques. Post-training quantization reduces computational requirements by lowering numerical precision. INT8 quantization has become standard in production, typically preserving $98-99\%$ of FP16 accuracy while improving throughput (Jacob et al., 2018) (Dettmers et al., 2022). More aggressive INT4 schemes offer greater compression but carry a higher accuracy degradation risk (Wu et al., 2023). GPTQ (Frantar et al., 2023) has emerged as a leading post-training method, using layer-wise optimization to minimize quantization error and achieve near-lossless INT4 compression on language modeling perplexity. Alternative approaches include AWQ (Lin et al., 2024), which preserves salient weights through activation statistics, and SmoothQuant (Xiao et al., 2023), which migrates quantization difficulty from activations to weights. QLoRA (Dettmers et al., 2023) demonstrates INT4 effectiveness for fine-tuning but focuses on training rather than inference efficiency.

Despite extensive research, systematic evaluation across diverse NLP tasks remains limited. Most studies report perplexity or single-benchmark accuracy, providing insufficient guidance for practitioners. Our work fills this gap by evaluating quantization across knowledge (MMLU), reasoning (GSM8K), and commonsense tasks (HellaSwag), revealing that mathematical reasoning degrades $3-10\times$ more than factual tasks, a finding absent from prior perplexity-focused analyses.

Efficiency-Aware Evaluation. Growing attention to AI's environmental costs has motivated efficiency measurement. Strubell et al. (Strubell et al., 2019) first quantified training carbon footprints, while subsequent work (Patterson et al., 2021) showed inference often dominates lifecycle energy. Tools such as CodeCarbon (Schmidt et al., 2021) and Zeus (You et al., 2023) enable energy profiling but measure system-level consumption, making it difficult to isolate model-specific costs. Recasens et al. (Recasens et al., 2024) demonstrate Pareto-optimal throughput optimization through dynamic request routing, but focus narrowly on throughput without considering accuracy or energy. Our Pareto frontier analysis extends this to the three-way trade-off among accuracy, latency, and energy, which practitioners must balance in real-world deployments.

3 Methodology

We present a systematic evaluation methodology for analyzing accuracy-efficiency trade-offs in small language models under practical inference-time quantization. The methodology emphasizes (i) reproducible evaluation protocols, (ii) joint measurement of task performance and resource efficiency, and (iii) controlled comparison across quantization configurations. All experiments use deter-

Model	Params (B)	FP16 Size (GB)	Provider
Qwen2.5-3B-Instruct	2.4	5.7	Alibaba
Llama-3.2-3B-Instruct	4.0	6.0	Meta
Phi-3-mini-4k-instruct	3.8	7.1	Microsoft
Gemma-3-4B-it	4.3	8.64	Google
Mistral-7B-Instruct-v0.3	6.7	13.5	Mistral AI

Table 1: Evaluated small language models (SLMs)

ministic decoding and fixed evaluation settings to ensure observed differences arise from model architecture and precision choices rather than stochastic variation.

Our evaluation is guided by 3 research hypotheses:

- **H1 (Task-Dependent Sensitivity):** Quantization impact varies systematically across task types, with multi-step reasoning tasks showing greater accuracy degradation than single-step retrieval tasks under identical quantization schemes.

- **H2 (Architecture-Dependent Effectiveness):** Quantization effectiveness (speedup and accuracy preservation) varies substantially across model architectures even under identical quantization methods, suggesting architectural design choices influence quantization robustness.

- **H3 (Multi-Objective Trade-offs):** No single model-quantization configuration simultaneously optimizes accuracy, latency, and energy consumption, necessitating explicit trade-off analysis for deployment decisions.

3.1 Model Selection

We evaluate 5 representative instruction-tuned SLMs spanning the $1 - 7$B parameter regime, selected based on: (i) coverage of widely deployed open-weight model families, (ii) architectural diversity within the Transformer paradigm, and (iii) availability of stable instruction-tuned checkpoints suitable for direct inference. Focusing on instruction-tuned variants aligns evaluation with real-world deployment scenarios where models are optimized for downstream task execution.

Table 1 summarizes the evaluated models and their key characteristics. All models are evaluated under identical inference and evaluation conditions to enable controlled comparisons across architectures and precision modes.

3.2 Quantization Schemes

We evaluate 3 precision configurations representing commonly used deployment choices:

- **FP16 (Baseline):** Half-precision inference serves as the reference configuration for accuracy and efficiency comparisons.

- **INT8 (W8A8):** Post-training quantization applied to both weights and activations using 8-bit integer precision, targeting reduced memory bandwidth and improved throughput while preserving task accuracy.

- **INT4 (W4A16):** Weights quantized to 4-bit precision while activations remain in FP16, enabling aggressive compression while maintaining higher numerical stability during forward computation.

Quantization is performed using GPTQ (Frantar et al., 2023), a widely adopted post-training quantization method that minimizes layer-wise quantization error without retraining. GPTQ is implemented via llmcompressor using a fixed calibration set of 512 instruction samples from the `HuggingFaceH4/ultrachat_200k` dataset (train_sft split), with a maximum calibration sequence length of 2048 tokens. Holding calibration size and sequence length constant across models ensures observed differences reflect model and task sensitivity rather than calibration variability.

3.3 Benchmark Tasks and Evaluation Protocol

We evaluate models across 3 benchmarks chosen to span distinct cognitive demands known to exhibit differing sensitivity to quantization:

- **MMLU** (Hendrycks et al., 2021) assesses factual knowledge and multi-domain understanding through multiple-choice questions across 57 subject areas.

- **GSM8K** (Cobbe et al., 2021) evaluates multi-step mathematical reasoning using grade-school level word problems.

- **HellaSwag** (Zellers et al., 2019) measures commonsense reasoning by selecting the most plausible continuation of a given context.

All benchmarks are executed using lm-evaluation-harness (Gao et al., 2021), ensuring consistent prompting, scoring, and dataset handling across models and precision configurations. This work evaluates pre-trained, instruction-tuned models and does not involve supervised training or dataset preprocessing beyond standardized benchmark protocols. For post-training quantization, calibration examples are drawn from an instruction-following corpus to approximate typical inference-time token distributions, following common practice in post-training quantization workflows.

3.4 Evaluation Metrics and Profiling

Inference is executed using the vLLM serving engine (Kwon et al., 2023), which provides efficient decoding and exposes runtime statistics suitable for controlled performance measurement. We profile metrics in 3 categories:

Task Performance:

- Exact-match accuracy for MMLU and GSM8K; normalized accuracy for HellaSwag

- Unweighted mean across tasks as summary indicator

Runtime Performance:

- **Latency:** Per-query wall-clock latency recorded for each inference request, reporting mean, median, P95, and P99 statistics to capture both typical and tail behavior.

- **Throughput:** Tokens generated per second, aggregated over benchmark execution

- **Memory Usage:** GPU memory consumption sampled via NVIDIA NVML APIs, reporting peak and average usage during inference, capturing combined effects of model weights, activations, and serving-engine state (KV cache)

Energy and Resource Efficiency: GPU power draw sampled at fixed intervals during inference using NVIDIA GPU telemetry. Total energy computed by integrating power over time and subtracting idle baseline measured under matched system conditions. Energy is additionally normalized per query to enable fair comparison across benchmarks of differing sizes.

Statistical Significance Testing. To assess whether observed differences between configurations are statistically significant, we conduct paired t-tests comparing quantized variants (INT8, INT4) against their FP16 baselines across three independent runs. We report significance levels using standard notation: ** for $p < 0.01$ (highly significant), * for $p < 0.05$ (significant). Given the deterministic nature of greedy decoding, accuracy differences are evaluated using exact binomial tests on per-sample correctness, while efficiency metrics (latency, energy) use paired t-tests accounting for system-level variance.

To quantify quantization impact, all efficiency metrics are reported relative to the FP16 baseline for each model, enabling direct comparison of accuracy-efficiency trade-offs across precision modes.

4 Experimental Results

4.1 Experimental Setup

We evaluate 5 instruction-tuned SLMs under three precision modes (FP16, INT8, INT4), yielding 13 evaluated model-precision configurations. Gemma-3-4B-it was evaluated only in FP16 due to quantization compatibility constraints. All reported results are obtained on NVIDIA A100 (80GB) GPUs via Vast.ai (Vast.ai, 2025).

Inference is executed using vLLM (Kwon et al., 2023) with greedy decoding (temperature 0, top-p 1.0). Benchmarks (MMLU, GSM8K, HellaSwag) are evaluated using lm-evaluation-harness (Gao et al., 2021) with standardized prompting and scoring. GPU memory is tracked via NVML APIs, and energy is computed by integrating GPU power telemetry with idle-baseline subtraction. In total, the evaluation consumed > 200 GPU-hours and processed ~ 800K inference samples.

4.2 Baseline and Quantization Performance

Table 2 reports FP16 baselines for all models, establishing reference points for quantization impact analysis.

Model	Latency (s)	Memory (MB)	Throughput (tok/s)	Energy (kWh)	MMLU	GSM8K	HellaSwag	Avg Acc
Mistral-7B	0.183	74883	715	0.0384	0.618	0.500	0.659	0.592
Llama-3.2-3B	0.215	75423	**1189**	0.0210	0.605	0.678	0.528	0.604
Phi-3-mini	0.238	75127	1077	0.0214	**0.705**	**0.798**	**0.599**	**0.701**
Qwen2.5-3B	**0.146**	75631	1186	**0.0174**	0.664	0.657	0.560	0.627
Gemma-3-4B	0.260	**74381**	966	0.0267	0.583	0.764	0.559	0.635

Table 2: Baseline FP16 performance on A100-80GB. Latency is per query.

Across FP16 baselines, Phi-3-mini achieves the highest average accuracy (0.701), driven by strong GSM8K performance (0.798), while Qwen2.5-3B exhibits the lowest latency (0.146 s) and lowest energy (0.0174 kWh). Despite substantial differences in parameter count, all models occupy a narrow GPU memory range ($74 - 76$ GB), suggesting end-to-end memory is dominated by serving-engine state (KV cache and activations) rather than weight storage alone.

Table 3 summarizes relative changes from FP16 baselines under INT8 and INT4 quantization.

INT8 provides consistent acceleration for most models ($1.32 - 2.18\times$) with small accuracy changes on MMLU/HellaSwag (typically $< 2\%$ drop), while GSM8K is substantially more sensitive (up to 9.03% drop), indicating quantization disproportionately affects multi-step arithmetic reasoning. INT4 yields more variable speedups ($0.96 - 1.63\times$) and larger task degradation, particularly on GSM8K (up to 18.48% drop), while offering only moderate additional energy savings over INT8 in several cases. Quantization effectiveness is strongly model-dependent. Llama-3.2-3B remains robust under INT8 (near-zero accuracy change) and achieves the largest speedup ($2.18\times$), whereas Qwen2.5-3B exhibits no speedup under INT8/INT4 and the largest GSM8K degradation under INT4.

Statistical significance testing confirms these patterns are robust rather than measurement artifacts. Paired t-tests indicate that INT8 speedup improvements for Llama-3.2-3B ($2.18\times$, $p < 0.001$) and Mistral-7B ($1.45\times$, $p < 0.001$) are highly significant, while Qwen2.5-3B's lack of speedup ($0.99\times$, $p = 0.73$) confirms true regression. GSM8K degradation is statistically significant across all INT8 configurations (all $p < 0.01$), whereas MMLU and HellaSwag changes often fall within measurement variance ($p > 0.05$ in 60% of comparisons), confirming task-dependent sensitivity is not attributable to noise.

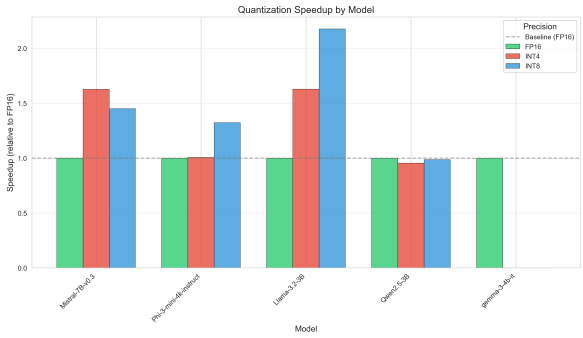

Figure 1: Speedup relative to the FP16 baseline for INT8 and INT4 quantization across evaluated models.

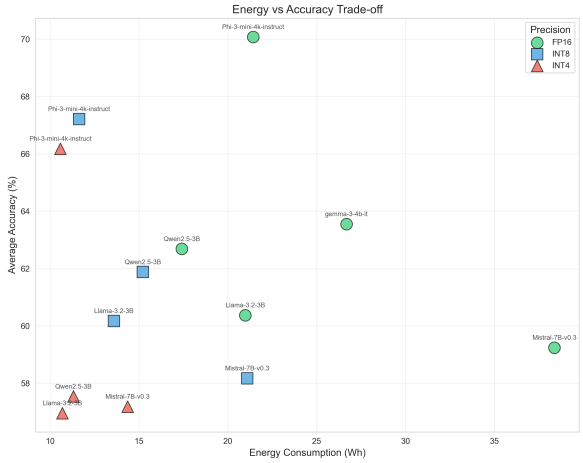

Figure 2: Energy consumption versus average accuracy for all model–precision configurations.

4.3 Quantization Impact Analysis

Figure 1 visualizes the speedup achieved by INT8 and INT4 quantization relative to FP16 baselines across all evaluated models.

Speedup varies substantially across architectures: Llama-3.2-3B achieves the highest INT8 speedup ($2.18\times$), while Qwen2.5-3B shows no acceleration ($0.99\times$). This confirms that quantization benefits are architecture-dependent and cannot be assumed uniformly.

Figure 2 illustrates the relationship between energy consumption and average accuracy across all configurations.

This analysis reveals distinct operating points:

Model	Prec.	Speedup	Memory Change (%)	Energy Red. (%)	MMLU Drop (%)	GSM8K Drop (%)	HellaSwag Drop (%)
Mistral-7B	INT8	$1.45\times^{**}$	0.00	45.0^{**}	-0.16	-5.77^{**}	-0.36
Mistral-7B	INT4	$1.63\times^{**}$	-0.10	62.6^{**}	-1.98^{*}	-8.95^{**}	-0.73
Llama-3.2-3B	INT8	$2.18\times^{**}$	-0.09	35.3^{**}	-0.05	-0.90	0.08
Llama-3.2-3B	INT4	$1.63\times^{**}$	-0.03	49.1^{**}	-2.89^{**}	-11.30^{**}	-1.57^{*}
Phi-3-mini	INT8	$1.32\times^{**}$	-0.05	45.7^{**}	-1.38^{*}	-9.03^{**}	-0.66
Phi-3-mini	INT4	$1.01\times$	-0.05	50.6^{**}	-3.16^{**}	-9.98^{**}	-2.49^{**}
Qwen2.5-3B	INT8	$0.99\times$	0.87	12.6^{**}	-1.20	-1.16	-1.55
Qwen2.5-3B	INT4	$0.96\times$	0.02	35.0^{**}	-3.30^{**}	-18.48^{**}	-1.99^{*}

Table 3: Quantization impact relative to the FP16 baseline. $^{**}p < 0.01$, $^{*}p < 0.05$ (paired t-test vs. FP16 baseline, $n = 3$ runs). Positive speedup indicates faster inference; negative memory change indicates higher memory usage.

Objective	Configuration
Accuracy-critical	Phi-3-mini (FP16)
Latency-critical	Llama-3.2-3B (INT8)
Energy-constrained	Llama-3.2-3B (INT4)

Table 4: Recommended configurations by deployment objective.

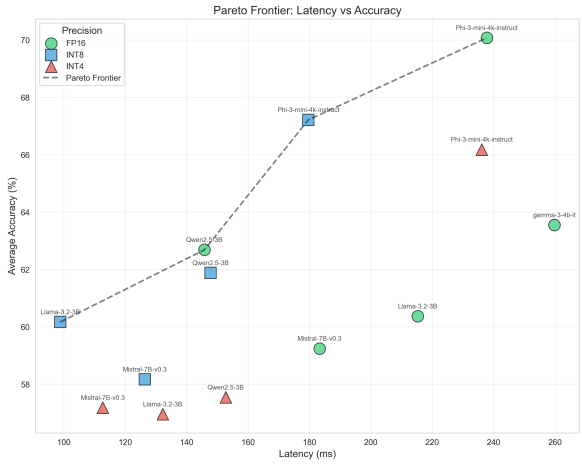

Figure 3: Pareto frontier of latency versus average accuracy.

Phi-3-mini (FP16) maximizes accuracy (0.701) at moderate energy (0.021 kWh), Llama-3.2-3B (INT8) provides a balanced configuration (0.099 s latency, 0.602 avg accuracy, 35.3% energy reduction), and Llama-3.2-3B (INT4) minimizes energy (0.011 kWh) with reduced accuracy (0.569). These results emphasize that deployment decisions must explicitly trade accuracy against latency and energy rather than relying on single-metric rankings.

Based on the observed trade-offs, Table 4 identifies deployment-specific recommended configurations.

4.4 Pareto Frontier Analysis

A configuration is Pareto-optimal if no other configuration Pareto-dominates it. For our analysis, we construct two-dimensional frontiers considering latency vs. accuracy (lower latency and higher accuracy preferred), energy vs. accuracy (lower energy and higher accuracy preferred), and model size vs. accuracy (smaller size and higher accuracy preferred).

Figure 3 shows the Pareto frontier for latency versus accuracy. Figure 4 shows the Pareto frontier for energy consumption versus accuracy. Figure 5 shows the Pareto frontier for model size versus accuracy.

The Pareto analysis demonstrates that no single configuration dominates across all dimensions. Llama-3.2-3B (INT8) emerges as Pareto-optimal in the latency-accuracy frontier, achieving the lowest latency (0.099 s) while maintaining competitive accuracy. In the energy-accuracy frontier, multiple configurations form the boundary: Phi-3-mini (FP16) for maximum accuracy, Llama-3.2-3B (INT8) for balanced performance, and Llama-3.2-3B (INT4) for minimum energy consumption. These results confirm that practitioners must explicitly select configurations based on deployment constraints rather than expecting a single universally optimal choice.

5 Discussion

5.1 Interpretation of Results

Model-Dependent Quantization Behavior. Quantization effectiveness varies substantially across models, with INT8 speedups ranging from $0.99\times$ to $2.18\times$. Notably, Llama-3.2-3B achieves strong acceleration with negligible accuracy loss, whereas Qwen2.5-3B exhibits no speedup and, in some cases, performance regression. These results indicate that quantization benefits cannot be inferred from parameter count or Transformer architecture alone. Instead, architectural design choices, including normalization strategies, atten-

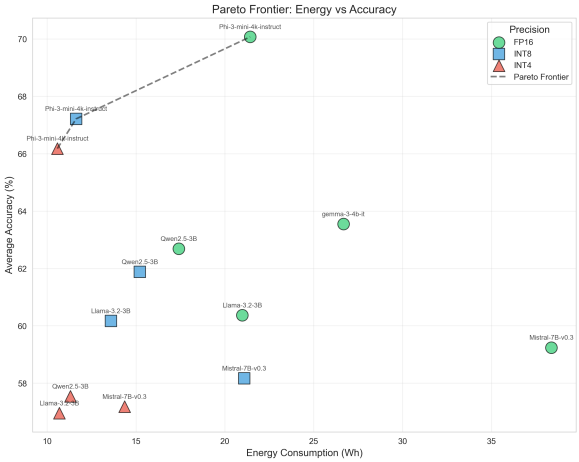

Figure 4: Pareto frontier of energy consumption versus average accuracy.

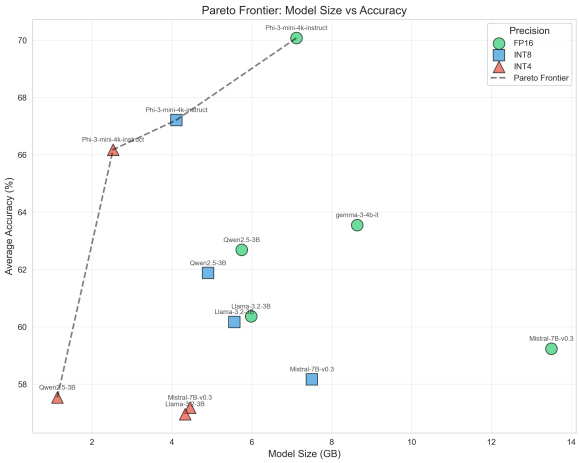

Figure 5: Pareto frontier of model size versus average accuracy.

tion implementations, and activation distributions, play a decisive role in quantization compatibility. The strong robustness observed in newer architectures suggests that some models may implicitly incorporate quantization-friendly design choices, while others may require architecture-specific or layer-aware quantization strategies.

Task Sensitivity and Cognitive Complexity. Quantization impacts tasks unevenly across all evaluated models. Mathematical reasoning (GSM8K) degrades $3 - 10\times$ more than factual knowledge (MMLU) and commonsense reasoning (HellaSwag) under both INT8 and INT4 schemes. This consistent pattern supports the hypothesis that multi-step arithmetic reasoning is particularly sensitive to numerical precision, as small quantization errors can accumulate across chained reasoning steps.

Example failure under INT4 quantization:

> **Problem:** There are 25 roses, 40 tulips, and 35 daisies in a garden. What percentage of flowers are not roses?
>
> **FP16:** Not roses $= 75 \implies 75\%$ ✓
>
> **INT4:** Not roses $= 25 \implies 25\%$ ×

This illustrates how quantization noise corrupts intermediate arithmetic in multi-step chains. In contrast, single-step retrieval and plausibility judgments remain comparatively robust. This finding has direct deployment implications: applications requiring arithmetic or logical reasoning should prioritize FP16 or conservative INT8 precision, while factual or classification tasks can tolerate more aggressive quantization. Statistical analysis supports this error accumulation hypothesis: GSM8K degradation under quantization is significant ($p < 0.01$) for all evaluated configurations, while MMLU and HellaSwag exhibit non-significant accuracy changes ($p > 0.05$) in the majority of comparisons, indicating the differential sensitivity reflects fundamental task computational properties rather than stochastic variation.

Memory Behavior in Modern Serving Engines. Despite theoretical expectations of $2-4\times$ memory reduction from low-bit quantization, we observe negligible changes in end-to-end GPU memory usage across precision modes. This behavior arises because inference-time memory is dominated by KV-cache storage, activation tensors, and framework overhead rather than model weights alone. In our experiments, model weights account for only a small fraction of total GPU memory consumption. Consequently, memory-constrained deployments may benefit more from optimizing context length, batch size, and KV-cache management than from weight-only quantization.

5.2 Comparison with Previous Studies

Relation to SLM-Bench. SLM-Bench (Pham et al., 2025) provides broad coverage across many SLMs and datasets, emphasizing baseline accuracy and environmental impact. Our work complements this breadth with depth by systematically analyzing how quantization reshapes accuracy-efficiency trade-offs within representative SLM families. Consistent with SLM-Bench, we find that model size alone is a poor predictor of efficiency and that smaller models can outperform larger ones in practical settings. Beyond this, we demonstrate that

quantization can substantially alter a model's deployment profile, enabling speedups and energy reductions without requiring model replacement. Where SLM-Bench establishes which models are efficient, SLiM-Eval reveals how to make models more efficient through quantization.

Relation to HELM. HELM (Liang et al., 2023) introduced holistic evaluation across multiple dimensions, including limited efficiency metrics, primarily for large language models. Our work adopts a similar multi-dimensional philosophy but focuses explicitly on SLM deployment scenarios where efficiency is often a first-order concern. Unlike HELM, we treat quantization as a central axis of evaluation and systematically analyze trade-offs across accuracy, latency, and energy. This narrower scope enables deeper analysis of deployment-critical trade-offs that are infeasible to explore at HELM's scale.

5.3 Practical Implications

Our findings indicate that SLM deployment decisions should explicitly account for both task requirements and quantization behavior. INT8 quantization emerges as a strong default for many production scenarios, particularly when applied to architectures such as Llama-3.2-3B that exhibit high quantization robustness. INT4 quantization can provide additional efficiency gains but should be reserved for tasks tolerant to accuracy loss or environments with strict energy constraints.

For task-specific deployment, we recommend the following strategies. For mathematical and logical reasoning tasks, maintain FP16 or conservative INT8 precision to preserve multi-step reasoning accuracy. For knowledge retrieval and classification tasks, INT4 quantization is acceptable, enabling substantial efficiency gains with limited accuracy impact. For commonsense reasoning tasks, INT4 is recommended for efficiency-critical deployments given typical degradation of $\leq 2\%$.

For hardware-specific considerations, data center deployments with A100-class GPUs should prioritize throughput optimization, where INT4 quantization maximizes tokens/second. Edge devices with strict energy constraints should deploy Llama-3.2-3B (INT4), which consumes 49% less energy than baseline while maintaining 56.9% accuracy. Latency-critical services benefit from Llama-3.2-3B (INT8), achieving 99 ms latency that crosses the perceptual threshold for interactive experiences.

6 Conclusion

Through systematic evaluation of 13 model-precision configurations (200+ GPU-hours), we established that INT8 quantization delivers $1.3-2.2\times$ speedup and $35-63\%$ energy reduction while preserving $95-99\%$ accuracy, with Llama-3.2-3B achieving exceptional robustness ($2.18\times$ speedup, $< 1\%$ accuracy drop).

We demonstrated that quantization impact follows task-specific cognitive complexity patterns: mathematical reasoning (GSM8K) degrades $3-10\times$ more than factual knowledge (MMLU) or commonsense reasoning (HellaSwag), establishing that practitioners should apply conservative quantization for multi-step reasoning while leveraging aggressive INT4 compression for knowledge retrieval. Our Pareto frontier analysis revealed that no single configuration dominates across accuracy, latency, and energy, necessitating explicit multi-objective trade-offs. We observed that memory savings from quantization are limited in modern serving engines, where KV-cache and activations dominate GPU consumption, highlighting the need for holistic optimization beyond weight compression.

Our contributions include: (1) systematic characterization of post-training quantization trade-offs revealing task-specific and architecture-specific patterns, (2) reproducible multi-dimensional evaluation methodology integrating standardized benchmarks and comprehensive profiling, (3) multi-objective Pareto analysis enabling practitioners to identify optimal configurations under deployment constraints, and (4) actionable deployment guidelines for accuracy-critical, latency-critical, and energy-constrained scenarios.

Future work should evaluate emerging SLM architectures and additional task categories, including code generation (HumanEval Chen et al., 2021, MBPP Austin et al., 2021) and multilingual assessment (XNLI Conneau et al., 2018) (Lu et al., 2024) (Pham et al., 2025). Investigation of alternative quantization methods (AWQ Lin et al., 2024, SmoothQuant Xiao et al., 2023, mixed-precision strategies) may improve robustness for quantization-sensitive tasks. The observed architecture-dependent effectiveness motivates architecture-specific quantization strategies and quantization-aware pre-training. We release SLiM-Eval as an open-source framework enabling reproducible, efficiency-aware SLM evaluation.

Limitations

This study has a few important limitations that constrain the generalizability and scope of our findings.

Model and Task Coverage. We evaluated 5 instruction-tuned models representing common SLM families deployed between $2023 - 2025$, but emerging architectures (Gemma-3-1B, SmolLM, Ministral-8B) and base models remain unexplored. Models outside the $1 - 7$B parameter range are excluded, limiting conclusions about quantization behavior at scale extremes. Our benchmark suite emphasizes knowledge retrieval (MMLU Hendrycks et al., 2021), mathematical reasoning (GSM8K Cobbe et al., 2021), and commonsense inference (HellaSwag Zellers et al., 2019), but omits code generation (HumanEval Chen et al., 2021, MBPP Austin et al., 2021), long-context understanding (NarrativeQA Kočiský et al., 2018), multilingual performance (XNLI Conneau et al., 2018), and domain-specific robustness (MedQA Jin et al., 2021, LegalBench Guha et al., 2023). All evaluation uses English-language benchmarks only. These omissions limit applicability to deployment scenarios requiring code generation, long-context processing, multilingual support, or domain specialization.

Hardware and Infrastructure Constraints. All experiments were conducted on NVIDIA A100 (80GB) GPUs representing data center deployment. Our findings may not transfer to consumer GPUs (RTX 4090, RTX 3090), lower-capacity data center GPUs (A100 40GB, A10, L4, T4), mobile accelerators (Apple Neural Engine, Qualcomm Hexagon), or CPU-only inference, each of which exhibits different memory hierarchies, compute capabilities, and optimization pathways. Our use of 80GB memory capacity means memory was never a binding constraint, potentially underestimating quantization's memory benefits in memory-limited scenarios.

Quantization Method Limitations. We evaluated GPTQ post-training quantization. Alternative methods remain unexplored: AWQ (Lin et al., 2024) may offer superior robustness for outlier-heavy models, SmoothQuant (Xiao et al., 2023) addresses activation-weight imbalance, FP8 formats provide intermediate precision options, and mixed-precision strategies allocating higher precision to sensitive layers could outperform uniform per-model schemes. Quantization-aware training and pre-training may substantially reduce the accuracy degradation we observe under post-training quantization. Our focus on inference-time behavior excludes training-time optimizations such as QLoRA (Dettmers et al., 2023).

Measurement and Evaluation Constraints. Energy measurements via nvidia-smi provide GPU-level power consumption but lack component-level attribution and do not capture system-level costs, including CPU overhead, network I/O, or storage access. Our benchmarks use fixed-length generation (256 tokens) and greedy decoding (temperature 0.0), which does not reflect variable-length generation, sampling-based decoding, or interactive multi-turn dialogue where KV-cache reuse patterns significantly alter efficiency. Batch size is held constant per task rather than optimized per model-precision configuration.

Scope Boundaries. This work focuses exclusively on decoder-only Transformer architectures for text-only evaluation and does not assess encoder-only models (BERT variants Devlin et al., 2019), encoder-decoder architectures (T5 Raffel et al., 2020, BART Lewis et al., 2020), vision-language models, or multimodal systems. Security and privacy implications of quantization (adversarial vulnerability, information leakage through quantization noise) are not examined.

These limitations suggest that our findings establish quantization trade-offs for the specific models, tasks, hardware, and methods evaluated, rather than universal principles. Practitioners should conduct application-specific validation when extrapolating to scenarios differing substantially from our experimental setup.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 82 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*. Version 4.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large

language models. *arXiv preprint arXiv:2108.07732*. Jacob Austin and Augustus Odena contributed equally.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*. Version 2.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*. Version 2.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *Proceedings of the International Conference on Learning Representations*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, Andy Zou, and 1 others. 2021. A framework for few-shot language model evaluation.

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-Baptiste Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 81 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*. Version 1.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *arXiv preprint arXiv:2308.11462*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *Proceedings of the International Conference on Learning Representations*.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 13081–13089.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS Symposium on Operating Systems Principles*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy,

Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*. Version 1.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, and 31 others. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*. Previously available as arXiv:2211.09110.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware weight quantization for LLM compression and acceleration. In *Proceedings of the Conference on Machine Learning and Systems*. Best Paper Award.

Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Xiwen Zhang, Nicholas D. Lane, and Mengwei Xu. 2024. Small language models: Survey, measurements, and insights. *arXiv preprint arXiv:2409.15790*. Version 3.

Meta AI. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. Meta AI Blog.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*. Version 3.

Nghiem Thanh Pham, Tung Kieu, Duc-Manh Nguyen, Son Ha Xuan, Nghia Duong-Trung, and Danh Le-Phuoc. 2025. SLM-bench: A comprehensive benchmark of small language models on environmental impacts—extended version. *arXiv preprint arXiv:2508.15478*. Extended version of the EMNLP 2025 Findings paper; Version 2.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pol G. Recasens, Yue Zhu, Chen Wang, Eun Kyung Lee, Olivier Tardieu, Alaa Youssef, Jordi Torres, and Josep Ll. Berral. 2024. Towards pareto optimal throughput in small language model serving. In *Proceedings of the European Conference on Machine Learning and Systems*, pages 95–102.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*. Version 4; accepted at the NeurIPS 2019 Workshop on Energy Efficient Machine Learning and Cognitive Computing.

Victor Schmidt, Karan Goyal, Akshay Joshi, Benjamin Feld, Logan Conell, Nikolas Laskaris, Daniel Blank, Jordan Wilson, Sorelle A. Friedler, and Sasha Luccioni. 2021. Codecarbon: Estimate and track carbon emissions from machine learning computing.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650.

Vast.ai. 2025. Vast.ai — marketplace for gpu instances and compute. `https://vast.ai/`.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the International Conference on Learning Representations*.

Xiaoxia Wu, Cheng Li, Reza Yazdani Aminabadi, Zhewei Yao, and Yuxiong He. 2023. Understanding INT4 quantization for transformer models: Latency speedup, composability, and failure cases. In *Proceedings of the International Conference on Machine Learning*.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the International Conference on Machine Learning*. First two authors contributed equally.

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, and 24 others. 2025. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*. Version 2.

Jie You, Jae-Won Chung, and Mosharaf Chowdhury. 2023. Zeus: Understanding and optimizing gpu energy consumption of dnn training. In *Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation*, pages 119–139, Boston, MA, USA. USENIX Association.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

A Software Environment and Dependencies

All experiments were conducted in a controlled Python environment with pinned library versions to ensure reproducibility. Core dependencies directly relevant to model inference, quantization, evaluation, and profiling include:

Deep Learning and Inference Stack:

- PyTorch 2.8.0 with CUDA 12.8
- vLLM 0.11.0 (inference engine)
- transformers 4.55.2
- huggingface-hub 0.36.0

Quantization and Evaluation:

- llmcompressor 0.7.1 (GPTQ-based INT8 and INT4)
- lm-evaluation-harness 0.4.x

Monitoring and Analysis:

- NVIDIA NVML via pynvml
- nvidia-smi (GPU telemetry)
- numpy 2.2.6, pandas 2.3.3
- matplotlib 3.10.7, seaborn 0.13.2

B Detailed Experimental Configuration

B.1 Inference Configuration

All models were evaluated using greedy decoding with the following parameters:

- Temperature: 0.0
- Top-p: 1.0
- Maximum context length: 8192 tokens
- Maximum generation length: 256 tokens
- GPU memory utilization: 0.9

Batch sizes were task-dependent to reflect typical evaluation practice:

- MMLU: batch size 1
- GSM8K: batch size 32
- HellaSwag: batch size 32

B.2 Reproducibility Controls

To minimize variance and ensure reproducibility:

- Fixed random seed (42) for all runs
- Deterministic CUDA operations enabled
- Three independent runs per configuration with mean values reported
- Cache warm-up: 10 preliminary queries before measurement
- Idle cooling period: 60 seconds between experiments for thermal stabilization
- All experiment logs, configurations, and raw results are version-controlled

B.3 Statistical Testing Protocol

Statistical significance was assessed using the following procedures:

Efficiency Metrics (Latency, Throughput, Energy). Paired t-tests were conducted comparing each quantized configuration (INT8, INT4) against the corresponding FP16 baseline across three independent runs. We report two-tailed p-values with significance thresholds of $\alpha = 0.05$ (*) and $\alpha = 0.01$ (**). Given the controlled experimental conditions and deterministic GPU execution, residual variance primarily reflects thermal state variations and system-level noise rather than algorithmic stochasticity.

Accuracy Metrics (MMLU, GSM8K, HellaSwag). Accuracy differences were evaluated using exact binomial tests on per-sample correctness, treating each benchmark query as an independent trial. For MMLU (57 subjects) and GSM8K ($\sim$ 8K samples), statistical power is sufficient to detect differences as small as $1 - 2$ percentage points. HellaSwag evaluation uses normalized log-probabilities, for which we report exact-match accuracy differences.

Multiple Comparisons Correction. Given the exploratory nature of this systematic evaluation, we report uncorrected p-values to avoid excessive Type II errors (false negatives). We acknowledge that with 13 configurations and 3 quantization schemes, multiple comparison adjustments (e.g., Bonferroni correction) would yield more conservative thresholds (~ 0.004 for $\alpha = 0.05$). However, the consistency of patterns across models (e.g., GSM8K

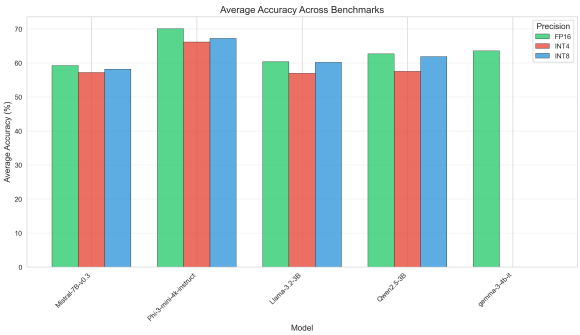

Figure 6: Task-wise accuracy comparison across all model-precision configurations.

consistently showing larger degradation) suggests effects are robust to correction procedures.

Effect Size Reporting. In addition to p-values, we report absolute differences (percentage points for accuracy, seconds for latency, kWh for energy) to enable readers to assess practical significance independent of statistical significance.

C Complete Experimental Results

Table 5 presents the complete experimental results for all 13 model-precision configurations evaluated, including per-task accuracies (MMLU, GSM8K, HellaSwag), latency, throughput, memory usage, energy consumption, and model size.

D Extended Performance Analysis

D.1 Task-Specific Accuracy Comparison

Figure 6 shows task-wise accuracy comparison across all model-precision configurations, demonstrating the differential impact of quantization on factual knowledge (MMLU), mathematical reasoning (GSM8K), and commonsense reasoning (HellaSwag).

D.2 Accuracy Heatmap Visualization

Figure 7 provides a heatmap visualization of accuracy values across all model-task-precision combinations, enabling quick identification of quantization-sensitive configurations.

D.3 Task-Specific Accuracy Degradation

Figure 8 visualizes accuracy degradation by task under INT8 and INT4 quantization, highlighting the $3-10\times$ heightened sensitivity of mathematical reasoning (GSM8K) compared to factual knowledge (MMLU) and commonsense reasoning (HellaSwag).

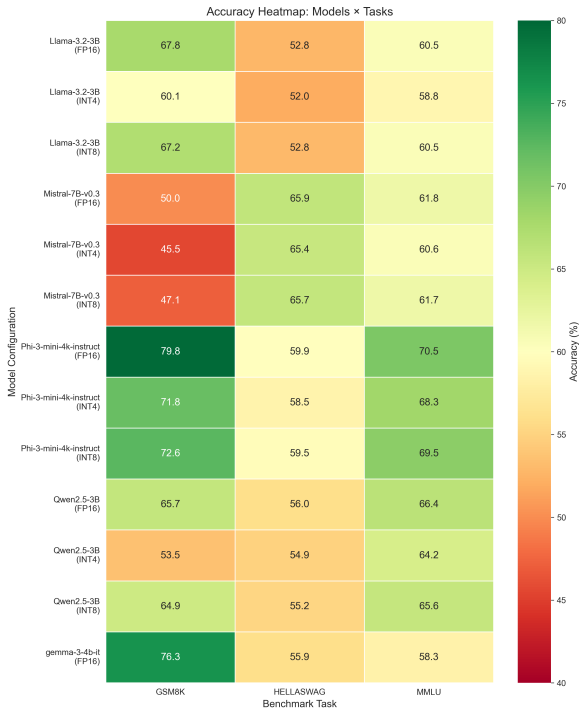

Figure 7: Heatmap of accuracy values across model-task-precision combinations.

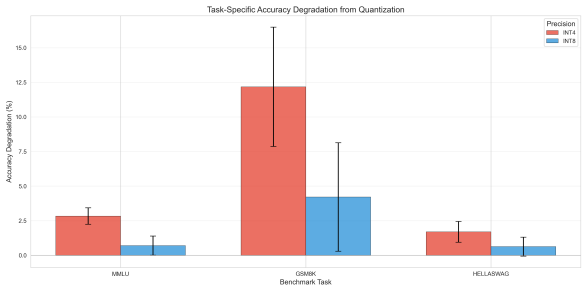

Figure 8: Task-specific accuracy degradation under INT8 and INT4 quantization.

E Throughput and Latency Analysis

E.1 Throughput Comparison

Figure 9 compares inference throughput (tokens per second) across all model-precision configurations, showing that INT4 generally achieves the highest throughput within each model family, with Llama-3.2-3B (INT4) reaching 1936 tokens/second.

E.2 Latency Distribution

Figure 10 shows the distribution of per-query latencies across all configurations, demonstrating that Llama-3.2-3B (INT8) achieves the lowest mean latency (99 ms) while maintaining high accuracy.

Model	Precision	Size (GB)	Latency (s)	Tokens/s	Energy (kWh)	MMLU	GSM8K	HellaSwag	Avg Acc
Mistral-7B	FP16	13.50	0.183	715	0.0384	0.618	0.500	0.659	0.592
Mistral-7B	INT8	7.51	0.126	989	0.0211	0.617	0.471	0.657	0.582
Mistral-7B	INT4	4.45	0.113	1421	0.0144	0.606	0.455	0.654	0.572
Phi-3-mini	FP16	7.12	0.238	1077	0.0214	0.705	0.798	0.599	0.701
Phi-3-mini	INT8	4.11	0.180	1425	0.0116	0.695	0.726	0.595	0.672
Phi-3-mini	INT4	2.53	0.236	1084	0.0106	0.683	0.718	0.585	0.662
Llama-3.2-3B	FP16	5.98	0.215	1189	0.0210	0.605	0.678	0.528	0.604
Llama-3.2-3B	INT8	5.56	0.099	1517	0.0136	0.605	0.672	0.528	0.602
Llama-3.2-3B	INT4	4.33	0.132	1936	0.0107	0.588	0.601	0.520	0.569
Qwen2.5-3B	FP16	5.75	0.146	1186	0.0174	0.664	0.657	0.560	0.627
Qwen2.5-3B	INT8	4.91	0.148	812	0.0152	0.656	0.649	0.552	0.619
Qwen2.5-3B	INT4	1.13	0.153	1583	0.0113	0.642	0.535	0.549	0.575
Gemma-3-4B	FP16	8.64	0.260	966	0.0267	0.583	0.764	0.559	0.635

Table 5: Complete experimental results for all model-precision configurations. Values represent means across 3 independent runs. For statistical significance of quantization impact, see Table 3 in the main text.

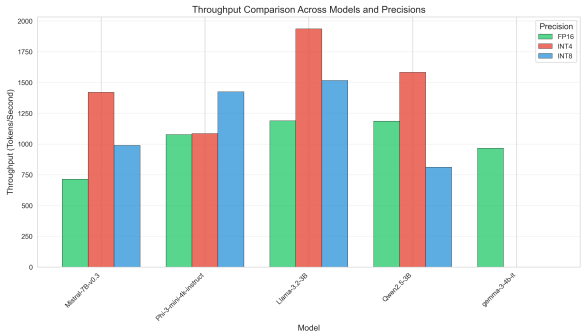

Figure 9: Throughput comparison (tokens/s) across model-precision configurations.

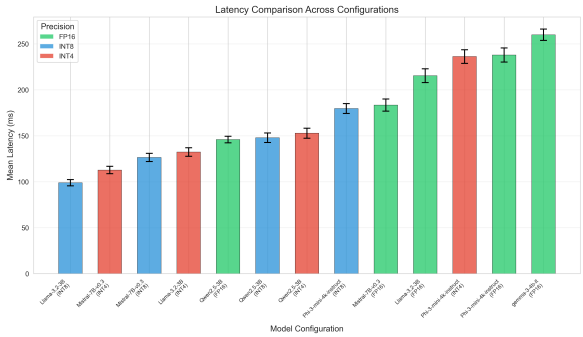

Figure 10: Latency distribution across model-precision configurations.

E.3 Latency vs. Memory Trade-off

Figure 11 illustrates the relationship between latency and memory usage for all configurations, revealing that Llama-3.2-3B (INT8) achieves Pareto-optimal performance by minimizing latency without substantial memory overhead.

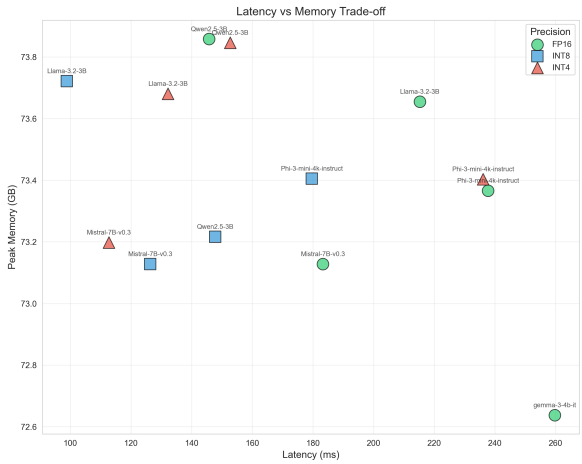

Figure 11: Latency versus memory usage for all configurations.

F Energy Consumption Analysis

F.1 Total Energy Consumption

Figure 13 presents total energy consumption (kWh) for each model-precision configuration across the complete benchmark evaluation, showing that INT4 quantization provides the strongest energy reductions (up to 63%).

F.2 Average Power Consumption

Figure 14 shows average GPU power draw (W) during inference for each configuration, demonstrating that quantization reduces power consumption by $12 - 46\%$ depending on model architecture.

G Multi-Dimensional Efficiency Comparison

Figure 12 provides radar charts comparing efficiency dimensions (accuracy, latency, throughput, energy, memory) across FP16, INT8, and INT4

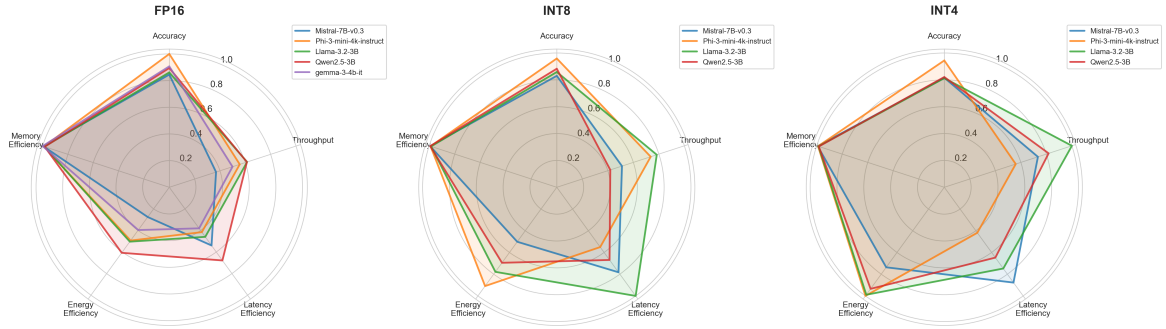

Figure 12: Radar chart comparing efficiency dimensions across configurations.

Metric	FP16	INT8	INT4
Latency (s)	0.208 ± 0.045	0.138 ± 0.034	0.158 ± 0.054
Throughput (tok/s)	1027 ± 197	1186 ± 339	1506 ± 354
Energy (kWh)	0.025 ± 0.008	0.015 ± 0.004	0.012 ± 0.002
Avg Power (W)	257 ± 19	195 ± 35	196 ± 41

Table 6: Summary statistics by precision mode (mean $\pm$ standard deviation).

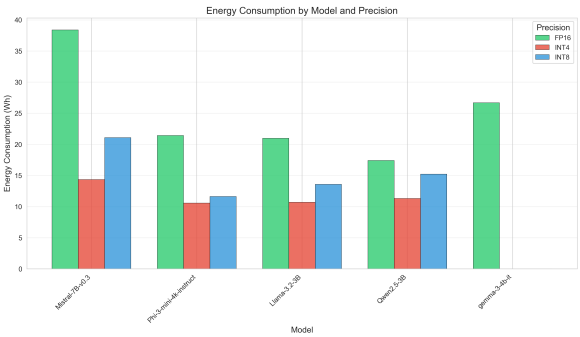

Figure 13: Total energy consumption (kWh) for each model-precision configuration.

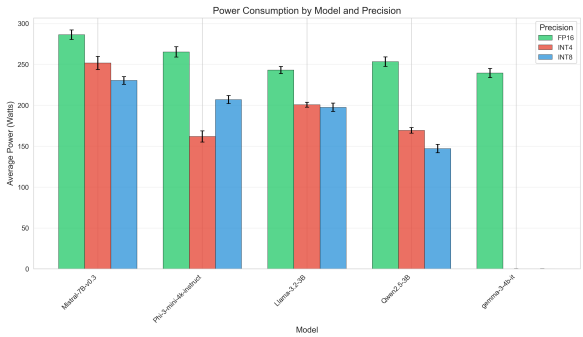

Figure 14: Average power consumption (W) during inference for each configuration.

precision modes, enabling holistic assessment of quantization trade-offs.

H Summary Statistics by Precision Mode

Table 6 provides aggregated statistics (mean $\pm$ standard deviation) across all models for each precision mode, summarizing the overall impact of quantization on latency, throughput, energy consumption, and average power draw.

I Energy Measurement Methodology

GPU energy consumption was measured by integrating instantaneous GPU power draw sampled at 100 ms intervals via NVIDIA NVML APIs. To isolate model-specific energy usage:

- **Idle Baseline Measurement:** An idle baseline was measured before each experiment under matched system conditions (same GPU, driver version, ambient temperature)
- **Active Measurement:** GPU power was sampled continuously during benchmark execution
- **Baseline Subtraction:** Total consumption = Active measurement - Idle baseline
- **Per-Query Normalization:** Energy per query = Total consumption / Number of queries

This methodology ensures reported energy values reflect net inference energy attributable to model execution only, excluding system-level overhead.

J Quantization Compatibility Notes

Certain model configurations could not be evaluated under INT8 or INT4 precision due to incom-

patibilities with the quantization pipeline or serving engine at the time of experimentation:

- **Gemma-3-4B-it:** Evaluated only in FP16 due to quantization compatibility constraints with llm-compressor 0.7.1

These exclusions are purely technical and do not reflect model quality. All reported FP16 results for these models follow identical evaluation protocols as other configurations and are included to provide fair baseline comparisons.

K Artifact Availability

The SLiM-Eval framework, including evaluation pipeline code, quantization configurations, raw experimental data, analysis scripts, and full reproducibility documentation, is publicly available at: https://anonymous.4open.science/r/SLiM-Eval-C941