Comp^{2}VLM: A Hybrid Framework Combining Quantization and Lossless Compression for Efficient Vision-Language Models

Dahun Choi[*], **Inseong Hwang**[*] , **Hyun Kim**[†]
Department of Electrical and Information Engineering and
Research Center for Electrical and Information Technology
Seoul National University of Science and Technology, Seoul, Korea
{dahun926, ish9707, hyunkim}@seoultech.ac.kr

Abstract

Vision–language models (VLMs) exhibit strong multimodal reasoning, yet their growing parameter counts incur prohibitive memory footprints and inference latency, hindering real-time deployment. Post-training quantization alleviates these costs, but aggressive low-bit quantization often degrades accuracy by disrupting visual–linguistic alignment. Meanwhile, lossless compression preserves accuracy but suffers from metadata overhead and sequential decoding, limiting GPU parallelism. We propose **Comp^{2}VLM**, a hybrid framework that combines entropy-aware quantization with low-overhead, parallel lossless coding for efficient VLM serving. We introduce **Duplication-Enhanced Quantization (DEQ)**, which adaptively selects group-wise scales to reduce the entropy of quantized representations while preserving reconstruction fidelity, thereby improving compressibility. To keep compression nearly free at runtime, we present **Quantized Data-aware Lossless Compression (QPress)**, a hardware-friendly scheme based on a **universal Huffman table** that enables block-wise metadata sharing and GPU-parallel decoding. Across diverse VLMs and benchmarks, Comp^{2}VLM maintains competitive accuracy while reducing the effective precision to **W**3.5**A**$3.2 \sim 4.9$. On LLaVA-OneVision-7B, it achieves **+4.7%** TextVQA and **+7.7%** OCRBench accuracy over W4A8-based SOTA at an average bit-width of **W3.5A4.6**. DEQ yields up to **33%** entropy reduction, and QPress improves compression efficiency by up to **13.3%** with only $\sim$**4%** end-to-end latency overhead. These results demonstrate the complementary benefits of entropy-aware quantization and low-overhead, parallel lossless coding for practical VLM inference.

1 Introduction

Recently, vision-language models (VLMs) have garnered significant attention for their strong performance in tasks requiring high-level contextual understanding, such as visual reasoning, which is far beyond simple object recognition [2, 4, 15, 18]. Similar to the trajectory of large language models (LLMs), these performance gains have been largely driven by scaling up the model parameters [32]. However, such scaling-based improvements come at the cost of substantially increased memory and computational requirements during inference, making real-time deployment on resource-constrained hardware challenging [10, 26]. For example, Qwen2-VL-72B [2] requires approximately 146GB of memory for inference, which is prohibitive even on high-end GPUs such as the H100 for single-device deployment.

[*]Equal contribution.
[†]Corresponding author: Hyun Kim, email: hyunkim@seoultech.ac.kr

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Model compression techniques to address these challenges can be broadly categorized into two approaches. First, lossy compression methods, such as quantization [5, 11] and pruning [12, 13], leverage the error tolerance of neural networks to achieve high compression and computational efficiency at the cost of some accuracy degradation [1, 19, 31]. However, because lossy compression inherently involves information loss, it has limitations in preserving model performance. This is particularly problematic for VLMs, which rely on fine-grained alignment between visual and linguistic representations; aggressively quantizing activations to low precision can disrupt multimodal feature alignment, leading to significant accuracy degradation [16, 27, 33]. Second, lossless compression preserves the original data entirely while improving the storage and transmission efficiency [6, 9]. However, the constraint of exact reconstruction limits the achievable compression ratio, and the variable-length nature of the compressed data poses challenges for efficient storage and transfer.

To overcome these limitations, combining quantization and lossless compression is a natural direction; however, naive integration at low bit-widths introduces two key challenges:

(C1) Accuracy–Compression Ratio Trade-off: Conventional quantization methods prioritize minimizing accuracy degradation, typically by allocating as many quantization bins as possible under a given bit-width. Although this strategy helps preserve model accuracy, it reduces redundancy in the data distribution, resulting in higher entropy and limiting the effectiveness of subsequent lossless compression. Therefore, effective integration of both approaches requires a quantization strategy that maintains accuracy while simultaneously achieving high redundancy.

(C2) Metadata & Decoding Overhead: Existing lossless compression methods often depend on complex dictionaries or heavy metadata to achieve high compression ratios. In low-bit representations, however, data are already compact, causing such overhead to dominate storage and reduce effective compression efficiency. Moreover, complex decoding logic limits hardware parallelism, leading to increased end-to-end latency.

To address these limitations, recent studies have explored double-compression approaches that combine quantization with lossless compression to improve overall compression ratios. WCM [28] enhances weight compressibility through compression-aware scaling and pruning, whereas QStore [23] exploits value redundancy in quantized integer codes and applies per-chunk Huffman coding to reduce memory footprint. However, these approaches have only been validated in INT8 settings, leaving their scalability to lower bit-widths unverified. In particular, the scaling and pruning strategies employed by WCM may further degrade the already limited representational capacity in low-bit regimes, potentially exacerbating **(C1)**. Moreover, while QStore focuses primarily on memory reduction, it overlooks the metadata overhead introduced by per-chunk codebooks and the sequential nature of Huffman decoding, leaving **(C2)** unaddressed.

To address **(C1)** the accuracy–compression trade-off and **(C2)** metadata/decoding overheads, we propose *Comp^{2}VLM*, a hybrid framework that combines entropy-aware quantization with low-overhead, parallel lossless coding. Our key contributions are:

- *Hybrid Quantization-Compression Framework:* We propose *Comp^{2}VLM*, a unified framework that combines entropy-aware quantization with low-overhead, parallel lossless coding to reduce the memory footprint and weight traffic of VLM inference. This approach improves compression efficiency over conventional single-stage compression methods while maintaining model accuracy.
- *Duplication-Enhanced Quantization (DEQ):* We introduce DEQ, which strategically adjusts the redundancy of quantized data to improve the effectiveness of subsequent lossless compression. By selecting group-wise scale factors that minimize the entropy of quantized data, DEQ achieves both accuracy preservation and improved compression efficiency (Solution for **C1**).
- *Quantized Data-aware Lossless Compression (QPress):* We propose QPress, a lossless compression algorithm optimized for low-bit integer formats. QPress reduces metadata overhead through a shared metadata structure based on a universal Huffman table (UHT) and supports block-wise parallel decoding so that decompression adds little latency on hardware accelerators (Solution for **C2**).

To the best of our knowledge, this is the first work to co-design quantization and lossless compression for efficient VLM serving. *Comp^{2}VLM* achieves superior compression ratios over existing 4-bit weight and 8-bit activation (W4A8) quantization methods across diverse VLM benchmarks while

preserving competitive accuracy. On LLaVA-OneVision (OV)-7B, it attains an average effective bit-width of W3.5A4.6, outperforming the W4A8-based MBQ [16] by 4.7% on TextVQA [24] and 7.7% on OCRBench [21].

2 Background

2.1 Limitations of Single Compression

Post-training quantization (PTQ) is widely used to reduce the memory footprint and computation cost of LLMs and VLMs. Recent PTQ methods improve low-precision robustness by suppressing high-magnitude activation outliers through channel-wise scaling [31], importance-aware weight preservation [19], or Hadamard-based feature rotation [1]. However, directly applying general-purpose PTQ to VLMs often leads to sub-optimal performance because vision and language tokens exhibit heterogeneous statistics. Modality-aware methods such as MBQ [16] and MQuant [33] address this issue by considering vision-language token sensitivity and modality-specific quantization, yet they remain largely limited to W4A8. In more aggressive low-bit regimes required for on-device deployment, quantization alone can cause severe information loss and catastrophic accuracy degradation.

Lossless compression offers a complementary direction by reducing storage without introducing additional quantization error. Classical entropy coding methods such as Huffman coding [9] and Zstandard [6] exploit symbol redundancy through variable-length codes, and recent works such as DFloat11 [36] and Huff-LLM [34] apply similar principles to compressed LLM weights. Nevertheless, lossless compression alone is not sufficient for efficient deployment. Variable-length decoding introduces sequential dependencies that limit GPU parallelism, while low-bit quantized tensors with high entropy provide limited redundancy to exploit. These limitations motivate a co-designed compression framework that combines quantization with hardware-friendly lossless coding.

2.2 Challenges in Combining Double Compression

Recent double-compression approaches apply lossless compression after quantization to further reduce model footprint. WCM [28] combines pruning with compression-aware quantization to improve weight compressibility, while QStore [23] exploits repeated integer codes in quantized weights and compresses their distribution using Huffman coding. However, these methods are primarily designed for unimodal LLMs and INT8 settings, leaving two key challenges unresolved for low-bit VLM compression.

(C1) Accuracy–Compression Trade-off. Compression-aware pruning or scaling can reshape weight distributions in ways that are harmful under limited precision. In particular, enforced sparsity and outlier-inducing transformations may amplify quantization error in sub-8-bit regimes, where representational capacity is already constrained. This issue is more critical for VLMs, whose performance depends on preserving fine-grained visual–linguistic alignment, yet existing double-compression methods provide limited evidence in such settings.

(C2) Metadata and Decoding Overhead. Although QStore reduces memory footprint by exploiting redundancy in quantized integer codes, its chunk-wise codebooks introduce non-negligible metadata overhead, especially when the symbol space is already small in low-bit regimes (*e.g.*, 6-bit). Moreover, conventional Huffman decoding remains inherently sequential, making it difficult to fully utilize parallel accelerators and potentially increasing inference latency.

In summary, prior double-compression methods either lack evidence for accurate sub-8-bit VLM quantization **(C1)** or overlook metadata efficiency and parallel decoding **(C2)**. These limitations call for a unified framework that jointly optimizes quantization and lossless compression for efficient VLM serving in resource-constrained environments.

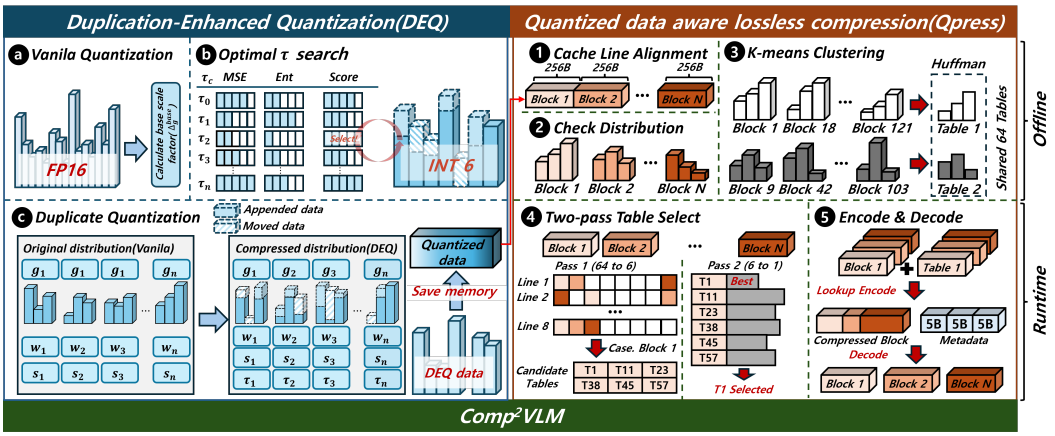


Figure 1: Overview of Hybrid Quantization-Compression Framework (Comp^{2}VLM).

3 Hybrid Quantization-Lossless Compression Framework: Comp^{2}VLM

3.1 Overview of Comp^{2}VLM Framework

This study proposes *Comp^{2}VLM*, a hybrid compression framework that jointly optimizes quantization and lossless compression for efficient VLM serving. The overall architecture is shown in Figure 1. The core design principle of *Comp^{2}VLM* is to co-design both compression stages such that each enhances the efficiency of the other. In the quantization stage, rather than simply minimizing the reconstruction error, the entropy of the quantized data is jointly optimized to improve the compression efficiency of the subsequent lossless compression. To this end, DEQ explores the trade-off between the quantization error and entropy when selecting the group-wise scale factor (τ) (ⓐ–ⓒ). In the lossless compression stage, QPress leverages the constrained symbol space of low-bit data quantized by DEQ to minimize the metadata overhead and enable hardware parallelism. QPress shares metadata across blocks through a pre-generated UHT and supports block-wise parallel decoding, enabling real-time lossless compression on hardware accelerators (❶–❺). As a result, *Comp^{2}VLM* achieves both ultra-low-bit compression—difficult to attain with quantization alone—and hardware efficiency, which is challenging for standalone lossless compression.

3.2 Duplication-Enhanced Quantization

Comp^{2}VLM applies lossless compression to quantized weights, whose compression efficiency is largely determined by the entropy of the quantized integer distribution. Motivated by this observation, we propose DEQ, which selects a group-wise scale factor by jointly considering reconstruction fidelity and compressibility. The key idea of DEQ is to select the optimal value from a set of candidate scale factors by considering the trade-off between the quantization error and entropy. Specifically, for b-bit signed uniform quantization, the integer range is defined as $[-q_{\max}, q_{\max}]$, where $q_{\max} = 2^{b-1} - 1$. For a quantization group g with elements $\{X_{g,j}\}_{j=1}^{N_g}$, the base scale is computed as

$$\Delta_g^{\text{base}} = \frac{\max_j |X_{g,j}|}{q_{\max}}, \tag{1}$$

where $X_{g,j}$ denotes the jth element of group g. Instead of directly using Δ_g^{base}, DEQ introduces a scale adjustment factor τ and searches over a predefined candidate set $\mathcal{T} = \{\tau_1, \ldots, \tau_C\}$. For each candidate τ_c, the adjusted scale is $\Delta_g(\tau_c) = \tau_c \Delta_g^{\text{base}}$, and quantization/dequantization are performed as

$$Q_{g,j}(\tau_c) = \text{clip}\left(\text{round}\left(\frac{X_{g,j}}{\Delta_g(\tau_c)}\right), -q_{\max}, q_{\max}\right), \hat{X}_{g,j}(\tau_c) = Q_{g,j}(\tau_c) \cdot \Delta_g(\tau_c). \tag{2}$$

where $Q_{g,j}(\tau_c)$ denotes the quantized integer value and $\hat{X}_{g,j}(\tau_c)$ denotes the value reconstructed via dequantization. Here, increasing τ enlarges the quantization step size, causing more values to

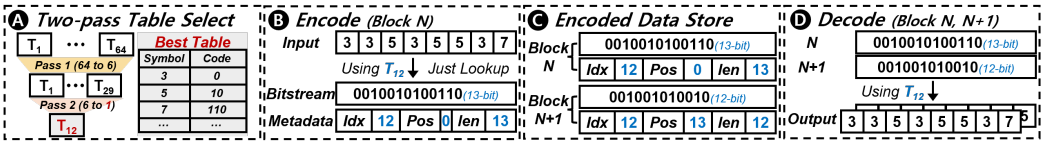


Figure 2: Workflow of QPress. Each block selects one of 64 shared UHTs via a two-pass search, stores only 5 bytes of metadata, and is decoded in parallel through simple table lookups.

collapse into the same integer bin. This increases symbol redundancy and lowers entropy, but may also increase reconstruction error.

To select the optimal τ from the predefined candidates, DEQ evaluates each candidate using two group-wise criteria. First, reconstruction fidelity is assessed via the group-wise mean squared error (MSE):

$$D_g(\tau_c) = \frac{1}{N_g} \sum_{j=1}^{N_g} \left(X_{g,j} - \hat{X}_{g,j}(\tau_c) \right)^2 . \tag{3}$$

Second, compressibility is estimated by the Shannon entropy of the quantized integer distribution:

$$p_{g,k}(\tau_c) = \frac{1}{N_g} \sum_{j=1}^{N_g} \mathbf{1}\{Q_{g,j}(\tau_c) = k\}, \quad H_g(\tau_c) = - \sum_{k=-q_{\max}}^{q_{\max}} p_{g,k}(\tau_c) \log_2(p_{g,k}(\tau_c) + \epsilon) . \tag{4}$$

Here, $p_{g,k}(\tau_c)$ denotes the empirical frequency of integer symbol k, $\mathbf{1}\{\cdot\}$ is the indicator function, and ϵ is a small constant for numerical stability. However, directly combining $D_g(\tau_c)$ and $H_g(\tau_c)$ can bias scale selection because groups within the same layer may have different magnitudes and variances. DEQ therefore applies group-wise min–max normalization over $\mathcal{T}$, capturing the local trade-off landscape of each group:

$$\widetilde{D}_g(\tau_c) = \frac{D_g(\tau_c) - \min_{c'} D_g(\tau_{c'})}{\max_{c'} D_g(\tau_{c'}) - \min_{c'} D_g(\tau_{c'}) + \epsilon}, \quad \widetilde{H}_g(\tau_c) = \frac{H_g(\tau_c) - \min_{c'} H_g(\tau_{c'})}{\max_{c'} H_g(\tau_{c'}) - \min_{c'} H_g(\tau_{c'}) + \epsilon}. \tag{5}$$

Through Eq. (5), both $\widetilde{D}_g(\tau_c)$ and $\widetilde{H}_g(\tau_c)$ are normalized to $[0, 1]$, alleviating bias in τ selection arising from scale variations across groups. DEQ then defines a weighted score that combines the two normalized metrics as follows:

$$S_g(\tau_c) = \alpha \cdot \widetilde{D}_g(\tau_c) + (1 - \alpha) \cdot \widetilde{H}_g(\tau_c), \quad \tau_g^\star = \arg\min_{\tau_c \in \mathcal{T}} S_g(\tau_c). \tag{6}$$

In Eq. (6), the coefficient $\alpha \in [0, 1]$ controls the accuracy–compression trade-off: larger values prioritize reconstruction fidelity, while smaller values favor entropy reduction. We set $\alpha = 0.4$ to emphasize compressibility for efficient lossless coding, with an ablation study provided in Appendix A.

3.3 Quantized Data-aware Lossless Compression

Low-bit quantized data still exhibits storage inefficiency relative to its theoretical entropy, presenting opportunities for further lossless compression. However, conventional methods such as Huffman coding and zstd incur substantial metadata overhead and inherent sequential dependencies that limit hardware parallelism, making them unsuitable for decoding on the inference path [36]. To address these challenges, we propose QPress, a lossless compression framework that combines shared metadata structures with hardware-friendly block-wise compression. Figure 2 illustrates the overall encoding and decoding workflow, and Algorithm 1 details the compression procedure.

As illustrated in Figure 2, the core idea of QPress is to minimize metadata overhead through UHTs. Since low-bit quantized data has a highly constrained symbol space, a small set of shared tables can efficiently approximate diverse data distributions. To construct the UHTs, we apply DEQ and

Algorithm 1 Quantized Data-aware Lossless Compression

Require: Tensor $\mathbf{X} \in \mathbb{Z}^n$, UHTs $\mathcal{T} = \{T_j\}_{j=1}^{64}$
Ensure: Bitstream $\mathcal{B}$, metadata $\mathcal{M}$, reconstructed $\mathbf{X}'$
1: *Notation:* f_s frequency, $\ell(T_j, s)$ code length, (t, o, l) table index/offset/length
2: **Offline:** generate $\{T_j\}_{j=1}^{64}$ via K-means on calibration data
3: **for** each $\mathbf{B}_i \in \text{Partition}(\mathbf{X})$ in parallel ▷ **Compression do**
4: **Pass 1:** $\mathcal{S}_k \leftarrow$ top-12 symbols; $\mathcal{C} \leftarrow$ top-6 of $\{L_j = \sum_{s \in \mathcal{S}_k} f_s \ell(T_j, s)\}$
5: **Pass 2:** $t_i \leftarrow \arg\min_{T_j \in \mathcal{C}} \sum_{s \in \mathbf{B}_i} f_s \ell(T_j, s)$; encode $\mathbf{B}_i$ with T_{t_i}; store (t_i, o_i, l_i) to $\mathcal{M}$
6: **end for**
7: **for** each $(t_i, o_i, l_i) \in \mathcal{M}$ in parallel ▷ **Decompression do**
8: $\mathbf{B}'_i \leftarrow \text{Decode}(\mathcal{B}[o_i : o_i + l_i], T_{t_i})$
9: **end for**

collect symbol frequency distributions from the quantized weight tensors themselves and from the quantized activations obtained by passing calibration inputs through the model. We then apply K-means clustering to extract 64 representative distribution patterns and generate a UHT from each cluster centroid. Weights and activations use separate banks of 64 UHTs. These UHTs are generated once offline and reused across all inference passes of the same model, eliminating the per-block table storage overhead required by conventional approaches. Each block requires only 5 bytes of metadata, enabling the effective bit-width to approach the theoretical entropy while preserving the compression efficiency of variable-length coding.

In addition to improving table storage efficiency, QPress introduces a two-pass table selection algorithm to minimize the computational cost of identifying the optimal UHT for each block during encoding (**A**). In Pass 1, to reduce per-block overhead, we extract only the top-12 most frequent symbols and compute approximate expected code lengths to select the top-6 candidate tables (Algorithm 1 Line 4). In Pass 2, we compute the exact code lengths using all symbol frequencies within the block for these candidates and select the optimal table (Algorithm 1 Line 5). Notably, this selection process relies solely on lookup and summation of symbol frequencies and corresponding UHT code lengths, avoiding actual bit manipulation during the encoding phase. By encoding only with the final selected table, QPress achieves near-optimal compression ratios at only $6/64 \approx 9.4\%$ of the computational cost of a greedy search.

To enable efficient execution on inference accelerators, QPress adopts a hardware-friendly block-wise compression scheme. The input is partitioned into independent 256-byte–aligned blocks, matching modern cache line sizes and confining the sequential dependencies of variable-length coding to the block level. Each block stores only 5 bytes of metadata—the table index, bit offset, and encoded length (**B**)—while all compressed blocks are concatenated into a single bitstream (**C**). As illustrated in Figure 2, despite variable block lengths, each thread can directly locate its corresponding bit range using block metadata alone, enabling parallel decoding via simple table lookups with predefined UHTs (Algorithm 1, Lines 7–9, **D**). This design removes inter-block dependencies, allowing low-latency parallel decoding even with variable-length codes.

4 Experimental Results

4.1 Experimental Setup

Models and Benchmarks. We evaluate *Comp^{2}VLM* on diverse VLMs and benchmarks using the LMMs-Eval framework [35]. All experiments are conducted with PyTorch on an NVIDIA H100 GPU. To assess scalability, we use Qwen2-VL-7B [2], LLaVA-OV-7B [15], and InternVL2-8B [4] as mid-scale models, and Qwen2-VL-72B and LLaVA-OV-72B for large-scale evaluation. We evaluate text recognition and document understanding on TextVQA [24] and OCRBench [21], and visual reasoning and scene understanding on ScienceQA [22], VizWiz [8], and SEED [14]. We compare against representative PTQ baselines, including Round-To-Nearest (RTN), SmoothQuant (SQ) [31], QuaRot [1], MBQ [16], and MQuant [33], under the standard W4A8 setting used in prior VLM quantization studies. Additional benchmark results and W6A6 comparisons are provided in Appendices B.1 and B.2.

Table 1: Main results on the models of LLaVA-OV, InternVL2, and Qwen2-VL families. All quantization methods use W4A8. Subscripts in Ours denote bit-width (W: weight, A: activation).

MODEL	METHOD	SCIENCEQA	TEXTVQA	VIZWIZ	OCRBENCH	SEED
QWEN2-VL 7B	BF16	84.7	84.4	68.8	84.2	76.4
	RTN	78.9	40.2	58.9	42.2	74.9
	SQ [31]	80.9	46.3	52.3	41.1	75.0
	QUAROT [1]	-	71.4	-	67.0	-
	MBQ [16]	81.2	75.0	59.3	72.8	**75.1**
	MQUANT [33]	-	**84.3**	-	**82.4**	-
	OURS	**84.1**$_{\text{W3.5A3.4}}$	83.3$_{\text{W3.5A3.2}}$	**66.8**$_{\text{W3.5A3.3}}$	81.0$_{\text{W3.5A3.3}}$	74.8$_{\text{W3.5A3.3}}$
LLAVA-OV 7B	BF16	90.0	76.1	60.4	62.2	72.0
	RTN	88.3	61.5	58.2	40.1	50.3
	SQ [31]	87.1	56.9	56.7	32.0	42.7
	MBQ [16]	88.5	68.3	58.9	52.3	67.7
	OURS	**89.3**$_{\text{W3.5A4.9}}$	**73.0**$_{\text{W3.5A4.6}}$	**58.9**$_{\text{W3.5A4.5}}$	**60.0**$_{\text{W3.5A4.5}}$	**71.3**$_{\text{W3.5A4.6}}$
INTERNVL2 8B	BF16	96.2	77.7	60.7	76.5	71.6
	RTN	95.5	40.1	57.1	30.2	50.3
	SQ [31]	95.1	46.5	54.2	31.0	42.7
	QUAROT [1]	-	49.1	-	36.1	-
	MBQ [16]	95.8	72.3	56.5	**73**	67.7
	MQUANT [33]	-	**76.6**	-	72.5	-
	OURS	**95.9**$_{\text{W3.6A4.7}}$	75.5$_{\text{W3.6A4.6}}$	**61.9**$_{\text{W3.6A4.6}}$	72.7$_{\text{W3.6A4.7}}$	**71.2**$_{\text{W3.6A4.6}}$
QWEN2-VL 72B	BF16	92.5	85.5	75.8	88.3	77.6
	RTN	90.1	29.2	71.3	42.6	**76.6**
	SQ [31]	88.8	47.1	69.7	41.3	76.4
	QUAROT [1]	-	71.9	-	70.1	-
	MBQ [16]	89.6	80.5	73.6	77.5	76.3
	MQUANT [33]	-	85.0	-	86.8	-
	OURS	**92.1**$_{\text{W3.5A3.6}}$	**85.5**$_{\text{W3.5A3.5}}$	**75.2**$_{\text{W3.5A3.5}}$	**87.7**$_{\text{W3.5A3.6}}$	76.5$_{\text{W3.5A3.6}}$
LLAVA-OV 72B	BF16	90.0	79.3	69.2	73.2	78.1
	RTN	89.0	74.5	64.7	64.5	76.6
	SQ [31]	89.1	74.4	65.7	64.2	76.6
	MBQ [16]	89.2	73.3	65.7	64.4	76.5
	OURS	**89.9**$_{\text{W3.5A5.0}}$	**77.4**$_{\text{W3.5A4.7}}$	**66.2**$_{\text{W3.5A4.6}}$	**70.1**$_{\text{W3.5A4.7}}$	**78.0**$_{\text{W3.5A4.6}}$

Comp^{2}VLM Configuration. DEQ is designed for the 5–7-bit regime, where quantization alone still suffers from accuracy loss but lossless compression can provide meaningful additional savings. Below 4-bit, metadata such as Huffman tables can dominate the compressed representation, while above 8-bit, existing PTQ methods already preserve accuracy well, reducing the benefit of DEQ. We adopt W6A6 as the default configuration because matched weight and activation precision enables more efficient packing and vectorized kernels than mixed-precision settings such as W4A8, while still providing sufficient redundancy for lossless compression. Results for other precisions, including W7A7 and W5A5, are reported in Appendix B.3. The DEQ scale factor τ and the universal Huffman table (UHT) used in QPress are determined during calibration and reused at inference time. The calibration set is a uniformly random 5% subset of the training splits of ScienceQA [22], TextVQA [24], and VizWiz [8], drawn with a fixed seed (3,392 samples in total), and contains no evaluation instances. Each UHT bank is built once per model and precision and shared across all benchmarks, so benchmarks without a training split, including OCRBench and SEED, are evaluated without any benchmark-specific calibration data. To measure QPress encoding and decoding overhead, we implement a custom GPU kernel in Triton [25] with CUDA 12.8.

4.2 Main Results

Table 1 compares *Comp^{2}VLM* with representative VLM quantization methods across models and benchmarks. The subscripts in the *Comp^{2}VLM* entries indicate the effective bit-width after DEQ and QPress. Since activation entropy depends on input characteristics, the effective activation precision varies across benchmarks. Overall, *Comp^{2}VLM* maintains competitive accuracy while reducing the effective precision to W3.5 $\sim$ 3.6A3.2 $\sim$ 4.9, substantially below the W4A8 setting used by prior VLM quantization methods [1, 16, 31, 33]. This is notable because aggressive activation quantization often disrupts visual–linguistic alignment, yet *Comp^{2}VLM* preserves strong performance with only 3.2–4.9 effective activation bits.

Table 2: Entropy comparison of quantized weights on InternVL2-8B.

Bit-width	RTN	SQ [31]	QuaRot [1]	MQuant [33]	DEQ
7-bit	5.98	5.99	6.07	6.04	**4.40**
6-bit	4.96	4.97	5.05	5.12	**3.43**
5-bit	3.91	3.93	4.00	4.24	**2.76**

The robustness of *Comp^{2}VLM* stems from its entropy-aware scale selection. In asymmetric settings such as W4A8, low-precision weight quantization can dominate the error budget and degrade accuracy [29]. DEQ mitigates this imbalance by incorporating entropy into the scale-selection score in Eq. (6), encouraging more regular quantized distributions that reduce error accumulation while improving compressibility. As a result, *Comp^{2}VLM* achieves strong accuracy even at lower effective activation precision than existing methods.

Across most models, *Comp^{2}VLM* performs particularly well on ScienceQA and VizWiz, achieving 84.1%/66.8% on Qwen2-VL-7B, 89.3%/58.9% on LLaVA-OV-7B, and 95.9%/61.9% on InternVL2-8B, respectively. It also scales robustly to 72B models, preserving accuracy under substantially reduced effective bit-width. Although MQuant achieves higher accuracy in some text-centric benchmarks such as TextVQA and OCRBench, it relies on 8-bit activations, whereas *Comp^{2}VLM* operates at 3.2–4.9 effective activation bits. These results demonstrate that jointly designing quantization and lossless compression enables ultra-low effective precision for VLMs with minimal accuracy degradation.

4.3 Ablation Studies

4.3.1 Entropy Reduction Effect of DEQ

Table 2 compares the Shannon entropy of quantized weights on InternVL2-8B under 5-, 6-, and 7-bit precision. DEQ consistently achieves the lowest entropy across all bit-widths, reducing entropy to 4.40, 3.43, and 2.76 bits at 7-, 6-, and 5-bit precision, respectively. These correspond to relative reductions of 26.4–27.5%, 30.8–33.0%, and 29.4–31.0% over RTN and representative baselines [1, 31, 33] at each precision level.

Existing quantization methods produce similar entropy values because they mainly optimize reconstruction fidelity or activation/weight scaling, without explicitly considering compressibility. In contrast, DEQ incorporates entropy into the scale-selection objective, encouraging more duplicated integer symbols after quantization. Since Shannon entropy lower-bounds the achievable code length of lossless compression, the average 1.4-bit entropy reduction over RTN indicates that approximately 28% additional compression can be theoretically achieved at the same nominal bit-width. These results confirm that DEQ produces quantized distributions that are better aligned with subsequent lossless compression.

4.3.2 Effectiveness of QPress in Diverse Low-bit Compression

Table 3 evaluates QPress on Qwen2-VL-7B by applying Huffman coding [9], zstd [6], and QPress to the same low-bit quantized weights and activations. Across 5-, 6-, and 7-bit precision, QPress consistently improves compression efficiency over both baselines. On average, it improves the compression ratio over Huffman coding by 7.2% for weights and 12.1% for activations, and over zstd by 8.4% for weights and 2.5% for activations.

The improvement patterns differ across baselines and data types. Huffman coding suffers from symbol-dependent table overhead, which becomes more pronounced for activations whose distributions vary across inputs. QPress avoids this issue through fixed, lightweight block-level metadata. Conversely, zstd is effective for locally redundant activations but less suitable for quantized weights with limited spatial correlation. By using calibration-derived universal Huffman tables, QPress provides stable compression gains for both weights and activations while preserving a hardware-friendly decoding structure.

Table 3: Effectiveness of QPress over Huffman and zstd baselines.

PRECISION	HUFFMAN (%)		ZSTD (%)	
	W	A	W	A
5-BIT	8.2	13.3	11.5	3.5
6-BIT	7.5	12.2	8.2	2.2
7-BIT	5.9	10.7	5.4	1.7
AVERAGE	7.2	12.1	8.4	2.5

Table 4: Lossless Compression Hardware Overhead on LLaVA-OV and Qwen2-VL families.

MODELS	QUANT ONLY (ms)	OVERHEAD (%)		
		HUFFMAN	ZSTD	QPRESS
LLAVA-OV-0.5B	635.4	72.24	48.85	**4.01**
LLAVA-OV-7B	842.6	68.67	50.40	**4.13**
QWEN2-VL-0.5B	627.4	66.40	48.45	**3.98**
QWEN2-VL-7B	821.9	64.81	49.34	**3.89**

4.3.3 Sensitivity of QPress to the Calibration Source

Activation UHTs are built from calibration inputs, so their rate could depend on how closely the calibration source matches the evaluation data. We therefore regenerate the activation UHTs for Qwen2-VL-7B (W6A6), changing only the calibration source and keeping every other step of Section 3.3 fixed. With the default calibration set described in Section 4.1, the effective activation bit-width is 3.42 on ScienceQA and 3.20 on TextVQA. Replacing it with the COCO Captions [3] training split, which is disjoint from both benchmarks, yields 3.43 and 3.21, an increase of only 0.01 bits. Calibrating on the VizWiz training split alone yields 3.45 and 3.23. The effective weight bit-width remains 3.53 in every case because weight UHTs do not depend on calibration inputs. The reported activation rates therefore do not materially depend on calibration data drawn from the evaluated benchmarks.

4.4 Quantized Data-aware Lossless Compression Hardware Overhead

To assess the hardware efficiency of QPress, we measure the end-to-end latency overhead of adding lossless compression to quantized VLM inference. We compare QPress with Huffman coding [9] and zstd [6] on LLaVA-OV and Qwen2-VL families, using Quant-Only inference as the baseline. QPress is implemented as a custom Triton GPU kernel with two-pass table selection and block-wise encoding/decoding performed entirely on the GPU. The kernel uses tiled block processing to reduce register pressure and exploits block-level parallelism to process many compressed blocks concurrently.

As shown in Table 4, conventional lossless compression introduces substantial latency overhead: Huffman coding adds 64.81–72.24%, and zstd adds 48.45–50.40% over Quant-Only inference. In contrast, QPress incurs only 3.89–4.13% overhead across all evaluated models. These results demonstrate that QPress makes lossless compression practical for VLM serving by combining strong compression efficiency with low-latency, GPU-parallel decoding.

4.5 End-to-end Inference Throughput

Figure 3 reports the end-to-end throughput gain of *Comp^{2}VLM* over the W4A8 RTN baseline. We integrate the DEQ quantization/dequantization path and the Triton-based QPress compression/decompression kernel into both the prefill and decode stages of Qwen2-VL-7B and InternVL2-8B. The input length is fixed to 512 tokens, while the output length varies from 256 to 8192 tokens. All gains are normalized to the W4A8 RTN throughput measured under the same setting at 128 output tokens. Both the W4A8 RTN baseline and *Comp^{2}VLM* run on the same custom Triton low-bit kernel family under PyTorch on an NVIDIA H100 GPU with batch size 1. The baseline stores weights in a packed 4-bit layout, fuses dequantization into the GEMM/GEMV prologue, and uses tensor-core execution paths. This matched implementation isolates the effect of the quantized and compressed representation from differences in kernel maturity, and the results are not intended as a comparison against independently optimized engines such as TensorRT-LLM or vLLM.

Comp^{2}VLM consistently improves throughput across all output token lengths, achieving average gains of 6.6% on Qwen2-VL-7B and 5.6% on InternVL2-8B. Notably, the margin over the baseline progressively widens as the output token length increases, reaching maximum gains of 10.8% and 8.5% at 8192 output tokens. This improvement can be attributed to the effective bit-width achieved by *Comp^{2}VLM* (*i.e.,* W3.5 for Qwen2-VL-7B and W3.6 for InternVL2-8B), which yields approximately

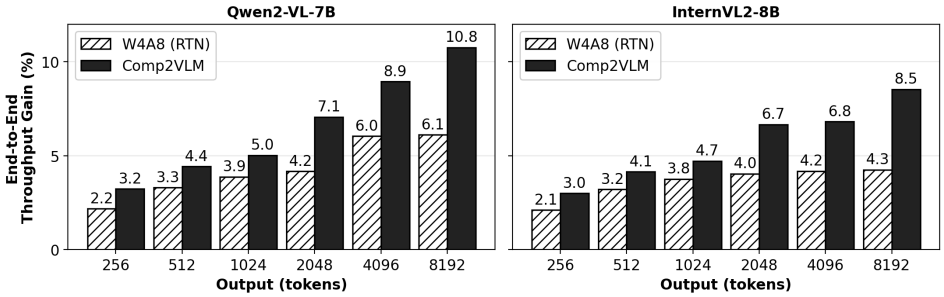


Figure 3: End-to-end inference throughput gain of *Comp*^{2}*VLM* over the W4A8 RTN baseline as a function of output token length on Qwen2-VL-7B and InternVL2-8B. All values are reported as percent gains normalized to the W4A8 RTN throughput at an output length of 128 tokens.

Table 5: Edge-device measurements of throughput, energy, and peak memory on Jetson AGX Orin (Qwen2-VL-7B).

METHOD	THROUGHPUT (TOK/S)	ENERGY (J/TOKEN)	MEMORY (GB)
BF16	12.41	3.44	13.64
RTN W6A6	26.26	1.88	5.59
OURS	26.24	1.87	3.42

12.5% weight memory savings over W4A8. Because autoregressive decoding is memory-bound, this reduction in weight traffic shortens each decode step. Furthermore, as the output token length increases, the fixed cost of the prefill stage becomes amortized while the memory bandwidth savings during the decode stage take on a larger share of the overall throughput, providing a consistent explanation for the progressive widening of the gap noted above. This gain stems from the reduced weight traffic rather than from faster computation. Relative to the same quantized model without QPress, QPress mainly reduces the memory and storage footprint while adding only a small runtime cost (Section 4.4).

4.6 On-Device Evaluation on Jetson AGX Orin

We evaluate on an NVIDIA Jetson AGX Orin Developer Kit (MAXN mode, fixed clocks), whose memory bandwidth is roughly 16$\times$ lower than that of an H100, making it a stringent platform for measuring decoder overhead. We implemented two custom CUDA kernels: a fused unpack–dequantize–GEMV path for W6 and a shared-LUT, block-parallel QPress decoder. Bit-exact reconstruction was verified on-device, so QPress does not alter the accuracy of the quantized model.

As shown in Table 5, *Comp*^{2}*VLM* differs from RTN W6A6 by only 0.08% in throughput and 0.5% in energy. Because both use the same W6 quantized computation and differ in the QPress storage and decoding path, this comparison directly estimates QPress's runtime cost. Relative to BF16, *Comp*^{2}*VLM* provides 2.11$\times$ higher throughput and 46% lower energy per token while reducing peak memory from 13.64 GB to 3.42 GB.

5 Conclusion

This paper presents *Comp*^{2}*VLM*, a hybrid compression framework that combines entropy-aware quantization with low-overhead, parallel lossless coding for efficient real-time serving of VLMs. The proposed **DEQ** module adaptively selects group-wise scaling factors to balance quantization error and entropy reduction, enhancing downstream compressibility, while **QPress** leverages a shared UHT with block-wise parallel decoding for high compression efficiency at minimal latency overhead. Across diverse VLMs, *Comp*^{2}*VLM* preserves competitive accuracy at an effective bit-width of W3.5A3.2 $\sim$ 4.9: **DEQ** reduces entropy by up to **33%**, and **QPress** improves compression efficiency by up to **13.3%** over conventional lossless methods with only $\sim$**4%** latency overhead, highlighting *Comp*^{2}*VLM* as a practical solution for deploying VLMs in resource-constrained environments.

Acknowledgments and Disclosure of Funding

This work was partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2026-25481790) and the Technology Innovation Program(or Industrial Strategic Technology Development Program)(RS-2025-02307330, Development and Validation of On-Device AI Semiconductor for Manufacturing Automation Robots through sVLM-Based Situational Awareness) funded By the Ministry of Trade Industry & Energy(MOTIE, Korea).

References

[1] S. Ashkboos, A. Mohtashami, M. L. Croci, B. Li, P. Cameron, M. Jaggi, D. Alistarh, T. Hoefler, and J. Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

[2] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 1(2):3, 2023.

[3] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[4] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024.

[5] D. Choi and H. Kim. GradQ-ViT: Robust and efficient gradient quantization for vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16019–16027, 2025. doi: 10.1609/aaai.v39i15.33759.

[6] Y. Collet. Rfc 8878: Zstandard compression and the'application/zstd'media type, 2021.

[7] C. Fu, P. Chen, Y. Shen, Y. Qin, M. Zhang, X. Lin, J. Yang, X. Zheng, K. Li, X. Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[8] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3608–3617, 2018.

[9] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 2007.

[10] Y. Jin, J. Li, T. Gu, Y. Liu, B. Zhao, J. Lai, Z. Gan, Y. Wang, C. Wang, X. Tan, et al. Efficient multimodal large language models: A survey. *Visual Intelligence*, 3(1):27, 2025.

[11] B. J. Kang, D. Choi, and H. Kim. GenPTQ: Green post-training quantization for large-scale ASR models with mixed-precision bit allocation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9561–9574, Suzhou, China, 2025. Association for Computational Linguistics. URL `https://aclanthology.org/2025.findings-emnlp.566`.

[12] N. J. Kim and H. Kim. Trunk pruning: Highly compatible channel pruning for convolutional neural networks without fine-tuning. *IEEE Transactions on Multimedia*, 26:5548–5559, 2024. doi: 10.1109/TMM.2023.3336528.

[13] K. Koo and H. Kim. V-skp: Vectorized kernel-based structured kernel pruning for accelerating deep convolutional neural networks. *IEEE Access*, 11:118547–118557, 2023.

[14] B. Li, Y. Ge, Y. Ge, G. Wang, R. Wang, R. Zhang, and Y. Shan. Seed-bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13299–13308, 2024.

[15] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[16] S. Li, Y. Hu, X. Ning, X. Liu, K. Hong, X. Jia, X. Li, Y. Yan, P. Ran, G. Dai, et al. Mbq: Modality-balanced quantization for large vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4167–4177, 2025.

[17] Y. Li, Y. Du, K. Zhou, J. Wang, W. X. Zhao, and J.-R. Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305, 2023.

[18] F. Lin. Vision language models: A survey of 26k papers. *arXiv preprint arXiv:2510.09586*, 2025.

[19] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6:87–100, 2024.

[20] Y. Liu, H. Duan, Y. Zhang, B. Li, S. Zhang, W. Zhao, Y. Yuan, J. Wang, C. He, Z. Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023.

[21] Y. Liu, Z. Li, M. Huang, B. Yang, W. Yu, C. Li, X. Yin, C.-l. Liu, L. Jin, and X. Bai. Ocrbench: On the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024. doi: 10.1007/s11432-024-4235-6.

[22] P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 2507–2521, 2022.

[23] R. Shah, Z. Li, and T. Abdelzaher. Qstore: Quantization-aware compressed model storage. *arXiv preprint arXiv:2505.04081*, 2025.

[24] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8317–8326, 2019.

[25] P. Tillet, H.-T. Kung, and D. Cox. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pages 10–19, 2019.

[26] Z. Wan, X. Wang, C. Liu, S. Alam, Y. Zheng, J. Liu, Z. Qu, S. Yan, Y. Zhu, Q. Zhang, et al. Efficient large language models: A survey. *arXiv preprint arXiv:2312.03863*, 2023.

[27] C. Wang, Z. Wang, X. Xu, Y. Tang, J. Zhou, and J. Lu. Q-vlm: Post-training quantization for large vision-language models. *Advances in Neural Information Processing Systems*, 37: 114553–114573, 2024.

[28] W. Wang, Y. Mao, D. Tang, H. Du, N. Guan, and C. J. Xue. When compression meets model compression: Memory-efficient double compression for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16973–16983, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics.

[29] X. Wu, Z. Yao, and Y. He. Zeroquant-fp: A leap forward in llms post-training w4a8 quantization using floating-point formats. *arXiv preprint arXiv:2307.09782*, 2023.

[30] xAI. RealWorldQA. `https://x.ai/blog/grok-1.5v`, 2024.

[31] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pages 38087–38099. PMLR, 2023.

[32] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, 2024.

[33] J. Yu, S. Zhou, D. Yang, S. Li, S. Wang, X. Hu, C. Xu, Z. Xu, C. Shu, and Z. Yuan. Mquant: Unleashing the inference potential of multimodal large language models via static quantization. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 1783–1792, 2025.

[34] P. Yubeaton, T. Mahmoud, S. Naga, P. Taheri, T. Xia, A. George, Y. Khalil, S. Q. Zhang, S. Joshi, C. Hegde, et al. Huff-llm: End-to-end lossless compression for efficient llm inference. *arXiv preprint arXiv:2502.00922*, 2025.

[35] K. Zhang, B. Li, P. Zhang, F. Pu, J. A. Cahyono, K. Hu, S. Liu, Y. Zhang, J. Yang, C. Li, and Z. Liu. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*, 2024.

[36] T. Zhang, Y. Sui, S. Zhong, V. Chaudhary, X. Hu, and A. Shrivastava. 70% size, 100% accuracy: Lossless llm compression for efficient gpu inference via dynamic-length float. *arXiv preprint arXiv:2504.11651*, 2025.

A Analysis of DEQ Characteristics

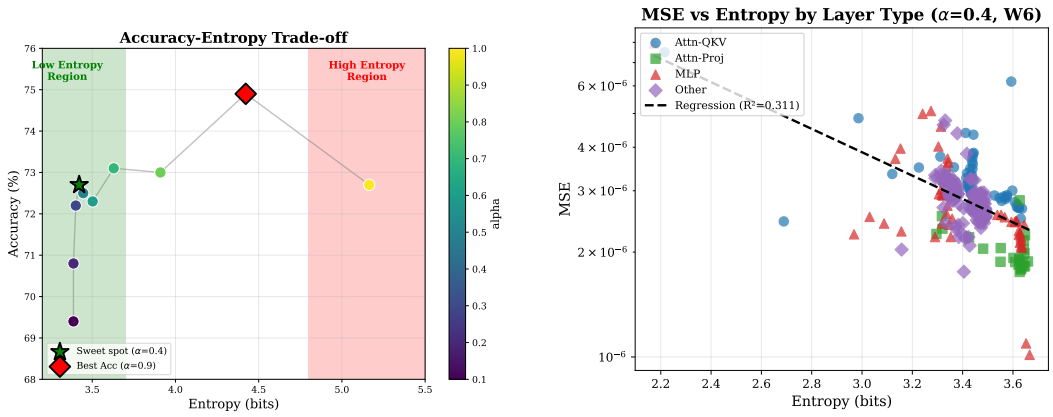
A.1 Accuracy-Compression Trade-off by α

Figure 4 presents the influence of DEQ's entropy-aware optimization on both model accuracy and compressibility. Figure 4a illustrates the trade-off between accuracy and entropy as a function of the entropy weighting parameter α. As α decreases, greater emphasis is placed on entropy minimization, leading to lower entropy levels that are favorable for compression. However, excessively low values of α impair reconstruction fidelity, thereby compromising model accuracy. In contrast, setting $\alpha = 1.0$ corresponds to minimizing only the MSE, equivalent to conventional RTN, and results in higher entropy. Experimental results indicate that $\alpha = 0.4$ serves as an empirical optimum, simultaneously achieving low entropy (green region) and high accuracy. This demonstrates that DEQ explicitly incorporates compression efficiency into its optimization objective, extending beyond conventional MSE minimization.

Figure 4b visualizes the relationship between MSE and entropy across layers at $\alpha = 0.4$, with layers grouped by type. A moderate positive correlation ($R^2 = 0.311$) was observed, suggesting that layers more resistant to quantization tend to exhibit both higher entropy and greater reconstruction error. This variation in layer-wise characteristics underscores the need for adaptive quantization strategies. It also supports the effectiveness of DEQ's group-wise τ selection in dynamically adjusting the entropy–MSE trade-off based on the unique properties of each layer.

A.2 Layer-wise Information Flow Analysis

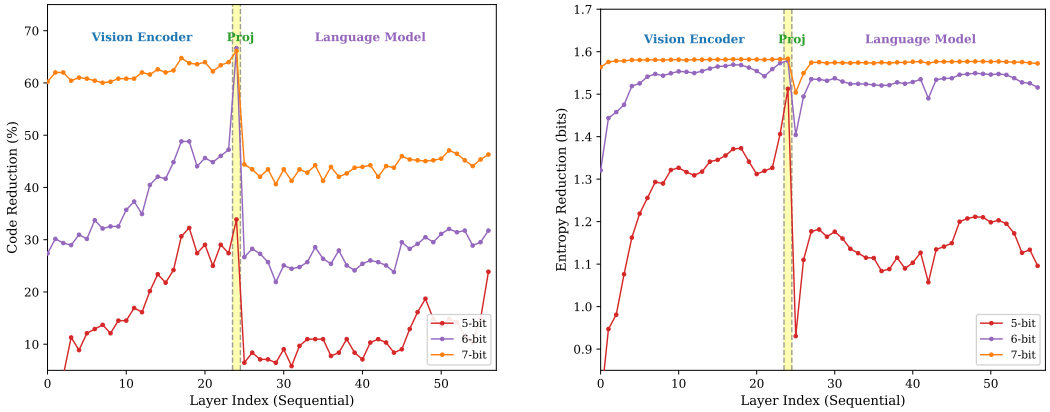
Figure 5 illustrates the compression behavior across modalities within the VLM architecture at the layer level. Both code reduction (Figure 5a) and entropy reduction (Figure 5b) clearly delineate the contrast between vision and language modalities. In the vision encoder, both metrics remain consistently high and stable, especially at 6–7-bit quantization, indicating high compressibility. In contrast, the language model exhibits substantially lower reduction margins even at the same bit-width, implying greater weight diversity and less compression-favorable characteristics. Notably, both figures reveal a sharp spike at the projection boundary, suggesting that this layer harbors high redundancy and is more amenable to compression. These observations confirm that visual and linguistic components exhibit genuinely heterogeneous weight statistics, which is precisely the property *Comp*^{2}*VLM* is designed to absorb. Rather than imposing a hand-crafted per-modality policy, the shared objective adapts to this heterogeneity through its data-dependent solution: group-wise scale selection in DEQ and block-wise table selection in QPress resolve differently on vision, projection, and language components, as reflected in the distinct reduction margins above. *Comp*^{2}*VLM* therefore operates as a modality-responsive framework in which a single optimization rule yields modality-conditioned compression behavior, without architecture- or modality-specific redesign.



(a) Accuracy-entropy trade-off

(b) MSE-entropy correlation by layer

Figure 4: Analysis of DEQ's entropy-accuracy characteristics on InternVL2-8B (W6).



(a) Code reduction (%)

(b) Entropy reduction (bits)

Figure 5: Layer-wise compression characteristics across VLM modalities on InternVL2-8B.

Table 6: Evaluation of Comp^{2}VLM across diverse benchmarks on various VLMs.

Model	Setting	MMBench-EN (GPT Score)	MME (Perception)	POPE (Accuracy)	RealWorldQA (Exact Match)
Qwen2-VL 7B	BF16	78.35	1678.78	88.59	65.62
	W6A6	$78.44_{\text{W3.5A4.5}}$	$1684.53_{\text{W3.5A4.5}}$	$88.07_{\text{W3.5A4.5}}$	$67.19_{\text{W3.5A4.5}}$
LLaVA-OV 7B	BF16	80.84	1578.66	89.18	66.67
	W6A6	$79.73_{\text{W3.5A4.6}}$	$1525.03_{\text{W3.5A4.6}}$	$89.31_{\text{W3.5A4.6}}$	$65.36_{\text{W3.5A4.5}}$
InternVL2 8B	BF16	82.04	1630.26	87.71	65.75
	W6A6	$80.67_{\text{W3.6A4.7}}$	$1650.62_{\text{W3.6A4.8}}$	$86.69_{\text{W3.6A4.8}}$	$62.35_{\text{W3.6A4.7}}$

B Comprehensive Experimental Evaluations.

B.1 Evaluation on Additional Benchmarks

Table 6 reports the performance of *Comp^{2}VLM* on four benchmarks not included in the main evaluation: MMBench-EN [20], MME [7] Perception, POPE [17], and RealWorldQA [30]. These benchmarks cover a broader range of vision-language capabilities, including instruction following, visual perception, hallucination robustness, and real-world question answering, providing a more comprehensive assessment of our method beyond the main experimental setup.

On Qwen2-VL-7B, *Comp^{2}VLM* achieves near-baseline or superior performance across all four benchmarks under W6A6 quantization at a uniform effective bitwidth of W3.5A4.5. Notably, MME Perception (1684.53 vs. 1678.78) and RealWorldQA (67.19 vs. 65.62) both exceed the BF16 baseline, while MMBench-EN and POPE exhibit only marginal degradations of 0.09 and 0.52 points, respectively.

On InternVL2-8B, our method maintains competitive performance at effective bitwidths of W3.6A4.7–W3.6A4.8. MME Perception again surpasses the baseline (1650.62 vs. 1630.26), and POPE Accuracy degrades by only 1.02 points. MMBench-EN and RealWorldQA show drops of 1.37 and 3.40 points, respectively, consistent with the accuracy-compression trade-off observed in the main results.

These additional results corroborate the findings of the main experiments, confirming that *Comp^{2}VLM* generalizes robustly across diverse benchmark types and VLM architectures beyond those evaluated in the primary comparison.

B.2 Additional Results on 6-bit Quantization

Table 7 presents the quantization performance under W6A6 settings on Qwen2-VL-7B, InternVL2-8B, and LLaVA-OV-7B across five benchmarks: ScienceQA, TextVQA, VizWiz, OCRBench, and SEED. On Qwen2-VL-7B, our method achieves 84.1 on ScienceQA and 83.3 on TextVQA, outperforming

Table 7: Comparison of quantization methods under W6A6 settings on Qwen2-VL-7B, InternVL2-8B, and LLaVA-OV-7B across five vision-language benchmarks.

	METHOD	SCIENCEQA	TEXTVQA	VIZWIZ	OCRBENCH	SEED
QWEN2-VL 7B	BASELINE	84.7	84.4	68.8	84.2	76.4
	MBQ	83.2	80.1	**66.9**	78.0	71.2
	MQUANT	80.4	80.2	66.3	77.0	71.2
	OURS	**84.1**$_{\text{W3.5A3.4}}$	**83.3**$_{\text{W3.5A3.2}}$	66.8$_{\text{W3.5A3.3}}$	**81.0**$_{\text{W3.5A3.3}}$	**74.8**$_{\text{W3.5A3.3}}$
INTERNVL2 8B	BASELINE	96.2	77.7	60.7	76.5	71.6
	MBQ	95.8	75.1	59.7	**72.8**	71.2
	MQUANT	91.8	73.0	61.4	70.7	70.8
	OURS	**95.9**$_{\text{W3.6A4.7}}$	**75.5**$_{\text{W3.6A4.6}}$	**61.9**$_{\text{W3.6A4.6}}$	72.7$_{\text{W3.6A4.7}}$	**71.2**$_{\text{W3.6A4.6}}$
LLAVA-OV 7B	BASELINE	90.0	76.1	60.4	62.2	72.0
	MBQ	88.2	65.8	58.1	50.1	67.3
	MQUANT	75.3	64.6	58.5	48.6	66.2
	OURS	**89.3**$_{\text{W3.5A4.9}}$	**73.0**$_{\text{W3.5A4.6}}$	**58.9**$_{\text{W3.5A4.5}}$	**60.0**$_{\text{W3.5A4.5}}$	**71.3**$_{\text{W3.5A4.6}}$

MBQ by 0.9% and 3.2% and MQuant by 3.7% and 3.1%, respectively, despite compressing to effective bitwidths of approximately W3.5A3.4 and W3.5A3.2. On OCRBench and SEED, we similarly surpass both MBQ and MQuant by notable margins (81.0 vs. 78.0/77.0 and 74.8 vs. 71.2/71.2), demonstrating that our compression does not disproportionately degrade visually grounded reasoning tasks.

On InternVL2-8B, our method achieves competitive or superior results across all benchmarks. Notably, on VizWiz, our method outperforms MBQ by 2.2% (61.9 vs. 59.7) and MQuant by 0.5% (61.9 vs. 61.4), and on TextVQA, we surpass MQuant by 2.5% (75.5 vs. 73.0). These results indicate that our approach maintains accuracy robustness even under aggressive compression, with effective bitwidths reaching as low as W3.6A4.6.

The contrast is most pronounced on LLaVA-OV-7B, where both baselines degrade sharply under W6A6: MBQ drops to 65.8 on TextVQA and 50.1 on OCRBench, and MQuant falls to 75.3 on ScienceQA and 48.6 on OCRBench. In contrast, our method preserves near-baseline accuracy across all five benchmarks (e.g., 89.3 on ScienceQA and 73.0 on TextVQA), with an average degradation of only 1.6% from BF16, compared to 6.2% for MBQ and 9.5% for MQuant. Taken together, our method outperforms MQuant on all 15 model–benchmark pairs and MBQ on 12 of 15 (with the remaining gaps within 0.1%), attaining near-baseline performance at substantially reduced effective bitwidths compared to existing W6A6 quantization baselines.

B.3 Evaluation of Comp^{2}VLM Across Various Precision Settings

Table 8 shows the performance of *Comp^{2}VLM* under different initial quantization configurations, including W7A7, W6A6, and W5A5, across multiple VLMs and benchmarks. The subscripts indicate the effective bit-width of weights and activations after applying DEQ and QPress. These results illustrate how accuracy and effective precision vary with the initial quantization setting.

At higher initial precision, W7A7 preserves accuracy close to the baseline across all evaluated models. For example, on Qwen2-VL-7B, W7A7 achieves 84.9% on ScienceQA and 84.0% on TextVQA, compared to baseline scores of 84.7% and 84.4%, while reducing the effective precision to approximately W4.5A$3.9 \sim 4.3$. Similar behavior is observed for LLaVA-OV-7B and InternVL2-8B, where performance differences from the baseline remain within a small margin on most benchmarks.

As the initial precision is reduced, W6A6 results in a larger decrease in effective bit-width, with moderate accuracy degradation. Across models, W6A6 achieves effective precisions of approximately W$3.5 \sim 3.6$A$3.2 \sim 4.9$. For instance, on InternVL2-8B, W6A6 attains 95.9% on ScienceQA and 61.9% on VizWiz, compared to 96.3% and 61.5% under W7A7. In contrast, W5A5 further reduces the effective precision to approximately W$2.6 \sim 2.9$A$2.9 \sim 3.8$, but incurs more noticeable performance drops on several benchmarks. This trend is particularly evident on text-centric, alignment-sensitive tasks such as TextVQA and OCRBench, where lower activation precision results in greater accuracy degradation.

Table 8: Comparing baseline and Comp^{2}VLM across various initial quantization configurations. Subscripts denote the effective bit-width (W: Weight, A: Activation) after compression.

MODEL	SETTING	SCIENCEQA	TEXTVQA	VIZWIZ	OCRBENCH	SEED
QWEN2-VL 7B	BF16	84.7	84.4	68.8	84.2	76.4
	W7A7	$84.9_{W4.5A4.3}$	$84.0_{W4.5A3.9}$	$67.4_{W4.5A3.9}$	$85.5_{W4.5A4.0}$	$75.5_{W4.5A3.9}$
	W6A6	$84.1_{W3.5A3.4}$	$83.3_{W3.5A3.2}$	$66.8_{W3.5A3.3}$	$81.0_{W3.5A3.3}$	$74.8_{W3.5A3.3}$
	W5A5	$79.0_{W2.6A2.9}$	$75.2_{W2.6A2.9}$	$63.3_{W2.6A2.9}$	$67.2_{W2.6A2.9}$	$70.6_{W2.6A2.9}$
LLAVA-OV 7B	BF16	90.0	76.1	60.4	62.2	72.0
	W7A7	$90.0_{W4.5A6.1}$	$75.9_{W4.5A5.8}$	$59.0_{W4.5A5.8}$	$62.0_{W4.5A5.8}$	$72.0_{W4.5A5.9}$
	W6A6	$89.3_{W3.5A4.9}$	$73.0_{W3.5A4.6}$	$58.9_{W3.5A4.5}$	$60.0_{W3.5A4.5}$	$71.3_{W3.5A4.6}$
	W5A5	$85.6_{W2.9A3.8}$	$61.4_{W2.9A3.5}$	$56.1_{W2.9A3.5}$	$56.1_{W2.9A3.5}$	$46.0_{W2.9A3.6}$
INTERNVL2 8B	BF16	96.2	77.7	60.7	76.5	71.6
	W7A7	$96.3_{W4.6A5.6}$	$76.9_{W4.6A5.5}$	$61.5_{W4.6A5.6}$	$76.3_{W4.6A5.6}$	$71.4_{W4.6A5.6}$
	W6A6	$95.9_{W3.6A4.7}$	$75.5_{W3.6A4.6}$	$61.9_{W3.6A4.6}$	$72.7_{W3.6A4.7}$	$71.2_{W3.6A4.6}$
	W5A5	$94.8_{W2.9A3.8}$	$69.0_{W2.9A3.6}$	$59.1_{W2.9A3.6}$	$62.3_{W2.9A3.7}$	$69.8_{W2.9A3.6}$

C Limitation

Comp^{2}VLM achieves effective compression in the 5–7-bit regime, but its efficiency diminishes as the bit-width decreases. In sub-4-bit settings, the representational capacity of quantized weights is already saturated, and the entropy-reduction behavior of DEQ amplifies this constraint. Specifically, DEQ encourages symbol duplication by collapsing weights into fewer quantization bins, which enables high compatibility with lossless compression and achieves higher compression ratios than quantization alone at moderate precision levels (5–7-bit) while maintaining competitive accuracy. However, when the quantization bin space is already extremely limited, as in 4-bit settings, the duplication-inducing behavior of DEQ causes non-negligible accuracy degradation. Consequently, aggressive low-bit quantization leads to substantial accuracy drops, and the entropy of the quantized distribution is already low enough that lossless compression provides only marginal additional gains. The complementary benefits of jointly optimizing quantization and lossless compression therefore become less pronounced in ultra-low-bit regimes, and *Comp^{2}VLM* is not designed for sub-4-bit quantization scenarios.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims in the abstract and Section 1 are supported by the experimental results in Tables 1–4.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations of this work are discussed in Section C.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: This paper does not include theoretical results; the equations in Section 3 define the proposed algorithms operationally.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The DEQ and QPress algorithms (Sections 3.2–3.3, Algorithm 1), all hyperparameters, and the experimental setup (models, benchmarks, runtime) are fully described in Section 4.1.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We submit our source code as supplementary material.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All hyperparameters and the calibration/evaluation protocol are specified in Sections 3.2–3.3 and Section 4.1. As this is a post-training quantization method, no training optimizer is involved.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Standard VLM benchmarks (TextVQA, OCRBench, ScienceQA, VizWiz, SEED, etc.) produce deterministic outputs under fixed greedy decoding, and PTQ involves no SGD-based stochasticity, leaving 5% calibration sampling as the only source of variability. Multi-seed evaluation across five VLMs (up to 72B parameters) and nine benchmarks was computationally prohibitive on our H100 setup, consistent with common practice in recent PTQ research.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: All experiments use a single NVIDIA H100 GPU with PyTorch and Triton (CUDA 12.8), as stated in Section 4.1.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The authors have reviewed the NeurIPS Code of Ethics and confirm that this work conforms in every respect.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [No]

    Justification: As a foundational model-compression technique, this work does not introduce societal impacts beyond those of the underlying VLMs, and a dedicated broader-impacts discussion is not included in this version.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This work proposes a compression algorithm and does not release any new pretrained models, datasets, or scraped data with a high risk of misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [No]

    Justification: All used models, benchmarks, and software libraries are properly cited with their original publications in Section 4.1, but a consolidated license summary is not included in this version.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

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

13. **New assets**

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

    Answer: [N/A]

    Justification: No new datasets or pretrained models are released. The source code will be released with documentation upon paper acceptance.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve human subjects research. IRB approval is not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: VLMs in this work are the target of compression, not a methodological tool used in developing the core method.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.