ASQ: AN IMPROVED APPROACH TO UNIFYING SPARSIFICATION AND QUANTIZATION OF LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparsification and quantization are widely adopted for the efficient deployment of large language models (LLMs). However, they are often optimized independently, overlooking their intricate interactions and yielding suboptimal compression trade-offs. To address this, we propose **ASQ**, an ADMM-based framework that jointly optimizes sparsification and quantization under a unified formulation, supported by theoretical convergence guarantees. At its core, **ASQ** admits an exact, closed-form joint projection that concurrently resolves sparse support and quantized values by accounting for both their importance to the objective criterion and their representability on a quantization grid. Across multiple model architectures and scales, **ASQ** substantially outperforms existing joint compression methods under aggressive compression, with the performance gap widening significantly under extreme sparsity and low-bit regimes. Moreover, under hardware-compatible 2:4 sparsity with INT4 quantization, **ASQ** achieves a 4.78× memory reduction and up to 3.20× end-to-end decode speedup on Qwen3-8B-Base relative to dense BF16 inference.

1 INTRODUCTION

While large language models (LLMs) have achieved remarkable success across diverse language tasks, their practical deployment is significantly constrained by substantial memory footprints and steep inference costs. To alleviate these constraints, model compression has emerged as an indispensable paradigm, with sparsification (or pruning) and quantization serving as the de facto standards. Specifically, sparsification induces sparsity by removing redundant parameters, whereas quantization reduces numerical precision by representing parameters with fewer bits. Although both techniques have been extensively studied (Han et al., 2015; Hoefler et al., 2021; Gholami et al., 2022) and successfully adapted to LLMs (Frantar & Alistarh, 2023; Sun et al., 2024; Lee et al., 2026; Frantar et al., 2023; Lin et al., 2024), relying on either approach in isolation exhibits severe performance degradation at high compression regimes (Zhao et al., 2025; Lee et al., 2026).

To overcome these individual limitations and achieve higher compression, recent efforts combine sparsification and quantization, simultaneously reducing parameter count and numerical precision (Frantar & Alistarh, 2022; Guo et al., 2024; Mozaffari et al., 2025; Guo et al., 2026). However, unifying the two paradigms is fundamentally non-trivial due to their complex interactions. In fact, the two objectives impose inherently conflicting demands: while sparsification tends to preserve salient, large-magnitude weights, quantization relies on a compact numerical range, meaning weights favored by pruning can severely penalize quantization fidelity (Guo et al., 2024; 2026). Indeed, prior studies reveal that their errors compound each other in an order-dependent manner, often inducing performance degradation that far exceeds the sum of their isolated effects (Harma et al., 2025; Kim et al., 2026). Consequently, optimizing sparsification and quantization in a decoupled manner artificially constricts the viable solution space, yielding suboptimal compression results under joint constraints.

In this work, we propose **ASQ**, an optimization framework based on the **A**lternating Direction Method of Multipliers (Boyd et al., 2011) designed for joint **S**parsification and **Q**uantization of LLMs. We cast joint compression directly into a constrained optimization problem governed by a unified sparse-quantized feasible set, where every parameter is either eliminated or mapped to a low-bit grid. A core technical merit of **ASQ** is that its constraint step admits an exact, closed-form projection onto the joint feasible set, thereby resolving the sparse mask and quantization assignments concurrently without

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

heuristic ordering. To improve practical fidelity, we extend this formulation with sensitivity-aware parameter weighting and quantization scale optimization. We further prove that **ASQ** is guaranteed to converge under suitable optimization conditions.

Our empirical study validates that **ASQ** effectively bridges the gap between aggressive compression and model capability. As the compression ratio increases, existing joint compression methods often suffer sharp performance degradation, whereas **ASQ** remains substantially more robust across model families and scales. For instance, on Qwen3-8B-Base, **ASQ** achieves perplexities of 15.10 and 36.18 at 70% and 90% sparsity with 4-bit quantization, whereas the strongest baseline reaches 20.13 and 387.90. Furthermore, **ASQ** remains effective under hardware-compatible settings, with 2:4 sparsity and INT4 quantization on Qwen3-8B-Base yielding a 4.78× memory saving and up to 3.20× acceleration in end-to-end decode performance over dense BF16 baselines.

2 Background

Sparsification, commonly referred to as pruning, compresses models by identifying and removing redundant parameters while preserving model performance (Han et al., 2016; Hoefler et al., 2021). With the rise of LLMs, various post-training and one-shot pruning methods have emerged, successfully identifying salient parameters using magnitude- or Hessian-based criteria (Sun et al., 2024; Frantar & Alistarh, 2023), with subsequent efforts pushing these approaches toward increasingly higher sparsity regimes (Meng et al., 2024; Bai et al., 2024; Lee et al., 2026). However, relying solely on sparsification to achieve aggressive compression remains inherently challenging. As sparsity is pushed to higher regimes, parameter removal inevitably damages critical linguistic representations and complex reasoning abilities, leading to steep performance degradation. Consequently, sparsification alone exhibits clear limits in unlocking high compression ratios without unacceptable fidelity loss.

Alongside sparsification, quantization has become an indispensable paradigm to reduce memory footprint and latency by mapping continuous weights to discrete, low-bit representations (Gholami et al., 2022). For LLMs, post-training quantization has enabled substantial precision reductions without prohibitive retraining, with landmark methods such as GPTQ (Frantar et al., 2023) and AWQ (Lin et al., 2024) mitigating quantization error via second-order error compensation and activation-aware scaling, respectively. Subsequent studies have further enhanced low-precision regimes by suppressing activation outliers (Xiao et al., 2023) and transforming representations via orthogonal rotations to be more quantization-friendly (Liu et al., 2025; Liang et al., 2026). Nevertheless, relying exclusively on quantization hits a steep performance cliff in ultra-low-bit regimes, where discrete rounding errors severely degrade model capacity (Zhao et al., 2025).

To surpass the performance ceilings of extreme sparsity and ultra-low bit-widths, combining sparsification and quantization offers a compelling avenue to scale compression across both structural and precision axes. The conventional paradigm applies these two procedures sequentially, a strategy originating from classical compression pipelines (Han et al., 2016) that remains attractive due to its modularity. However, recent studies reveal that such sequential pipelines suffer from severe order dependency: approximation errors incurred in the initial stage compound irreversibly into the second, leading to dramatic disparities in final model quality (Harma et al., 2025; Kim et al., 2026).

Beyond naive sequential pipelines, recent work has sought tighter integration between the two domains. For instance, Optimal Brain Compression (OBC; Frantar & Alistarh (2022)) provides a generalized second-order framework applicable to both sparsification and quantization. More recently, JSQ (Guo et al., 2024) introduces an activation-range-aware metric to prime weights for subsequent quantization, SLiM (Mozaffari et al., 2025) employs low-rank adapters to absorb composite compression errors, and OBR (Guo et al., 2026) utilizes second-order compensation to align both axes.

While these methods highlight the interdependence of sparsification and quantization, they predominantly rely on predetermined execution orders, stage-wise heuristics, or post-hoc error compensation. Crucially, enforcing discrete decisions along one dimension prior to the other results in a premature commitment that artificially constricts the feasible solution space, precluding the discovery of optimal configurations under the joint constraint. This fundamental drawback underscores the need for a principled optimization framework that natively co-optimizes sparsification and quantization without heuristic orderings.

3 METHOD

3.1 PROBLEM FORMULATION

Let $x \in \mathbb{R}^n$ denote the model parameters subject to compression and let $f(x)$ denote the objective function to minimize. We formulate joint sparsification and quantization as follows:

$$\min_x \ f(x) \quad \text{s.t.} \quad x \in \mathcal{S} \cap \mathcal{Q}, \tag{1}$$

where $\mathcal{S} = \{x : \|x\|_0 \leq k\}$ enforces the sparsity constraint with $k = \lfloor n(1-s) \rfloor$ for a target sparsity ratio s, and $\mathcal{Q} = \{x : x_i \in \mathcal{G},\ \forall i \in \{1, \ldots, n\}\}$ restricts each weight to a given finite quantization grid $\mathcal{G}$. Consequently, a solution to (1) simultaneously determines the optimal sparse support and the quantized values of the retained parameters, effectively circumventing the suboptimality of sequential compression pipelines while minimizing model degradation.

3.2 ADMM FOR JOINT COMPRESSION

Directly solving (1) is computationally intractable. The intersection $\mathcal{S} \cap \mathcal{Q}$ imposes a non-convex, combinatorial constraint set, precluding standard continuous optimization at LLM scale. We address this problem through the Alternating Direction Method of Multipliers (ADMM; Boyd et al., 2011), a variable-splitting framework that decomposes constrained optimization into simpler subproblems.

Precisely, this introduces an auxiliary variable z to decouple the loss and the constraint:

$$\min_{x,\,z} \ f(x) + I_{\mathcal{S}\cap\mathcal{Q}}(z) \quad \text{s.t.} \quad x = z, \tag{2}$$

where the indicator function encodes the joint constraint:

$$I_{\mathcal{S}\cap\mathcal{Q}}(z) = \begin{cases} 0 & z \in \mathcal{S} \cap \mathcal{Q}, \\ \infty & \text{otherwise.} \end{cases} \tag{3}$$

The augmented Lagrangian of (2) is

$$\mathcal{L}_\lambda(x, z, u) = f(x) + I_{\mathcal{S}\cap\mathcal{Q}}(z) + \tfrac{\lambda}{2}\|x - z + u\|^2 - \tfrac{\lambda}{2}\|u\|^2, \tag{4}$$

where $\lambda > 0$ controls the strength of the proximal penalty and u is the scaled dual variable. ADMM alternates between minimizing (4) over the primal variables x and z, and updating u via a dual ascent step:

$$x^{t+1} = \operatorname*{argmin}_x \ f(x) + \tfrac{\lambda}{2}\|x - z^t + u^t\|^2 \tag{5}$$

$$z^{t+1} = \operatorname*{argmin}_{z\in\mathcal{S}\cap\mathcal{Q}} \|z - v^t\|^2, \quad \text{where } v^t = x^{t+1} + u^t \tag{6}$$

$$u^{t+1} = u^t + x^{t+1} - z^{t+1}. \tag{7}$$

The x-update minimizes the training objective augmented with a proximal penalty, and the u-update performs dual ascent to update the Lagrange multipliers. A key advantage of this variable splitting is that the discrete combinatorial nature of joint compression is strictly confined to the z-update in (6). Letting $\mathcal{C} := \mathcal{S} \cap \mathcal{Q}$ denote the joint sparse-and-quantized feasible set, the z-update corresponds to the Euclidean projection:

$$\Pi_{\mathcal{C}}(v) = \operatorname*{argmin}_{z\in\mathcal{C}} \|z - v\|^2. \tag{8}$$

Thus, enforcing the joint constraint reduces to projecting v onto the sparse–quantized feasible set $\mathcal{C}$. Although this projection simultaneously couples both sparsity and discrete quantization, we show in the following section that it admits an exact, closed-form solution that can be computed efficiently.

3.3 JOINT SPARSE–QUANTIZED PROJECTION

We now derive an exact solution to the joint projection in (8) for a fixed quantization grid $\mathcal{G}$. For each coordinate v_i, if it is retained, its optimal projection onto the non-zero grid is given by:

$$\hat{q}_i \in \operatorname*{argmin}_{g\in\mathcal{G}\setminus\{0\}} (v_i - g)^2. \tag{9}$$

We then define the retention gain G_i as:

$$G_i = \underbrace{v_i^2}_{\text{if pruned}} - \underbrace{(v_i - \hat{q}_i)^2}_{\text{if quantized}}, \tag{10}$$

which quantifies the reduction in projection error obtained by retaining coordinate i at its nearest quantized level rather than pruning it to zero. Notably, G_i naturally generalizes classical magnitude-based pruning: in the absence of quantization error ($\hat{q}_i = v_i$), G_i reduces to v_i^2, whereas in practice it penalizes coordinates that suffer from severe quantization distortion.

For any feasible support $I \subseteq \{1, \ldots, n\}$ with $|I| \leq k$, let $z(I) \in \mathbb{R}^n$ denote the parameter vector setting $z(I)_i = \hat{q}_i$ for $i \in I$ and $z(I)_i = 0$ otherwise. The resulting projection error can be decomposed as:

$$\|z(I) - v\|_2^2 = \sum_{i \notin I} v_i^2 + \sum_{i \in I} (v_i - \hat{q}_i)^2 = \sum_{i=1}^{n} v_i^2 - \sum_{i \in I} G_i. \tag{11}$$

Because $\sum_{i=1}^{n} v_i^2$ is independent of the support choice I, minimizing the joint Euclidean projection error is equivalent to maximizing the aggregate retention gain over feasible supports. This yields the following exact solution:

Proposition 3.1 (Exact Joint Sparse–Quantized Projection). *For a given $v \in \mathbb{R}^n$, let $\hat{q}_i$ and G_i be defined by (9) and (10), respectively. Then, the projection of v onto the joint feasible set $\mathcal{C}$ is given by:*

$$z_i^\star = \begin{cases} \hat{q}_i, & \text{if } i \in I^\star, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where the optimal support $I^\star$ corresponds to the indices of the up to k largest positive retention gains:

$$I^\star \in \operatorname*{argmax}_{I \subseteq \{1,\ldots,n\},\, |I| \leq k} \sum_{i \in I} G_i. \tag{13}$$

A complete proof is provided in Appendix A. Proposition 3.1 establishes that the joint projection onto the non-convex discrete set $\mathcal{C}$ can be solved exactly through coordinate-gain computation and top-k selection, without requiring heuristic greedy iterations or exhaustive search. Crucially, unlike sequential pipelines, this formulation couples sparsity with quantization fidelity into a single metric, pruning weights not merely by magnitude, but by their net contribution under discrete representability.

3.4 Quantization Scale Optimization

The exact projection in Section 3.3 determines the optimal sparse support for a given quantization grid. In low-bit quantization, the grid is parameterized by a scale Δ, so the support and quantization scale are naturally coupled. We capture this interaction through the joint projection objective

$$\mathcal{J}(I, \Delta; v) = \sum_{i \notin I} v_i^2 + \sum_{i \in I} \big(v_i - Q_\Delta(v_i)\big)^2, \qquad |I| \leq k. \tag{14}$$

For a fixed scale Δ, minimizing (14) over I is exactly the joint projection characterized by Proposition 3.1. For a fixed support I, we update the scale by minimizing the quantization error over the retained coordinates:

$$\Delta^\star \in \operatorname*{argmin}_{\Delta \in \mathcal{D}} \sum_{i \in I} \big(v_i - Q_\Delta(v_i)\big)^2, \tag{15}$$

where $\mathcal{D}$ is a finite set of candidate scales.

ASQ applies these two block updates iteratively: the support is updated exactly under the current quantization grid, and the scale is then adapted to the retained weights. Each update minimizes the same joint projection objective with respect to one block while holding the other fixed, thereby coordinating sparse support selection and quantization-scale adaptation.

3.5 Sensitivity-Weighted Joint Projection

While the isotropic Euclidean projection in (8) treats all coordinates uniformly, individual parameters exhibit vastly disparate influence on the task loss. Motivated by second-order Taylor approximations of the model objective (LeCun et al., 1989), where perturbation error is governed by parameter curvature, we replace the uniform metric with a Hessian-weighted quadratic form:

$$z^{t+1} \in \operatorname*{argmin}_{z \in \mathcal{C}} (z - v^t)^\top H (z - v^t). \tag{16}$$

Directly computing or storing the full Hessian H is computationally intractable at LLM scale and would destroy the coordinate-wise separability essential for our exact projection. To reconcile curvature awareness with computational tractability, we restrict H to its diagonal empirical Fisher information matrix (Martens, 2020):

$$H \approx \operatorname{diag}\left(\hat{F}_{11}, \ldots, \hat{F}_{nn}\right), \qquad \hat{F}_{ii} = \mathbb{E}[g_i^2]. \tag{17}$$

Crucially, this diagonal approximation incurs zero additional computational overhead: we directly repurpose the running second-moment gradient statistics natively tracked by the Adam optimizer as an efficient proxy for $\hat{F}_{ii}$ (Li et al., 2025). Under this diagonal metric, the objective decouples element-wise, seamlessly extending the alternating optimization presented earlier. Specifically, for a fixed quantization scale Δ, the support is selected by maximizing the sensitivity-weighted retention gain:

$$I^\star \in \operatorname*{argmax}_{|I| \le k} \sum_{i \in I} \hat{F}_{ii} \left[v_i^2 - (v_i - \hat{q}_i)^2 \right]. \tag{18}$$

Conversely, for a fixed support I, the scale is updated by minimizing the weighted quantization distortion:

$$\Delta^\star \in \operatorname*{argmin}_{\Delta \in \mathcal{D}} \sum_{i \in I} \hat{F}_{ii} \big(v_i - Q_\Delta(v_i)\big)^2. \tag{19}$$

Thus, the proposed weighting incorporates parameter-wise sensitivity into both support selection and scale optimization while preserving the separable structure of the joint projection.

3.6 Convergence Analysis

We next analyze the convergence of ASQ under exact projection onto the joint sparse-and-quantized feasible set. Since the quantization grid G is finite, the joint feasible set $\mathcal{C}$ is also finite. Hence, Euclidean projection onto $\mathcal{C}$ is always attained and $f_{\min} := \min_{z \in \mathcal{C}} f(z)$ is guaranteed to exist. We make a standard regularity assumption on the objective function f.

Assumption 3.1 (Smoothness). *f is continuously differentiable with a β-Lipschitz continuous gradient: $\|\nabla f(x) - \nabla f(y)\|_2 \le \beta \|x - y\|_2$ for all $x, y \in \mathbb{R}^n$.*

Since the x-update (5) is solved only approximately in practice, we require only a relative accuracy condition on it. Let $\phi_t(x) := f(x) + \frac{\lambda}{2}\|x - z^t + u^t\|_2^2$ denote its objective.

Assumption 3.2 (Inexact x-update). *There exists a constant $\varepsilon \ge 0$ such that, for every $t \ge 0$,*

$$\left\|\nabla \phi_t(x^{t+1})\right\|_2 \;\le\; \varepsilon \left\|x^{t+1} - z^t\right\|_2. \tag{20}$$

In nonconvex constrained optimization, stationary points can be characterized via projected-gradient fixed points.

Definition 3.1 (λ-stationary point). A point $\bar{x} \in \mathcal{C}$ is a λ-stationary point of (1) if $\bar{x} \in \Pi_{\mathcal{C}}\big(\bar{x} - \lambda^{-1}\nabla f(\bar{x})\big)$, which corresponds to a fixed point of a gradient step followed by projection onto $\mathcal{C}$.

Theorem 3.1 (Convergence of ASQ). *Suppose Assumptions 3.1 and 3.2 hold. If the penalty parameter $\lambda > 0$ is chosen sufficiently large such that $\lambda > 5(\beta + \varepsilon)$ then the ADMM iterates satisfy:*

1. ***Successive-step convergence:*** *$\sum_{t=1}^{\infty} \|x^{t+1} - x^t\|_2^2 < \infty$ and $\|x^{t+1} - z^{t+1}\|_2 \to 0$ as $t \to \infty$.*
2. ***Finite-time identification:*** *There exist a finite iteration index $T_0 < \infty$ and a point $\bar{z} \in \mathcal{C}$ such that $z^t = \bar{z}$ for all $t \ge T_0$.*
3. ***Stationarity:*** *$x^t \to \bar{z}$ as $t \to \infty$, and $\bar{z}$ is a λ-stationary point of (1).*

Crucially, the finite-time identification property ensures that the discrete compression structure—both the sparse support and the quantized levels—stabilizes after finitely many iterations, ruling out persistent switching among distinct sparse–quantized configurations.

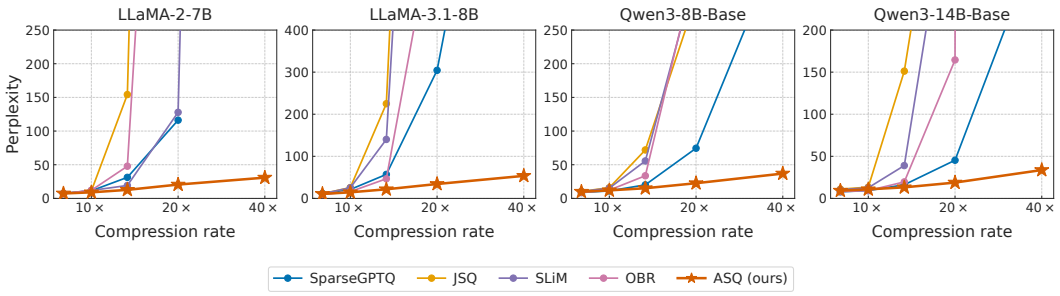


Figure 1: WikiText-2 perplexity (↓) under joint sparsification and 4-bit quantization across model families, shown against the compression rate relative to dense 16-bit. Across all four model families, **ASQ** maintains substantially lower perplexity under aggressive compression, while competing methods degrade sharply.

Proof sketch. Under the theorem's condition, a potential function decreases sufficiently at each iteration. Combined with the finite separation between distinct points of $\mathcal{C}$, this implies that the discrete iterate becomes fixed after finitely many iterations. The full proof is provided in Appendix B.

4 Experiments

In this section, we evaluate **ASQ** under various joint compression settings and compare it with representative joint compression baselines. We first assess its effectiveness on preserving model fidelity across different model families and increasingly aggressive compression rates (Section 4.1). We then analyzes how the continuous parameters and discrete sparse–quantized structure progressively align and stabilize during **ASQ** optimization (Section 4.2). Next, we conduct ablation studies to isolate the contributions of the key components of **ASQ** (Section 4.3). Finally, we evaluate its practical utility under hardware-compatible compression constraints through end-to-end inference measurements (Section 4.4).

Setup. We evaluate **ASQ** on widely used LLMs from the LLaMA and Qwen families, LLaMA-2-7B (Touvron et al., 2023), LLaMA-3.1-8B (Grattafiori et al., 2024), Qwen3-8B-Base, and Qwen3-14B-Base (Yang et al., 2025). Following standard evaluation protocols in post-training LLM compression, we evaluate language-modeling quality using perplexity on WikiText-2 (Merity et al., 2017) and C4 (Raffel et al., 2020), and downstream performance using zero-shot accuracy on seven benchmarks: PIQA (Bisk et al., 2020), ARC-Easy and ARC-Challenge (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), RTE (Wang et al., 2019), and WinoGrande (Sakaguchi et al., 2021). All zero-shot evaluations are conducted using `lm-eval-harness` (Gao et al., 2024). Detailed experimental settings, including the calibration data and protocols used for **ASQ** and each baseline, are provided in Appendix C.

Compression metric. To compare sparsity–precision configurations on a common scale, we define the theoretical compression rate relative to dense 16-bit weights as $r_{\text{comp}} = \frac{16}{b(1-s)}$, where b is the quantization bit-width and s is the sparsity ratio. For brevity, we refer to this quantity as the compression rate.

Baselines. We compare **ASQ** with representative methods for joint sparsification and quantization, including SparseGPT+GPTQ (SparseGPTQ; Frantar & Alistarh, 2023), JSQ (Guo et al., 2024), SLiM (Mozaffari et al., 2025), and OBR (Guo et al., 2026).

4.1 Main Results

We first examine how well **ASQ** preserves model quality as joint compression becomes increasingly aggressive. Figure 1 compares **ASQ** with existing joint compression methods under 4-bit quantization as sparsity increases from 50% to 90%, corresponding to compression rates from $8\times$ to $40\times$. Additional numerical results are provided in Appendix D.

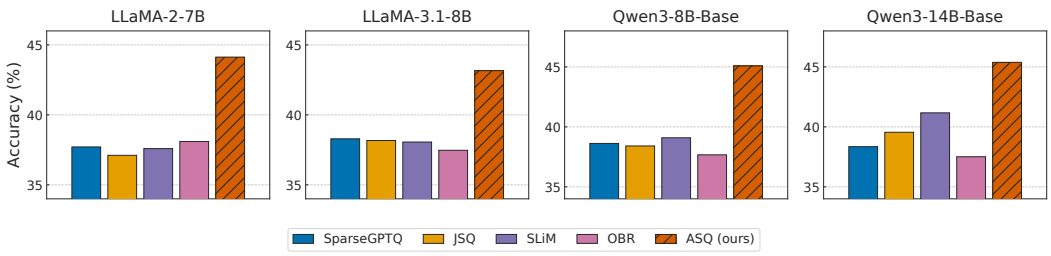


Figure 2: Average zero-shot accuracy (%, ↑), across seven benchmarks at $40\times$ compression ratio across model families. ASQ achieves higher accuracy than competing methods across all four model families, showing that its advantage under aggressive compression extends to downstream tasks.

Across all four model families, **ASQ** remains competitive at moderate compression and degrades more gracefully as the compression rate increases. In contrast, competing methods show sharp increases in perplexity at higher compression rates. For example, on Qwen3-8B-Base, **ASQ** achieves perplexities of 15.10 and 36.18 at $13.3\times$ and $40\times$ compression, respectively, whereas the strongest competing method reaches 20.13 and 387.90 under the same settings. Notably, the advantage of **ASQ** becomes more pronounced as the compression rate increases. This trend highlights the benefit of resolving sparse support and quantized values jointly, especially when the feasible compressed representation becomes increasingly constrained.

We next examine whether this advantage also extends to downstream task performance. Figure 2 reports the average zero-shot accuracy across seven benchmarks at $40\times$ compression, the most aggressive compression setting considered in Figure 1. **ASQ** consistently outperforms the strongest competing method across the evaluated models, with gains ranging from 4.21 to 6.03 percentage points in average accuracy. This shows that the advantage observed in perplexity is not limited to language-modeling loss, but also translates to better preservation of downstream performance under aggressive compression.

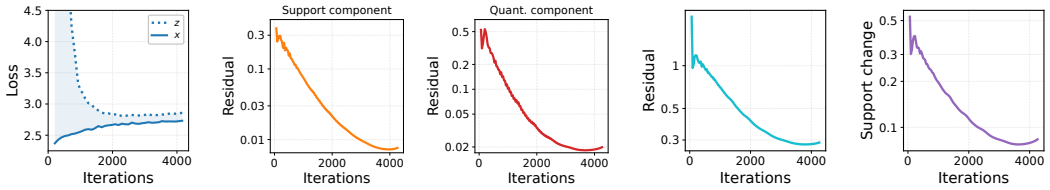
4.2 Optimization Dynamics of ASQ

We next investigate how the joint sparse–quantized representation develops over the course of **ASQ** optimization. Recall that the ADMM formulation enforces agreement between the continuous primal variable x and the sparse–quantized split variable z. We first measure how closely these two variables satisfy the coupling constraint $x = z$, using the primal residual. We further decompose this residual into support and quantization components to distinguish residuals on pruned coordinates from quantization discrepancies on retained coordinates. To examine whether the discrete solution stabilizes, we track changes between successive split iterates and changes in the sparse support. The former is measured by the relative dual residual, while the latter is measured by the normalized Hamming distance between successive support masks. Exact metric definitions are provided in Appendix C.2.

Figure 3 reveals a progressive alignment between the continuous parameters and the joint compressed representation. Both the support and quantization components of the primal residual decrease throughout optimization (Figure 3b), indicating increasing compatibility with the joint constraint. Consistently, the validation losses of the dense and sparse–quantized iterates also become increasingly close (Figure 3a). This suggests that the sparse–quantized representation increasingly preserves the predictive objective of the continuous parameters as optimization proceeds.

The discrete compression structure exhibits a complementary stabilization pattern. Changes in the sparse support become substantially smaller over time (Figure 3d), while the relative dual residual also decreases (Figure 3c). Together, these observations show that the continuous parameters and discrete compression structure progressively become mutually aligned, eventually yielding a stable sparse–quantized representation.

These dynamics provide an empirical view of the optimization mechanism underlying **ASQ**. Rather than determining sparsity and quantization as isolated decisions, **ASQ** progressively organizes the parameter representation around their joint structure.



(a) Loss (b) Primal residual (c) Dual residual (d) Support change

Figure 3: Optimization dynamics of **ASQ** on Qwen3-8B-Base with 4-bit quantization and 70% sparsity. (a) Validation loss at the two ADMM iterates: the primal variable x and the sparse–quantized split variable z. (b) Support and quantization components of the squared relative primal residual. (c) Relative dual residual between successive split iterates. (d) Normalized Hamming distance between successive support masks.

4.3 Ablation Study

We analyze three key components of **ASQ**: joint sparse–quantized projection (JP), quantization-scale optimization (QSO), and sensitivity weighting (SW). We ablate each component by replacing it with a conventional alternative and evaluate all combinations on Qwen3-8B-Base under 4-bit quantization at $8\times$, $13.3\times$, and $40\times$ compression. Details of the ablation variants are provided in Appendix C.3.

Table 1 shows that all three components contribute to the final performance, with clearer differences as compression becomes more aggressive. Removing JP causes a relatively modest degradation at lower compression but a substantially larger drop at $40\times$. This trend highlights the importance of accounting for quantization compatibility when determining the sparse support, rather than fixing the support independently before quantization. QSO provides a smaller but consistent improvement across the evaluated regimes, supporting the benefit of adapting the quantization scale to the current sparse representation. SW also becomes increasingly important at higher compression, indicating that parameter-wise sensitivity matters more as the feasible compressed representation becomes more constrained.

Table 1: Ablation of joint projection (JP), quantization-scale optimization (QSO), and sensitivity weighting (SW) across compression rates. Values are WikiText-2/C4 perplexities.

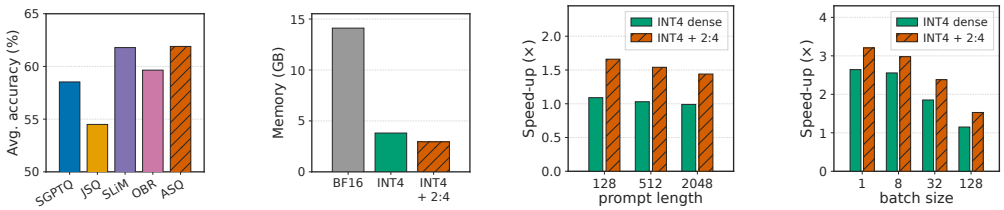
JP	*QSO*	*SW*	Compression rate 8×	13.3×	40×
✓	✓	✓	**9.9/15.5**	**15.1/20.8**	**36.2/37.8**
×	✓	✓	11.0/17.5	17.7/24.1	69.5/65.4
✓	×	✓	10.4/16.6	15.4/21.4	37.1/38.3
✓	✓	×	10.9/17.2	16.6/22.2	49.4/47.4
×	×	✓	11.7/18.2	18.4/25.1	69.9/64.2
×	✓	×	11.8/18.4	18.2/24.8	75.6/69.8
✓	×	×	11.0/17.6	16.6/22.6	63.1/60.8
×	×	×	12.0/18.5	18.9/25.7	83.6/77.7

The combined ablations further show that these components are complementary. Overall, the results indicate that JP plays a central role in coordinating sparsity and quantization, while QSO and SW further improve the fidelity of the resulting sparse–quantized representation.

4.4 Hardware-Compatible Compression and Inference Efficiency

We next examine whether the benefits of **ASQ** extend to hardware-compatible joint compression and translate into practical inference efficiency. Because **ASQ** enforces sparsity and quantization through projection onto a joint feasible set, it can accommodate hardware-compatible sparsity patterns such as $N:M$. We consider 2:4 semi-structured sparsity, which retains two weights in every group of four, together with INT4 quantization. Under this setting, Figure 4a shows that **ASQ** achieves the highest average zero-shot accuracy among the compared methods (61.90%), indicating that its advantage is preserved when the joint compression constraints are restricted to an efficiently executable format.

We further benchmark end-to-end inference using vLLM (Kwon et al., 2023) with the Sparse-Marlin kernel (Frantar et al., 2025) on a single NVIDIA RTX 3090. Compared with BF16 dense inference, INT4+2:4 reduces the memory of the compression-target weights from 14.11 to 2.95 GB (4.78×), excluding the embedding and `lm_head`, and achieves 1.44–1.66× prefill speedup. For decoding,



(a) Performance (b) Memory (c) Prefill speed-up (d) Decoding speed-up

Figure 4: Evaluation of **ASQ** under the hardware-compatible INT4 + 2:4 setting on Qwen3-8B-Base. (a) Average zero-shot accuracy across seven benchmarks. (b) Memory footprint of the compression-target weights. (c) Prefill speed-up across prompt lengths. (d) Decoding speed-up across batch sizes. Speed-ups are measured relative to BF16 dense.

it provides a 3.20× speedup at batch size 1 and 1.53× at batch size 128 (Figure 4). Overall, these results show that the compression gains of **ASQ** can be realized in a hardware-supported form with tangible reductions in memory footprint and end-to-end inference latency.

5 Discussion

Toward general principles for joint compression. **ASQ** explicitly captures the interplay between sparsification and quantization through a unified joint formulation, unlocking substantial Pareto improvements under aggressive compression. Recent theoretical studies have characterized order effects and compounded errors in sequential compression pipelines (Harma et al., 2025; Kim et al., 2026), whereas the optimization geometry of simultaneous sparse–quantized constraints remains less understood. To begin characterizing this setting, our convergence analysis and optimization-dynamics study provide stationary guarantees for the core exact-projection formulation and empirical evidence of progressive alignment and stabilization. These results point toward a deeper theoretical understanding of joint compression, including non-asymptotic composite error bounds and a formal characterization of how discrete grids and sparsity constraints jointly reshape the loss landscape. Such analyses could yield more general design principles for joint model compression.

Bridging compression and efficient execution. Translating aggressive theoretical compression into practical inference speedups remains an important systems challenge. At an extreme theoretical compression rate of $40\times$, corresponding to 4-bit quantization with 90% sparsity, **ASQ** exhibits notable robustness compared to prior methods. In practice, current commodity hardware provides structured acceleration primarily for restricted formats, such as the INT4+2:4 setting evaluated in our end-to-end benchmarks. However, rapid progress in specialized sparse engines and low-bit execution libraries (Xia et al., 2024; Agarwalla et al., 2024; Fan et al., 2025; Frantar et al., 2025; Macko & Boža, 2026) highlights the growing viability of fine-grained compression. By demonstrating that deep networks retain high fidelity under aggressive joint compression, **ASQ** exposes a significantly broader compression frontier. Bridging this algorithmic frontier with dedicated systems—spanning co-designed custom GEMM kernels, mixed-precision sparse representations, and specialized hardware—represents a vital direction for future systems research.

6 Conclusion

In this work, we have presented **ASQ**, a principled ADMM-based framework that formulates the joint sparsification and quantization of LLMs as a unified constrained optimization problem. By deriving an exact, closed-form joint projection, **ASQ** simultaneously resolves parameter sparsity and discrete quantization levels based on objective importance and grid compatibility, underpinned by theoretical convergence guarantees. Extensive empirical evaluations demonstrate that **ASQ** substantially outperforms existing joint compression methods under aggressive compression regimes, establishing superior Pareto trade-offs across aggressive and hardware-oriented structured settings. Overall, our findings underscore that directly co-optimizing sparsification and quantization is essential for pushing the practical efficiency frontier of LLMs.

AI USE STATEMENT

In this work, we used generative AI tools to assist with research ideation and execution, including discussion of experimental analyses, ablation studies, and the presentation and interpretation of results, as well as to assist in checking and developing mathematical arguments. We have not used generative AI tools for generating synthetic datasets, and the other required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for literature search and polishing the writing of the manuscript, including improving clarity, grammar, and conciseness. All AI-assisted content was reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects or the collection of sensitive personal data. All experiments are conducted using publicly available models and datasets.

REPRODUCIBILITY STATEMENT

We provide detailed information to facilitate reproduction of our results. The formulation and implementation of ASQ are described in Section 3, with complete proofs of the joint projection and convergence results provided in Appendices A and B. Appendix C reports the experimental setup, including datasets, optimization settings, hyperparameters, evaluation protocols, and inference benchmarking details. Additional numerical results across models and compression settings are provided in Appendix D. The implementation code will be released.

REFERENCES

Abhinav Agarwalla, Abhay Gupta, Alexandre Marques, Shubhra Pandit, Michael Goin, Eldar Kurtic, Kevin Leong, Tuan Nguyen, Mahmoud Salem, Dan Alistarh, et al. Enabling high-sparsity foundational llama models with efficient pretraining and deployment. *arXiv preprint arXiv:2405.03594*, 2024.

Guangji Bai, Yijiang Li, Chen Ling, Kibaek Kim, and Liang Zhao. Sparsellm: Towards global pruning of pre-trained language models. *NeurIPS*, 2024.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. *AAAI*, 2020.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 2011.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *NAACL*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Ruibo Fan, Xiangrui Yu, Peijie Dong, Zeyu Li, Gu Gong, Qiang Wang, Wei Wang, and Xiaowen Chu. Spinfer: Leveraging low-level sparsity for efficient large language model inference on gpus. *EuroSys*, 2025.

Elias Frantar and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. *NeurIPS*, 2022.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. *ICML*, 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *ICLR*, 2023.

Elias Frantar, Roberto L Castro, Jiale Chen, Torsten Hoefler, and Dan Alistarh. Marlin: Mixed-precision auto-regressive parallel inference on large language models. *PPoPP*, 2025.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-power computer vision*. 2022.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Hang Guo, Yawei Li, and Luca Benini. Optimal brain restoration for joint quantization and sparsification of llms. *ICLR*, 2026.

Jinyang Guo, Jianyu Wu, Zining Wang, Jiaheng Liu, Ge Yang, Yifu Ding, Ruihao Gong, Haotong Qin, and Xianglong Liu. Compressing large language models by joint sparsification and quantization. *ICML*, 2024.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *NeurIPS*, 2015.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *ICLR*, 2016.

Simla Burcu Harma, Ayan Chakraborty, Elizaveta Kostenok, Danila Mishin, Dongho Ha, Babak Falsafi, Martin Jaggi, Ming Liu, Yunho Oh, Suvinay Subramanian, et al. Effective interplay between sparsity and quantization: From theory to practice. *ICLR*, 2025.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *JMLR*, 2021.

Minjun Kim, Jaehyeon Choi, Hyunwoo Yang, Jongjin Kim, Jinho Song, and U Kang. Prune-then-quantize or quantize-then-prune? understanding the impact of compression order in joint model compression. *ICLR*, 2026.

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. *SOSP*, 2023.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *NeurIPS*, 1989.

Kwanhee Lee, Hyeondo Jang, Dongyeop Lee, Dan Alistarh, and Namhoon Lee. The unseen frontier: Pushing the limits of llm sparsity with surrogate-free admm. *ICLR*, 2026.

YuXin Li, Felix Dangel, Derek Tam, and Colin Raffel. Fishers for free? approximating the fisher information matrix by recycling the squared gradient accumulator. *ICML*, 2025.

Yesheng Liang, Haisheng Chen, Zihan Zhang, Song Han, and Zhijian Liu. ParoQuant: Pairwise Rotation Quantization for Efficient Reasoning LLM Inference. *ICLR*, 2026.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *MLSys*, 2024.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. *ICLR*, 2025.

Vladimír Macko and Vladimír Boža. Macko: Sparse matrix-vector multiplication for low sparsity. *ICML*, 2026.

James Martens. New insights and perspectives on the natural gradient method. *JMLR*, 2020.

Xiang Meng, Kayhan Behdin, Haoyue Wang, and Rahul Mazumder. Alps: Improved optimization for highly sparse one-shot pruning for large language models. *NeurIPS*, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *ICLR*, 2017.

Mohammad Mozaffari, Amir Yazdanbakhsh, and Maryam Mehri Dehnavi. Slim: One-shot quantization and sparsity with low-rank approximation for llm weight compression. *ICML*, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 2021.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *ICLR*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *ICLR*, 2019.

Haojun Xia, Zhen Zheng, Yuchao Li, Donglin Zhuang, Zhongzhu Zhou, Xiafei Qiu, Yong Li, Wei Lin, and Shuaiwen Leon Song. Flash-llm: Enabling cost-effective and highly-efficient large generative model inference with unstructured sparsity. *VLDB*, 2024.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. *ICML*, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *ACL*, 2019.

Jiaqi Zhao, Miao Zhang, Ming Wang, Yuzhang Shang, Kaihao Zhang, Weili Guan, Yaowei Wang, and Min Zhang. Ptq1.61: Push the real limit of extremely low-bit post-training quantization methods for large language models. *ACL*, 2025.

Appendix

CONTENTS

A Proof of Proposition 3.1 **14**

B Convergence Analysis **14**

B.1 ADMM with an Inexact x-Update 14

B.2 Assumptions 14

B.3 Stationarity 14

B.4 Convergence Result 15

C Experimental Details **17**

C.1 Details for Section 4.1 17

C.2 Details for Section 4.2 18

C.3 Details for Section 4.3 19

C.4 Details for Section 4.4 19

D Additional Results **20**

D.1 Numerical Results for Section 4.1 20

D.2 Numerical Results for Section 4.4 25

A Proof of Proposition 3.1

Proof. Fix a support $I \subseteq \{1, \dots, n\}$ with $|I| \le k$. For a fixed support, the optimal nonzero quantized value at coordinate $i \in I$ is $\hat{q}_i$ by (9). The corresponding projection cost is

$$\begin{aligned} \|z(I) - v\|_2^2 &= \sum_{i \notin I} v_i^2 + \sum_{i \in I} (v_i - \hat{q}_i)^2 \\ &= \sum_{i=1}^{n} v_i^2 - \sum_{i \in I} \left(v_i^2 - (v_i - \hat{q}_i)^2 \right) \\ &= \sum_{i=1}^{n} v_i^2 - \sum_{i \in I} G_i. \end{aligned} \tag{21}$$

The first term is independent of I, so minimizing the projection cost is equivalent to maximizing $\sum_{i \in I} G_i$ subject to $|I| \le k$. Since this sum is additive over indices, it is maximized by selecting the indices of the (at most) k largest positive gains, which proves (13). □

B Convergence Analysis

We consider the joint sparse–quantized optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad x \in \mathcal{C}, \tag{22}$$

where

$$\mathcal{C} := \mathcal{S}_k \cap \mathcal{Q}, \qquad \mathcal{S}_k := \left\{ x \in \mathbb{R}^n : \|x\|_0 \le k \right\}, \qquad \mathcal{Q} := G^n. \tag{23}$$

Here $G \subset \mathbb{R}$ is a fixed finite quantization grid containing zero. Consequently, $\mathcal{C}$ is a finite set and f attains its minimum on $\mathcal{C}$; we write $f_{\min} := \min_{z \in \mathcal{C}} f(z)$. The (possibly set-valued) Euclidean projection onto $\mathcal{C}$ is denoted by $\Pi_{\mathcal{C}}(v) := \arg\min_{z \in \mathcal{C}} \|z - v\|_2^2$.

B.1 ADMM with an Inexact x-Update

For a penalty parameter $\lambda > 0$ and the scaled dual variable u, define the x-subproblem objective

$$\phi_t(x) := f(x) + \frac{\lambda}{2} \left\| x - z^t + u^t \right\|_2^2. \tag{24}$$

Starting from $z^0 \in \mathcal{C}$ and $u^0 \in \mathbb{R}^n$, the iterates are generated according to

$$x^{t+1} \approx \arg\min_{x \in \mathbb{R}^n} \phi_t(x) \quad \text{(in the sense of Assumption B.2)}, \tag{25}$$

$$z^{t+1} \in \Pi_{\mathcal{C}}\left(x^{t+1} + u^t \right), \tag{26}$$

$$u^{t+1} = u^t + x^{t+1} - z^{t+1}. \tag{27}$$

B.2 Assumptions

Assumption B.1 (Smoothness)**.** *The function f is continuously differentiable and has a β-Lipschitz continuous gradient: $\|\nabla f(x) - \nabla f(x')\|_2 \le \beta \|x - x'\|_2$ for all $x, x' \in \mathbb{R}^n$.*

Assumption B.2 (Inexact x-update)**.** *There exists a constant $\varepsilon \ge 0$ such that, for every $t \ge 0$,*

$$\left\| \nabla \phi_t(x^{t+1}) \right\|_2 \le \varepsilon \left\| x^{t+1} - z^t \right\|_2. \tag{28}$$

The exact x-update corresponds to $\varepsilon = 0$.

B.3 Stationarity

Definition B.1 (λ-stationarity)**.** A point $\bar{x} \in \mathcal{C}$ is called λ-stationary for Problem (22) if $\bar{x} \in \Pi_{\mathcal{C}}\left(\bar{x} - \lambda^{-1} \nabla f(\bar{x}) \right)$.

B.4 Convergence Result

Theorem B.1 (Convergence with inexact x-updates)**.** *Suppose that Assumptions B.1 and B.2 hold. Let $\beta_\varepsilon := \beta + \varepsilon$ and assume that*

$$\lambda > 5\,\beta_\varepsilon = 5\,(\beta + \varepsilon). \tag{29}$$

Then the iterates generated by (25)–(27) satisfy

$$\sum_{t=1}^{\infty} \left\|x^{t+1} - x^t\right\|_2^2 < \infty, \tag{30}$$

$$\left\|x^{t+1} - x^t\right\|_2 \longrightarrow 0, \tag{31}$$

$$\left\|x^{t+1} - z^{t+1}\right\|_2 \longrightarrow 0. \tag{32}$$

Moreover, there exist $\bar{z} \in \mathcal{C}$ and $T_0 \geq 1$ such that $z^t = \bar{z}$ for all $t \geq T_0$, the sequences converge,

$$x^t \longrightarrow \bar{z}, \qquad u^t \longrightarrow -\lambda^{-1}\nabla f(\bar{z}), \tag{33}$$

and $\bar{z}$ is a λ-stationary point of Problem (22). Finally, the number of changes of the discrete iterate is bounded explicitly:

$$\#\left\{t \geq 1 : z^{t+1} \neq z^t\right\} \;\leq\; \frac{4\left(\Phi^1 - f_{\min}\right)}{\sigma\,\delta^2}, \tag{34}$$

where Φ^1, σ and δ are defined in (45), (44) and (49), respectively.

Proof of Theorem B.1. Throughout, we write

$$g^t := \nabla f(x^t), \quad e^{t+1} := \nabla \phi_t(x^{t+1}), \quad \tilde{g}^t := g^t - e^t, \quad d^{t+1} := x^{t+1} - z^t, \quad \Delta z^t := z^t - z^{t-1}. \tag{35}$$

By Assumption B.2, $\|e^{t+1}\|_2 \leq \varepsilon\,\|d^{t+1}\|_2$ for all $t \geq 0$.

Step 1: Dual-gradient identity. Condition (29) gives $\lambda > \beta$, so ϕ_t is $(\lambda - \beta)$-strongly convex and has a unique minimizer, which satisfies (28); hence an admissible x^{t+1} always exists. By the definition of e^{t+1},

$$g^{t+1} + \lambda\big(x^{t+1} - z^t + u^t\big) = e^{t+1}, \qquad \text{that is,} \qquad x^{t+1} + u^t = z^t - \lambda^{-1}\tilde{g}^{t+1}. \tag{36}$$

Substituting (36) into (26) and (27) yields

$$z^{t+1} \in \Pi_{\mathcal{C}}\big(z^t - \lambda^{-1}\tilde{g}^{t+1}\big), \tag{37}$$

$$u^{t+1} = z^t - z^{t+1} - \lambda^{-1}\tilde{g}^{t+1}. \tag{38}$$

For every $t \geq 1$, combining (36) with (38) at index $t-1$ gives

$$d^{t+1} = \Delta z^t - \lambda^{-1}\big(g^{t+1} - g^t\big) + \lambda^{-1}\big(e^{t+1} - e^t\big). \tag{39}$$

Step 2: Sufficient decrease. Since z^{t+1} is an exact solution of (37) and $z^t \in \mathcal{C}$, $\|\Delta z^{t+1} + \lambda^{-1}\tilde{g}^{t+1}\|_2^2 \leq \|\lambda^{-1}\tilde{g}^{t+1}\|_2^2$, which is equivalent to

$$\langle \tilde{g}^{t+1}, \Delta z^{t+1}\rangle \leq -\frac{\lambda}{2}\,\|\Delta z^{t+1}\|_2^2. \tag{40}$$

By the descent lemma, (40), Assumption B.1 and $\|e^{t+1}\|_2 \leq \varepsilon\|d^{t+1}\|_2$,

$$\begin{aligned}
f(z^{t+1}) &\leq f(z^t) + \langle \nabla f(z^t), \Delta z^{t+1}\rangle + \frac{\beta}{2}\|\Delta z^{t+1}\|_2^2 \\
&= f(z^t) + \langle \tilde{g}^{t+1}, \Delta z^{t+1}\rangle + \langle e^{t+1}, \Delta z^{t+1}\rangle \\
&\qquad + \langle \nabla f(z^t) - \nabla f(x^{t+1}), \Delta z^{t+1}\rangle + \frac{\beta}{2}\|\Delta z^{t+1}\|_2^2 \\
&\leq f(z^t) - \frac{\lambda - \beta}{2}\|\Delta z^{t+1}\|_2^2 + (\beta + \varepsilon)\,\|d^{t+1}\|_2\,\|\Delta z^{t+1}\|_2 \\
&\leq f(z^t) - \frac{\lambda - \beta_\varepsilon}{2}\|\Delta z^{t+1}\|_2^2 + \beta_\varepsilon\,\|d^{t+1}\|_2\,\|\Delta z^{t+1}\|_2.
\end{aligned} \tag{41}$$

Next, for $t \geq 1$ we have $x^{t+1} - x^t = \Delta z^t + d^{t+1} - d^t$. Taking norms in (39) and using $\|e^{t+1}\|_2 \leq \varepsilon\|d^{t+1}\|_2$ and $\|e^t\|_2 \leq \varepsilon\|d^t\|_2$,

$$\begin{aligned}\|d^{t+1}\|_2 &\leq \|\Delta z^t\|_2 + \frac{\beta}{\lambda}\big(\|\Delta z^t\|_2 + \|d^{t+1}\|_2 + \|d^t\|_2\big) + \frac{\varepsilon}{\lambda}\big(\|d^{t+1}\|_2 + \|d^t\|_2\big) \\ &\leq \|\Delta z^t\|_2 + \frac{\beta_\varepsilon}{\lambda}\big(\|\Delta z^t\|_2 + \|d^{t+1}\|_2 + \|d^t\|_2\big),\end{aligned}$$

and rearranging,

$$\|d^{t+1}\|_2 \leq \kappa\,\|d^t\|_2 + c\,\|\Delta z^t\|_2, \qquad \kappa := \frac{\beta_\varepsilon}{\lambda - \beta_\varepsilon}, \qquad c := \frac{\lambda + \beta_\varepsilon}{\lambda - \beta_\varepsilon}. \tag{42}$$

Condition (29) implies $\kappa \in [0, 1)$, so by the convexity of $s \mapsto s^2$,

$$\|d^{t+1}\|_2^2 \leq \kappa\,\|d^t\|_2^2 + \frac{c^2}{1-\kappa}\,\|\Delta z^t\|_2^2. \tag{43}$$

Define

$$m := \frac{c}{1-\kappa} = \frac{\lambda + \beta_\varepsilon}{\lambda - 2\beta_\varepsilon}, \qquad \sigma := \lambda - \beta_\varepsilon - 2\beta_\varepsilon m = \frac{\lambda\,(\lambda - 5\beta_\varepsilon)}{\lambda - 2\beta_\varepsilon}, \qquad A := \frac{1}{1-\kappa}\Big(\frac{\beta_\varepsilon}{2m} + \frac{\sigma}{4m^2}\Big), \tag{44}$$

and the potential

$$\Phi^t := f(z^t) + A\,\|d^{t+1}\|_2^2. \tag{45}$$

Condition (29) implies $\sigma > 0$ and $A > 0$. By Young's inequality,

$$\beta_\varepsilon\,\|d^{t+1}\|_2\,\|\Delta z^{t+1}\|_2 \leq \frac{\beta_\varepsilon}{2m}\,\|d^{t+1}\|_2^2 + \frac{\beta_\varepsilon m}{2}\,\|\Delta z^{t+1}\|_2^2,$$

and by the definitions in (44),

$$A(1-\kappa) = \frac{\beta_\varepsilon}{2m} + \frac{\sigma}{4m^2}, \qquad \frac{Ac^2}{1-\kappa} = \frac{\beta_\varepsilon m}{2} + \frac{\sigma}{4}.$$

Combining (41) with (43) at index $t+1$, we obtain, for every $t \geq 1$,

$$\begin{aligned}\Phi^{t+1} - \Phi^t &\leq -\frac{\lambda - \beta_\varepsilon - \beta_\varepsilon m}{2}\,\|\Delta z^{t+1}\|_2^2 + \frac{\beta_\varepsilon}{2m}\,\|d^{t+1}\|_2^2 \\ &\quad - A(1-\kappa)\,\|d^{t+1}\|_2^2 + \frac{Ac^2}{1-\kappa}\,\|\Delta z^{t+1}\|_2^2 \\ &= -\frac{\sigma}{4}\,\|\Delta z^{t+1}\|_2^2 - \frac{\sigma}{4m^2}\,\|d^{t+1}\|_2^2.\end{aligned} \tag{46}$$

Step 3: Lower boundedness and vanishing residuals. Since $A > 0$ and $z^t \in \mathcal{C}$, we have $\Phi^t \geq f(z^t) \geq f_{\min}$. Summing (46) from $t = 1$ to T gives

$$\frac{\sigma}{4}\sum_{t=1}^{T}\|\Delta z^{t+1}\|_2^2 + \frac{\sigma}{4m^2}\sum_{t=1}^{T}\|d^{t+1}\|_2^2 \leq \Phi^1 - \Phi^{T+1} \leq \Phi^1 - f_{\min}. \tag{47}$$

Taking $T \to \infty$, both sequences are square-summable and

$$\|\Delta z^{t+1}\|_2 \longrightarrow 0, \qquad \|d^{t+1}\|_2 \longrightarrow 0, \qquad \|e^{t+1}\|_2 \leq \varepsilon\,\|d^{t+1}\|_2 \longrightarrow 0. \tag{48}$$

Since $x^{t+1} - x^t = \Delta z^t + d^{t+1} - d^t$, this proves (30) and consequently (31). In addition, $\|x^{t+1} - z^{t+1}\|_2 = \|d^{t+1} - \Delta z^{t+1}\|_2 \leq \|d^{t+1}\|_2 + \|\Delta z^{t+1}\|_2 \longrightarrow 0$, which proves (32).

Step 4: Finite settling and stationarity. Since $\mathcal{C}$ is a finite set, the quantity

$$\delta := \min\big\{\|z - z'\|_2 : z, z' \in \mathcal{C},\ z \neq z'\big\} \tag{49}$$

is strictly positive. Every index $t \geq 1$ with $z^{t+1} \neq z^t$ contributes at least δ^2 to the first sum in (47), which gives (34). In particular, there exists $T_0 \geq 1$ such that $z^t = \bar{z}$ for all $t \geq T_0$, for some $\bar{z} \in \mathcal{C}$.

For $t \geq T_0$, $x^{t+1} = \bar{z} + d^{t+1} \longrightarrow \bar{z}$ by (48). By the continuity of ∇f and (48), $\tilde{g}^{t+1} = g^{t+1} - e^{t+1} \longrightarrow \nabla f(\bar{z})$, and hence, by (38), $u^{t+1} = -\lambda^{-1}\tilde{g}^{t+1} \longrightarrow -\lambda^{-1}\nabla f(\bar{z})$. This proves (33).

It remains to show that $\bar{z}$ is λ-stationary. For $t \geq T_0$, (37) gives $\bar{z} \in \Pi_{\mathcal{C}}\big(\bar{z} - \lambda^{-1}\tilde{g}^{t+1}\big)$, so for every $z \in \mathcal{C}$,

$$\big\|\bar{z} - \big(\bar{z} - \lambda^{-1}\tilde{g}^{t+1}\big)\big\|_2^2 \leq \big\|z - \big(\bar{z} - \lambda^{-1}\tilde{g}^{t+1}\big)\big\|_2^2.$$

Letting $t \to \infty$ yields $\big\|\bar{z} - \big(\bar{z} - \lambda^{-1}\nabla f(\bar{z})\big)\big\|_2^2 \leq \big\|z - \big(\bar{z} - \lambda^{-1}\nabla f(\bar{z})\big)\big\|_2^2$ for every $z \in \mathcal{C}$, that is, $\bar{z} \in \Pi_{\mathcal{C}}\big(\bar{z} - \lambda^{-1}\nabla f(\bar{z})\big)$. Thus $\bar{z}$ is a λ-stationary point of Problem (22). $\square$

C Experimental Details

C.1 Details for Section 4.1

Calibration data. Baselines (SparseGPTQ, JSQ, SLiM, OBR) follow the convention of Frantar & Alistarh (2023), sampling 128 calibration sequences of length 2048 from C4.

Optimization data. **ASQ** is optimized for one epoch over C4 sequences of length 2048, using an effective batch size of 16. The optimization set is sampled once and held fixed across sparsity levels for each model.

Details. A full list of hyperparameters is in Tables 2 and 3.

Table 2: Global hyperparameters of **ASQ** shared across all models.

Hyperparameter	Value
LR schedule	Linear decay
Adam (β_1, β_2)	(0.9, 0.999)
Batch size	16
Quantization	symmetric
Group size	128

Table 3: Learning rate (η) and penalty (λ) under 4-bit quantization, across models at different sparsity levels.

	Sparsity	LLaMA-2-7B	LLaMA-3.1-8B	Qwen3-8B-Base	Qwen3-14B-Base
η	50%	2e-5	1e-5	2e-5	2e-5
	60%	5e-5	2e-5	5e-5	2e-5
	70%	5e-5	5e-5	5e-5	5e-5
	80%	5e-5	5e-5	1e-4	5e-5
	90%	1e-4	5e-5	1e-4	1e-4
λ	50%	1e-2	1e-2	1e-2	1e-2
	60%	1e-2	1e-3	1e-2	1e-3
	70%	1e-2	1e-2	1e-3	1e-3
	80%	1e-3	1e-3	1e-3	1e-3
	90%	1e-4	2e-4	2e-4	1e-4

Evaluation. Perplexity is measured on the held-out WikiText-2 (Merity et al., 2017) and C4 (Raffel et al., 2020) sets. Zero-shot performance is evaluated with `lm-eval-harness` (Gao et al., 2024) across seven tasks: ARC-Easy/Challenge (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), RTE (Wang et al., 2019) and WinoGrande (Sakaguchi et al., 2021), and we report the average accuracy.

C.2 Details for Section 4.2

We compute all optimization-dynamics metrics over the collection of compressed weight tensors. Let x^t and z^t denote the continuous primal variable and the sparse–quantized split variable at projection update t, respectively, and let $m_i^t := \mathbb{I}[z_i^t \neq 0]$ denote the corresponding binary support mask.

Validation loss. The validation-loss curves in Figure 3a report the language-modeling loss on the same held-out data, evaluated once at x^t and once at z^t. They measure the predictive objective attained by the continuous model and by its feasible sparse–quantized counterpart, respectively. The gap between the two curves therefore reflects the loss increase induced by enforcing the joint compression constraint; a narrowing gap indicates that the continuous and compressed iterates are becoming functionally aligned.

Primal residual. The support and quantization components shown in Figure 3b are

$$R_{\text{supp}}^t := \frac{\left\|(1-m^t)\odot(x^t-z^t)\right\|_2^2}{\|x^t\|_2^2}, \qquad R_{\text{quant}}^t := \frac{\left\|m^t\odot(x^t-z^t)\right\|_2^2}{\|x^t\|_2^2}.$$

Because z^t is zero outside its support, the first term measures the energy of the coordinates removed by sparsification, whereas the second measures the quantization discrepancy on retained coordinates. The two terms satisfy

$$R_{\text{supp}}^t + R_{\text{quant}}^t = \frac{\|x^t - z^t\|_2^2}{\|x^t\|_2^2},$$

and therefore decompose the squared relative primal residual exactly. Their sum measures the relative violation of the ADMM coupling constraint $x = z$, while the decomposition identifies whether the violation arises from support selection or quantized values. Decreasing values indicate that x^t is approaching the joint sparse–quantized constraint set represented by z^t.

Dual residual. The relative dual residual in Figure 3c is

$$R_{\text{dual}}^t := \frac{\|z^t - z^{t-1}\|_2}{\|z^{t-1}\|_2},$$

which measures the relative change in the split iterate between consecutive projection updates, including changes in both its support and quantized values. Thus, this metric tracks stabilization of the complete compressed representation rather than the magnitude of the dual variable itself. A value approaching zero indicates that successive projection updates produce increasingly similar sparse–quantized iterates.

Support change. For n compressed weights, the support-change metric in Figure 3d is the normalized Hamming distance

$$H^t := \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\left[m_i^t \neq m_i^{t-1}\right].$$

Thus, H^t is the fraction of coordinates whose support membership changes at update t. It isolates discrete support turnover from changes in the quantized values on retained coordinates. A value approaching zero indicates that the selected sparse support is stabilizing. In implementation, denominators are lower-bounded by a small constant for numerical stability.

C.3 Details for Section 4.3

We evaluate all 2^3 combinations of joint projection (JP), quantization-scale optimization (QSO), and sensitivity weighting (SW). Table 4 summarizes the full and ablated variants of each component. When multiple components are ablated, the corresponding alternatives in Table 4 are combined.

Table 4: Details of the ablation variants used in Section 4.3.

Component	Enabled	Ablated
JP	Joint sparse–quantized projection	Support selection followed by quantization
QSO	Scale optimized over the selected support	Fixed absmax scale
SW	Diagonal empirical-Fisher metric	Standard Euclidean metric

C.4 Details for Section 4.4

We benchmark Qwen3-8B-Base on a single NVIDIA RTX 3090 using vLLM v0.15.1. The **ASQ** checkpoint uses symmetric INT4 quantization with group size 128 and 2:4 sparsity. BF16 dense linear layers use cuBLAS, INT4 dense uses Marlin, and **ASQ** INT4+2:4 uses the semi-structured Sparse-Marlin kernel (Frantar et al., 2025). All variants use the same engine limits (128 sequences, 16,384 batched tokens, and 4096-token context), greedy decoding, disabled prefix caching, and a matched 4 GiB KV cache. We measure prefill latency as time to first token and decode latency as the mean latency over the final 63 steps of a 64-token generation. Each entry reports the median of five trials following two warm-up trials. Configurations requiring scheduler-side chunking are excluded. Kernel dispatch is verified for all linear modules; the embedding layers and `lm_head` remain in 16-bit precision.

D ADDITIONAL RESULTS

Here we provide additional experimental results that complement Section 4. Section D.1 provides the full numerical results for Section 4.1, and Section D.2 provides the numerical results for Section 4.4.

D.1 NUMERICAL RESULTS FOR SECTION 4.1

Table 5 provides the full WikiText-2 perplexity results under different compression rates. Tables 6, 7, 8, and 9 provide the corresponding full zero-shot evaluation results. **CR** denotes the compression rate relative to dense 16-bit weights. **Bold** and underlined values indicate the best and second-best results, respectively.

Table 5: WikiText-2 perplexity (↓) under combined sparsification and quantization, across model families.

Model	Method	Compression rate				
		8×	10×	13.3×	20×	40×
LLaMA-2-7B (Dense: 5.47)	SparseGPTQ	7.34	11.34	31.25	116	NaN
	JSQ	7.29	11.15	154	2937	2.9e4
	SLiM-LoRA	7.32	11.82	19.19	128	5165
	OBR	**7.00**	12.02	47.79	1283	3.0e4
	ASQ	7.32	**9.01**	**12.69**	**20.54**	**30.84**
LLaMA-3.1-8B (Dense: 6.23)	SparseGPTQ	11.73	21.25	57.03	304	1372
	JSQ	11.10	25.73	225	2807	2.8e4
	SLiM-LoRA	11.02	25.38	140	2174	9158
	OBR	**9.18**	16.02	46.63	702	9.1e5
	ASQ	10.61	**14.49**	**21.66**	**34.22**	**53.57**
Qwen3-8B-Base (Dense: 7.00)	SparseGPTQ	8.77	**11.13**	20.13	74.52	388
	JSQ	10.47	15.89	71.96	297	1.7e5
	SLiM-LoRA	10.17	15.26	55.62	336	4.3e5
	OBR	**8.72**	12.30	33.61	349	4.3e4
	ASQ	9.91	11.94	**15.10**	**22.57**	**36.18**
Qwen3-14B-Base (Dense: 6.37)	SparseGPTQ	**7.84**	9.29	16.22	45.35	315.94
	JSQ	10.98	13.39	151	526	2.2e7
	SLiM-LoRA	10.32	12.67	39.01	414	6.7e4
	OBR	8.07	**9.28**	19.60	165	4.1e4
	ASQ	9.18	10.77	**13.15**	**18.81**	**33.70**

Table 6: Zero-shot accuracy (%, ↑) of LLaMA-2-7B across seven tasks under combined sparsification and quantization.

CR	Method	ARC-c	ARC-e	BoolQ	HellaSwag	PIQA	RTE	WinoG	Avg
1×	-	43.43	76.35	77.71	57.16	78.07	62.82	69.14	66.38
8×	SparseGPTQ	37.54	71.04	**75.11**	52.04	75.68	54.87	**68.51**	**62.11**
	JSQ	38.23	**71.93**	74.95	52.29	**76.17**	53.79	67.01	62.05
	SLiM-LoRA	**39.42**	70.71	70.37	53.06	75.30	**59.57**	66.14	62.08
	OBR	36.09	68.48	70.95	47.48	73.23	58.48	67.25	60.28
	ASQ	37.97	70.66	72.72	**53.55**	75.08	56.32	66.22	61.79
10×	SparseGPTQ	29.44	63.01	**70.12**	44.00	70.62	54.51	65.04	56.68
	JSQ	29.86	63.09	65.93	43.08	69.80	53.43	62.35	55.36
	SLiM-LoRA	**35.07**	**65.91**	69.17	49.04	**73.88**	54.15	**65.27**	**58.93**
	OBR	26.19	53.03	63.73	35.17	64.09	**62.82**	59.51	52.08
	ASQ	31.91	64.52	69.14	**49.12**	73.29	55.60	62.67	58.04
13.3×	SparseGPTQ	20.22	39.65	62.42	31.65	61.43	52.35	54.06	45.97
	JSQ	19.37	29.25	54.13	27.37	55.17	52.71	50.28	41.18
	SLiM-LoRA	**26.28**	51.56	**64.46**	37.69	64.96	**55.60**	54.54	**50.73**
	OBR	18.17	29.29	38.99	27.29	55.22	53.43	49.33	38.82
	ASQ	25.60	**54.38**	53.85	**41.34**	**69.31**	53.43	**56.27**	50.60
20×	SparseGPTQ	17.83	27.53	37.98	27.27	53.59	55.60	48.93	38.39
	JSQ	20.48	27.44	37.83	25.94	52.99	52.71	48.78	38.02
	SLiM-LoRA	19.71	27.10	61.04	27.34	55.22	51.62	48.78	41.54
	OBR	**21.16**	26.64	52.11	26.06	52.18	47.29	50.12	39.37
	ASQ	20.90	**45.75**	**61.38**	**35.82**	**65.56**	**55.96**	**50.91**	**48.04**
40×	SparseGPTQ	20.82	25.84	37.95	25.68	51.69	52.71	49.25	37.71
	JSQ	**22.10**	24.41	39.39	25.77	52.18	47.29	48.62	37.11
	SLiM-LoRA	20.65	27.06	37.86	26.06	53.16	47.29	**51.07**	37.59
	OBR	20.90	25.67	37.83	25.98	53.65	52.71	49.96	38.10
	ASQ	19.45	**39.06**	**54.25**	**30.45**	**61.97**	**53.07**	50.67	**44.13**

Table 7: Zero-shot accuracy (%, ↑) of LLaMA-3.1-8B across seven tasks under combined sparsification and quantization.

CR	Method	ARC-c	ARC-e	BoolQ	HellaSwag	PIQA	RTE	WinoG	Avg
1×	-	51.37	81.44	82.05	60.02	80.09	69.68	73.64	71.18
8×	SparseGPTQ	**40.96**	72.14	76.70	52.07	75.63	56.68	**68.82**	**63.29**
	JSQ	40.02	70.41	74.46	50.51	74.10	53.79	67.40	61.53
	SLiM-LoRA	40.87	**72.73**	**77.61**	50.44	75.57	56.32	68.03	63.08
	OBR	37.71	71.34	68.59	48.30	74.27	**58.48**	67.25	60.85
	ASQ	**40.96**	72.60	74.74	**53.58**	**75.79**	53.79	66.30	62.54
10×	SparseGPTQ	31.06	62.29	**72.63**	44.04	70.57	53.43	**65.98**	57.14
	JSQ	27.30	59.43	68.41	38.69	68.72	53.07	61.09	53.82
	SLiM-LoRA	33.19	**63.89**	66.54	42.66	71.38	53.43	62.98	56.30
	OBR	26.11	57.45	62.29	37.29	66.27	52.71	59.12	51.61
	ASQ	**33.28**	63.17	66.91	**47.70**	**72.74**	**58.48**	64.72	**58.14**
13.3×	SparseGPTQ	19.71	37.46	**63.09**	33.17	60.07	52.71	54.93	45.88
	JSQ	17.92	33.84	48.93	27.68	56.91	52.71	49.64	41.09
	SLiM-LoRA	22.61	48.78	62.45	31.48	62.73	**53.07**	53.83	47.85
	OBR	16.04	34.01	45.32	27.58	55.33	52.35	50.43	40.15
	ASQ	**27.82**	**55.05**	61.59	**40.92**	**69.75**	**53.07**	**56.12**	**52.04**
20×	SparseGPTQ	17.49	29.92	46.09	26.97	54.84	51.99	48.62	39.42
	JSQ	17.83	27.48	37.83	26.34	54.03	52.71	47.83	37.72
	SLiM-LoRA	19.11	30.43	**51.77**	26.49	54.62	**54.87**	48.86	40.88
	OBR	19.37	26.60	38.04	25.82	52.50	52.35	49.17	37.69
	ASQ	**23.12**	**44.57**	48.56	**34.69**	**66.05**	53.07	**51.46**	**45.93**
40×	SparseGPTQ	18.77	27.74	37.98	26.15	53.59	52.71	51.07	38.29
	JSQ	20.14	26.64	37.83	26.04	52.77	52.71	51.07	38.17
	SLiM-LoRA	**20.73**	24.83	37.83	26.07	52.77	52.35	**51.85**	38.06
	OBR	20.39	24.75	38.10	25.68	52.88	52.71	47.75	37.47
	ASQ	18.69	**38.38**	**53.39**	**28.89**	**59.79**	**53.07**	49.88	**43.16**

Table 8: Zero-shot accuracy (%, ↑) of Qwen3-8B-Base across seven tasks under combined sparsification and quantization.

CR	Method	ARC-c	ARC-e	BoolQ	HellaSwag	PIQA	RTE	WinoG	Avg
1×	-	53.07	81.86	83.00	58.90	79.33	75.45	72.69	72.04
8×	SparseGPTQ	48.04	77.02	**79.94**	**53.81**	**76.88**	70.04	**71.35**	**68.15**
	JSQ	45.22	76.47	74.31	50.73	76.12	**76.53**	68.03	66.77
	SLiM-LoRA	**51.02**	78.96	68.32	51.44	76.17	70.04	69.46	66.49
	OBR	46.84	78.24	79.45	49.20	74.59	75.09	69.06	67.50
	ASQ	49.83	**79.67**	76.76	52.74	76.39	71.12	69.30	67.97
10×	SparseGPTQ	**43.17**	73.74	76.24	48.37	75.08	57.40	**69.22**	**63.32**
	JSQ	38.31	68.10	74.37	42.43	71.71	58.84	62.12	59.41
	SLiM-LoRA	39.59	68.48	**78.35**	44.48	72.25	65.34	65.90	62.06
	OBR	36.09	68.31	67.58	40.03	68.61	**65.70**	61.09	58.20
	ASQ	42.41	**74.41**	70.24	**49.90**	**75.30**	62.09	64.80	62.74
13.3×	SparseGPTQ	29.10	60.77	62.75	38.42	68.34	52.71	**62.04**	53.45
	JSQ	20.22	46.84	62.11	29.79	60.99	52.71	51.93	46.37
	SLiM-LoRA	29.95	59.34	63.15	34.94	65.29	65.34	59.19	53.89
	OBR	19.88	38.05	56.97	29.69	57.40	52.71	52.41	43.87
	ASQ	**37.54**	**69.11**	**66.33**	**45.75**	**71.76**	**67.87**	58.48	**59.55**
20×	SparseGPTQ	17.24	34.34	**62.02**	28.53	56.47	52.71	50.67	43.14
	JSQ	17.32	29.38	52.75	27.33	54.24	53.07	49.57	40.52
	SLiM-LoRA	19.37	36.28	61.80	28.40	57.29	52.71	50.12	43.71
	OBR	19.54	27.15	43.55	26.17	53.16	52.35	49.25	38.74
	ASQ	**28.33**	**56.61**	56.64	**38.54**	**67.95**	**53.79**	**55.64**	**51.07**
40×	SparseGPTQ	19.28	27.36	42.08	26.45	53.97	52.71	48.46	38.62
	JSQ	**22.27**	25.21	37.83	25.77	53.48	52.71	51.62	38.41
	SLiM-LoRA	19.03	28.58	39.11	26.50	53.75	**54.87**	**51.78**	39.09
	OBR	20.48	24.83	37.83	25.56	53.37	52.71	48.93	37.67
	ASQ	18.69	**41.62**	**60.40**	**31.27**	**61.04**	52.71	49.80	**45.08**

Table 9: Zero-shot accuracy (%, ↑) of Qwen3-14B-Base across seven tasks under combined sparsification and quantization.

CR	Method	ARC-c	ARC-e	BoolQ	HellaSwag	PIQA	RTE	WinoG	Avg
1×	-	55.80	83.50	86.79	61.84	80.63	76.53	74.35	74.21
8×	SparseGPTQ	47.18	78.91	80.49	**57.44**	**79.33**	75.81	**74.74**	70.56
	JSQ	50.94	79.08	74.43	51.30	76.61	75.45	68.43	68.03
	SLiM-LoRA	51.45	80.13	**81.68**	56.37	78.94	76.17	69.93	**70.67**
	OBR	50.77	79.38	64.37	51.81	76.50	**76.90**	71.03	67.25
	ASQ	**52.39**	**80.72**	80.18	55.95	78.45	70.40	71.35	69.92
10×	SparseGPTQ	41.72	72.90	74.10	**53.19**	**77.80**	75.81	**71.67**	66.74
	JSQ	43.86	73.82	74.65	46.02	72.52	54.87	68.35	62.01
	SLiM-LoRA	46.50	76.68	**80.49**	52.02	75.73	**76.53**	69.93	**68.27**
	OBR	40.19	72.43	62.23	41.65	70.24	71.48	65.98	60.60
	ASQ	**47.18**	**77.27**	72.32	53.00	76.77	66.43	67.56	65.79
13.3×	SparseGPTQ	33.36	64.39	66.36	43.65	70.95	**71.84**	**67.17**	59.67
	JSQ	19.88	37.75	62.08	30.09	59.14	55.96	49.80	44.96
	SLiM-LoRA	37.54	68.56	71.99	43.20	70.18	59.21	66.06	59.53
	OBR	22.10	49.92	62.14	31.46	61.43	53.07	53.91	47.72
	ASQ	**42.24**	**73.61**	**72.23**	**49.56**	**73.61**	64.26	64.40	**62.85**
20×	SparseGPTQ	18.86	41.79	62.17	31.51	60.39	52.71	55.01	46.06
	JSQ	16.81	31.19	42.45	27.37	54.84	53.07	47.36	39.01
	SLiM-LoRA	24.32	51.09	62.17	30.46	61.59	52.71	54.78	48.16
	OBR	18.77	28.62	37.86	26.85	52.99	52.35	48.62	38.01
	ASQ	**30.03**	**60.94**	**65.17**	**41.42**	**69.97**	**62.45**	**58.01**	**55.43**
40×	SparseGPTQ	19.03	27.99	39.48	26.85	54.46	52.35	48.30	38.35
	JSQ	**22.01**	24.75	51.90	25.51	52.18	49.10	51.38	39.55
	SLiM-LoRA	18.26	30.60	**54.74**	26.68	53.97	**53.07**	50.91	41.17
	OBR	20.73	25.55	37.83	25.61	52.88	50.54	49.41	37.51
	ASQ	21.16	**42.38**	51.77	**33.34**	**64.09**	52.71	**52.17**	**45.38**

D.2 Numerical Results for Section 4.4

Table 10 provides the full perplexity results and zero-shot evaluation results under a hardware-compatible setting. Table 11 summarizes the measured operating points. The speed-up of INT4 dense diminishes at larger decode batch sizes and is negligible during prefill, whereas **ASQ** INT4+2:4 retains a clear advantage in both regimes.

Table 10: Performance of the hardware-compatible INT4+2:4 checkpoints of Qwen3-8B-Base. We report perplexity (↓) on WikiText-2 and C4 and zero-shot accuracy (%, ↑) on seven tasks. **Bold** and underlined values mark the best and second-best in each column.

	Perplexity (↓)		Zero-shot accuracy (↑)							
Method	Wiki2	C4	ARC-c	ARC-e	BoolQ	HellaSwag	PIQA	RTE	WinoG	Avg.
Dense (BF16)	7.00	10.41	53.07	81.86	83.00	58.90	79.33	75.45	72.69	72.04
SparseGPTQ	12.91	19.09	37.70	64.54	65.57	55.72	65.50	58.84	61.82	58.53
JSQ	17.16	25.07	29.41	54.34	64.02	49.31	64.82	58.21	61.39	54.50
SLiM-LoRA	12.94	20.35	43.34	**65.93**	68.41	54.84	**73.92**	61.29	64.82	61.79
OBR	13.27	21.46	37.02	65.84	65.63	54.66	69.88	**62.09**	62.46	59.65
ASQ	**12.81**	**18.79**	**43.42**	63.64	**68.65**	**55.87**	73.66	**62.09**	**65.94**	**61.90**

Table 11: Inference efficiency of the **ASQ** INT4+2:4 checkpoint. Decode entries report latency in ms followed by speedup over BF16 dense, while prefill entries report mean speedup over feasible batch sizes with the minimum–maximum range in parentheses. Decode uses $L_{\text{in}} = 128$ and 64 generated tokens.

Metric	Setting	BF16 Dense	INT4 Dense	**ASQ** (INT4+2:4)
GPU weight footprint	–	16.60 GB	6.30 GB	**5.44 GB**
Decode latency (speedup)	B=1	20.16 (1.00×)	7.63 (2.64×)	**6.29 (3.20×)**
	B=8	21.47 (1.00×)	8.40 (2.56×)	**7.21 (2.98×)**
	B=32	25.57 (1.00×)	13.80 (1.85×)	**10.74 (2.38×)**
	B=128	48.77 (1.00×)	42.41 (1.15×)	**31.91 (1.53×)**
Mean prefill speedup	L_{in}=128	1.00×	1.09× (0.98–1.20)	**1.66× (1.49–1.79)**
	L_{in}=512	1.00×	1.03× (0.97–1.08)	**1.54× (1.46–1.63)**
	L_{in}=2048	1.00×	0.99× (0.97–1.00)	**1.44× (1.42–1.46)**