Bi-VLM: Binary Post-Training Quantization for Vision-Language Models

Xijun Wang*, Rayyan Abdalla*, Junyun Huang*, Chengyuan Zhang, Ruiqi Xian, Dinesh Manocha

University of Maryland, College Park, USA
{xijun, rsabdall, jhuang34, czhang35, rxian, dmanocha}@umd.edu

Abstract

We address the critical gap between the computational demands of vision-language models and the possible ultra-low-bit weight precision (bitwidth $\leq$2 bits) we can use for higher efficiency. Our work is motivated by the substantial computational cost and memory requirements of VLMs, which restrict their applicability in hardware-constrained environments. We propose Bi-VLM, which separates model weights non-uniformly based on the Gaussian quantiles. Our formulation groups the model weights into outlier and multiple inlier subsets, ensuring that each subset contains a proportion of weights corresponding to its quantile in the distribution. We propose a saliency-aware hybrid quantization algorithm and use it to quantize weights by imposing different constraints on the scaler and binary matrices based on the saliency metric and compression objective. We have evaluated our approach on different VLMs. For the language model part of the VLM, our Bi-VLM outperforms the SOTA by 3%-47% on the visual question answering task in terms of four different benchmarks and three different models. For the overall VLM, our Bi-VLM outperforms the SOTA by 4%-45%.

Introduction

In recent years, vision-language models (VLMs) have emerged as powerful tools for performing multi-modal tasks by combining visual and textual information. The pretrained VLMs, LLaVA (Liu et al. 2024b) and Qwen (Bai et al. 2025), have demonstrated remarkable performance on a wide range of benchmarks, including single-image analysis, multi-image interpretation, and video understanding. These achievements are achieved by their vast model parameters and extensive training data, enabling high accuracy and generalization across diverse tasks. However, they have high computational demands and high memory usage, which pose significant challenges for deployment on resource-constrained devices such as wearables, mobile devices, and FPGAs (Zeng et al. 2024).

Post-training quantization (PTQ) (Nagel et al. 2019, 2020; Krishnamoorthi 2018) has gained significant traction for VLMs due to its efficiency and practicality. Unlike quantization-aware training (QAT) (Jacob et al. 2017; Gupta et al. 2015), which requires access to the training datasets, PTQ operates on frozen network parameters and uses a small calibration set to determine optimal rounding functions. Recently, PTQ methods (Frantar et al. 2022; Lin et al. 2024) have achieved promising results by reducing the bitwidths for weights and activations.

Despite notable advances in 8-bit and 4-bit quantization for VLMs (Dettmers et al. 2023; Xiao et al. 2023; Wang et al. 2024), the increasing size and complexity of these models require more aggressive quantization strategies. Neural network binarization, which reduces the bit width of the weight to a single bit, is a promising direction to achieve ultra-low bit quantization (Helwegen et al. 2019; Qin et al. 2020, 2023). However, existing PTQ methods may not work in terms of ultra-low-bit quantization ($\leq$2 bits), leading to substantial performance degradation. State-of-the-art binary PTQ approaches, such as PB-LLM (Shang et al. 2023), provide limited performance levels with significant trade-offs in accuracy. We need better ultra-low-bit quantization methods for VLMs that can preserve the task performance.

Main Results: We present a novel approach that uses ultra-low-bit precision for VLMs. Our empirical analysis reveals that most weight values exhibit a near-Gaussian distribution. Furthermore, outlier density varies across layers, necessitating an adaptive quantization approach that assigns higher precision to critical weights while aggressively binarizing most of the model to maximize compression efficiency. Additionally, outliers, constitute about 5% of Vision Model weights and 1% of Language Model weights have a substantial impact on the model's performance. As a result, any uniform quantization strategy may not work well.

We design an efficient quantization strategy for VLMs that analyzes key weight distribution properties, including saliency, sparsity, and outliers. In particular, we present a saliency-aware hybrid quantization algorithm, where salient weights receive higher bit precision and unsalient weights are binarized, ensuring a balance between storage efficiency and quantization error minimization. Our approach, Bi-VLM, separates model weights non-uniformly based on Gaussian quantiles. Model weights are grouped into outlier (salient) and multiple inlier (unsalient) subsets, ensuring each subset contains a proportion of weights corresponding to its respective quantile in the distribution. We use our

*These authors contributed equally.

saliency-aware hybrid quantization algorithm to quantize the weights, where we solve the quantization problem by imposing different constraints on the scaler and binary matrices based on the saliency metric and compression objective.

To the best of our knowledge, we are the first work to explore the ultra-low-bit post training quantization ($\leq$2 bits) for VLMs. We also present strategies for achieving aggressive bit-width reduction while maintaining robust performance across diverse benchmarks. Some of the key contributions of our work include:

1. We propose Bi-VLM which separates weights non-uniformly based on Gaussian quantiles and then use our proposed saliency-aware hybrid quantization to quantize the weights.
2. We push post-training quantization to bit-level for large VLMs in terms of four different benchmarks and three different model series. From our experiments, our Bi-VLM outperforms the SOTA by 3%-47% on the language model part of the VLM and 4%-45% for the overall VLMs.

Related Works

Post-Quantization on VLM

Post-training quantization (PTQ) has become a practical solution for efficiently deploying Vision-Language Models (VLMs), as it replaces full-precision tensors with low-precision values, significantly reducing storage requirements and computational overhead. Unlike quantization-aware training (QAT), which demands costly retraining and access to the training dataset, PTQ methods leverage small calibration sets to optimize rounding functions with reduced data requirements and computational costs.

GPTQ (Frantar et al. 2022) employs Hessian-based second-order error compensation to minimize block-wise quantization errors, achieving excellent performance at ultra-low bit-widths (e.g., 4-bit quantization). Techniques such as AWQ (Lin et al. 2024) and OWQ (Lee et al. 2023) further enhance PTQ by preserving critical weight channels and scaling them appropriately for activation features, thereby retaining information representation capacity. Additionally, methods like PB-LLM(Shang et al. 2023) and SpQR (Dettmers et al. 2023) adopt feature segmentation strategies, selectively quantizing non-critical components to mitigate performance loss while maintaining low bit-width. QLoRA (Dettmers et al. 2024) introduces a memory-efficient fine-tuning method by backpropagating through frozen 4-bit quantized models using Low Rank Adapters (LoRA) and proposes innovations like 4-bit NormalFloat (NF4), double quantization, and paged optimizers to minimize memory usage without sacrificing performance. Q-VLM (Wang et al. 2024) proposes a post-training quantization approach for large vision-language models (LVLMs) that optimizes cross-layer dependency using activation entropy as a proxy, enabling block-wise partitioning to reduce discretization errors and search costs while maintaining performance efficiency.

While these methods demonstrate success in efficiently quantizing VLMs, they often rely on layer-wise or block-wise optimization, which overlooks dependencies across layers or components of the model. Addressing these cross-layer interactions remains a key challenge for achieving optimal quantization performance in large-scale VLMs.

Network Binarization

Binarization reduces neural network parameters to a single bit, significantly decreasing storage requirements and computational costs. This is achieved by converting full-precision weights into binary values using the sign function, combined with a scaling factor to minimize binarization errors. Typically, the scaling factor is applied in a channel-wise manner to better preserve information (Rastegari et al. 2016; Qin et al. 2023).

Most existing binarization methods rely on quantization-aware training (QAT), where the training process accounts for quantization effects. Straight Through Estimator (STE) (Bengio, Léonard, and Courville 2013) is often used to overcome gradient vanishing issues caused by the non-differentiable nature of the binarization function. Binary Weight Network (BWN) (Rastegari et al. 2016) was among the first to demonstrate binarized weights while maintaining full-precision activations, whereas XNOR-Net (Rastegari et al. 2016) extended this approach by binarizing both weights and activations for higher efficiency. Methods like DoReFa-Net (Zhou et al. 2016) further improved the training speed by introducing quantized gradients.

Recent advancements have explored group-wise binarization strategies, where network weights are divided into smaller groups to minimize binarization errors (Faraone et al. 2018). Notably, binarization techniques have been successfully applied to Transformers (Wang et al. 2023) and BERT models (Qin et al. 2022), demonstrating the feasibility of deploying binarized networks in real-world scenarios.

For large language models (LLMs), PB-LLM (Shang et al. 2023) investigates the use of binarized QAT and post-training quantization (PTQ) strategies. However, it reveals the challenge of retaining a significant portion of weights, typically over 30%, at INT8 precision to maintain acceptable performance. BiLLM (Huang et al. 2024) aims to push the boundaries of PTQ-based binarization for LLMs. By minimizing reliance on higher bit-widths while ensuring performance, these methods make significant strides toward fully binarized LLMs, offering a promising solution for resource-efficient deployment in large-scale applications.

Our Approach

Statistical Analysis of Weights: Histograms and Gaussian Fit

Understanding weight matrix statistics is essential for designing an efficient group-wise post-training quantization strategy. We analyze weight distributions through histograms, validating their tendency toward a Gaussian distribution. This supports our Gaussian assumption and highlights the effectiveness of a quantile-based approach for partitioning weights into consistent subsets. Since weight values vary significantly, saliency analysis is crucial for identifying critical weights. Our findings, observed across

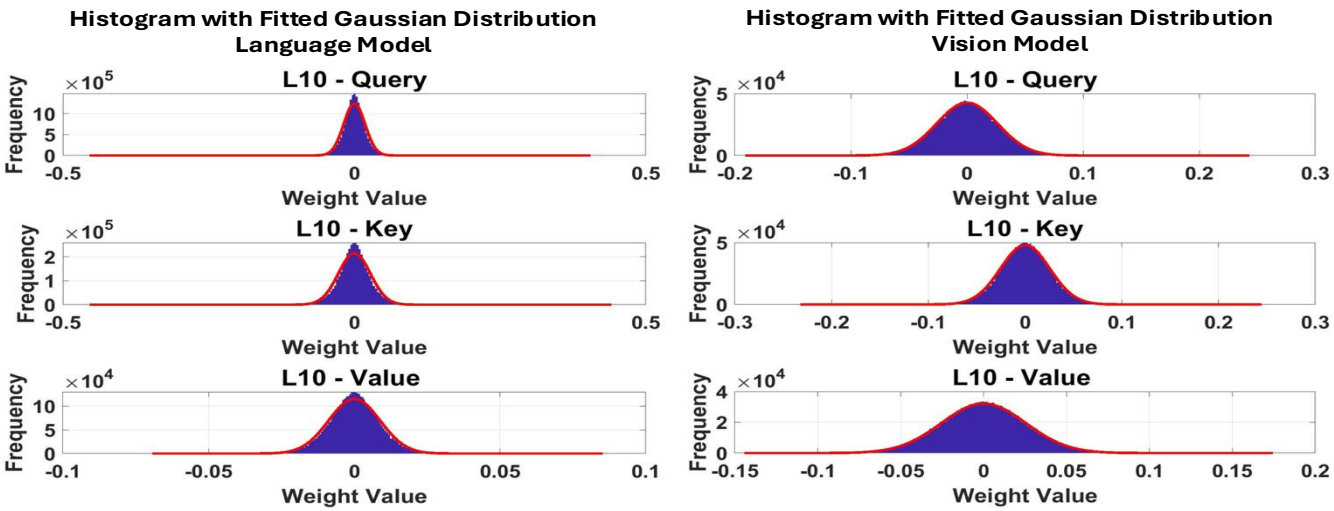

Figure 1: Histogram Analysis and Fitted Gaussian Curve of Layer 10 (other layers share the same pattern) of the Language/Vision Model (Llama 3.2-Vision). The curve represents the fitted Gaussian distribution over the histogram bar plot.

both Vision and Language Models, inform our post-training quantization approach.

Key Observations We analyze the element-wise weight histograms of both Vision and Language Models, using varying bin sizes based on layer dimensions. Our focus is on the key, query, and value projection layers in self-attention, as well as the final output layer. To approximate the weight distribution, we fit the closest Gaussian curve using the sample mean and variance. As shown in Figures 1, for both Vision and Language models, most weights exhibit a Gaussian-like distribution with a near-zero mean, consistent with prior findings on deep learning architectures and LLMs(Fang et al. 2020; Yu et al. 2020; Cholakov, Guo, and Kim 2023). This suggests that weights remain close to their initial values, reinforcing the assumption that large-scale models preserve their initialized weight structure. Therefore, we adopt Gaussian assumption on layer weights distribution for both Vision and Language models.

Binarization Formulation

Network binarization is a quantization technique that restricts weight values to discrete binary levels, typically $\{-1, 1\}$. The purpose is to reduce memory footprint and computational complexity while maintaining accuracy. To enhance expressiveness, it is often combined with low-bit quantization, where more sensitive weights are assigned higher precision.

We propose a generalized formulation of the weight quantization process that unifies both binarization and low-bit quantization as a discrete optimization problem, aiming to find the best binary representation of weights while introducing a row-wise scaling factor to preserve reconstruction accuracy. Given a weight matrix $W \in \mathbb{R}^{m\times n}$, we minimize the reconstruction error:

$$\|W - AB\|_F^2, \tag{1}$$

where $A \in \mathbb{R}^{m\times m}$ is a diagonal scaling matrix, defined as $A = \text{diag}(a_{11}, a_{22}, ..., a_{mm})$, ensuring row-wise scaling. The binary matrix $B \in \mathbb{R}^{m\times n}$ contains discrete elements b_{ij} constrained to take N_b discrete values within the range $[-1, 1]$. For storage purposes, each b_{ij} is later mapped to a set of quantized values $\{0, 1, ..., 2^{N_b} - 1\}$, where N_b represents the number of bits assigned per weight element.

The objective function in Equation (1) is minimized with respect to the Frobenius norm $\|\cdot\|_F$, which reduces the element-wise squared differences between W and AB, providing an optimal approximation by minimizing residual energy. This formulation is particularly effective when outliers are not dominant, as confirmed by empirical analysis. For $N_b = 1$, the constraints in Equation (1) reduce to $b_{ij} \in \{-1, 1\}$, corresponding to the standard binarization process.

Bi-VLM

Quantile-Based Weight Partitioning A major challenge in quantizing large models, such as LLMs and VLMs, is identifying salient weights—those whose quantization may significantly impact performance. Saliency is typically assessed using either magnitude or the Hessian-based metric, though the latter is computationally expensive with limited benefits (Shang et al. 2023).

We use a magnitude-based criterion to classify model weights within each layer into distinct subsets. Instead of linear partitioning, we divide the weights non-uniformly based on Gaussian quantiles. Specifically, weights are grouped into outlier (salient) and multiple inlier (unsalient) subsets, ensuring each subset contains a proportion of weights corresponding to its respective quantile in the distribution.

Given a model with N_l layers, where each layer is indexed by $l = 1, 2, \ldots, N_l$, we analyze the weight values of each layer l and define the set of all weights as $\mathcal{W}_l$. We partition

$\mathcal{W}_l$ into salient weights $\mathcal{S}_l$ and unsalient weights $\mathcal{S}_l^c$, such that:

$$\mathcal{W}_l = \mathcal{S}_l \cup \mathcal{S}_l^c, \quad \mathcal{S}_l \cap \mathcal{S}_l^c = \varnothing. \tag{2}$$

Furthermore, we divide the unsalient set $\mathcal{S}_l^c$ into N_{uns} disjoint subsets $\mathcal{S}_l^{c(k)}$, indexed by $k = 1, \ldots, N_{\text{uns}}$, such that:

$$\mathcal{S}_l^c = \bigcup_{k=1}^{N_{\text{uns}}} \mathcal{S}_l^{c(k)}. \tag{3}$$

Our classification, guided by Gaussian quantiles, partitions the weights into $\mathcal{S}_l^c$ and $\mathcal{S}_l^{c(k)}$ for $k = 1, \ldots, N_{\text{uns}}$. This partitioning is based on the layer-specific mean μ_l and standard deviation σ_l and assumes, based on empirical evidence, that the weight distribution in each layer is symmetric about its mean. Consequently, quantile analysis is performed on the right side of the distribution and mirrored to the left.

We define the percentile p_{l}^{sal} as the proportion of the highest-magnitude weights classified as salient. This corresponds to the upper quantile of the Gaussian distribution. Consequently, the total quantile of the unsalient region is $1 - p_{\text{l}}^{sal}$. Since this region is divided into N_{uns} subsets, each subset corresponds to a quantile of size:

$$p_l^{uns} = \frac{1 - p_l^{sal}}{N_{\text{uns}}}, \tag{4}$$

In order to determine the magnitude cutoff values for salient and unsalient subsets, we compute the z-scores from the standard normal distribution corresponding to the respective quantiles. These are given by:

$$z^{(k)} = \Phi^{-1}\left(\frac{1 + k \cdot p_l^{\text{uns}}}{2}\right), \quad k = 1, 2, \ldots, N_{\text{uns}}, \tag{5}$$

where $\Phi^{-1}(\cdot)$ is the *probit function*, i.e., the inverse cumulative distribution function (CDF) of the Gaussian distribution. Therefore, the salient and non-salient subsets for layer l are defined as:

$$\begin{aligned} \mathcal{S}_l &= \{w \in \mathcal{W}_l \mid |w| > \mu_l + \sigma_l z^{(N_{\text{uns}})}\} \\ \mathcal{S}_l^{c(k)} &= \{w \in \mathcal{W}_l \mid \mu_l + \sigma_l z^{(k-1)} < |w| \leq \mu_l + \sigma_l z^{(k)}\}, \end{aligned} \tag{6}$$

where μ_l, σ_l are the mean and standard deviation of the weight distribution in layer l, and z^k is the z-score corresponding to the upper magnitude boundary of the non-salient region indexed $k = 1, 2, \ldots, N_{\text{uns}}$. Overall, our approach enables independent analysis of layer weights while parameterizing the proportion of salient weights and the number of non-salient partitions to optimize the accuracy-compression tradeoff.

Saliency-Aware Hybrid Quantization We present a hybrid quantization method applied independently to each model layer. The weight matrix W_l of layer l is decomposed into a salient weight matrix $W_l^{\text{sal}} \in \mathbb{R}^{m \times n}$ and multiple unsalient weight component matrices $W_l^{\text{uns}(k)} \in \mathbb{R}^{m \times n}$ for $k = 1, 2, \ldots, N_{\text{uns}}$, enabling distinct processing strategies for quantization.

Algorithm 1: Row-wise Quantization with Scale and Binary Matrix

Require: Weight matrix $W \in \mathbb{R}^{m \times n}$, iterations iters, bits N_b
Ensure: Scale factors $a \in \mathbb{R}^m$, quantized matrix $B \in \mathbb{R}^{m \times n}$
Initialize: $B \leftarrow \text{sign}(W)$, $a \leftarrow \mathbf{0}$
for iter $= 1$ to iters **do**
  $a_{\text{old}} \leftarrow a$
  $a \leftarrow \frac{\sum_j W_{ij} B_{ij}}{\sum_j B_{ij}^2}$ where $\sum_j B_{ij}^2 > 0$, else $a_i = 0$
  $B \leftarrow \text{clip}\left(\frac{W}{a[:, None]}, -1, 1\right)$
end for
Adaptive Quantization Step:
Compute μ_B, σ_B from nonzero elements of B
Compute quantization levels using (14)
$r_{\text{centers}} \leftarrow \frac{r_{\text{levels}}[:-1] + r_{\text{levels}}[1:]}{2}$
$B \leftarrow r_{\text{centers}}[\arg\min |B - r_{\text{centers}}|]$
return a, B

$$W_l = [w_{ij}]_{1 \leq i \leq m, 1 \leq j \leq n} = W_l^{\text{sal}} + \sum_{k=1}^{N_{\text{uns}}} W_l^{\text{uns}(k)}, \tag{7}$$

where W_l^{sal} and $W_l^{\text{uns}(k)}$ are defined as:

$$W_l^{\text{sal}} = \{w_{l,ij}^{\text{sal}} \mid w_{l,ij} = w_{ij} \text{ if } w_{ij} \in \mathcal{S}, \text{ else } 0\}, \tag{8}$$

$$W_l^{\text{uns}(k)} = \{w_{l,ij}^{\text{uns}(k)} \mid w_{l,ij}^{(k)} = w_{ij} \text{ if } w_{ij} \in \mathcal{S}_l^{c(k)}, \text{ else } 0\}. \tag{9}$$

In order to optimize compression while preserving accuracy, we solve the quantization problem in Equation (1) separately for the salient and unsalient components, imposing different constraints on the matrices A and B based on the saliency metric and compression objective. Specifically, salient weights, which are highly sensitive and impact the model's performance, are quantized using two bits, whereas unsalient weights within the inlier subsets are binarized using a single bit.

Salient Weights Two-Bit Quantization: The salient weight matrix W_l^{sal} (Equation 8) is quantized into a binary matrix B_l^{sal} using $N_b = 2$ bits, along with a row-wise scaling vector $\mathbf{a} \in \mathbb{R}^m$, yielding:

$$W_l^{\text{sal,quantized}} = \mathbf{a} \odot B_l^{\text{sal}}. \tag{10}$$

Here, $\odot$ denotes row-wise multiplication. The optimization problem (Equation 1) is solved with $B = B_l^{\text{sal}}$, where elements take $2^{N_{bit}}$ discrete values in $[-1, 1]$ and are mapped to the set $\{0, 1, 2, 3\}$, binary represented for storage. The scaling matrix is:

$$A_l^{\text{sal}} = \text{diag}(a_1^{\text{sal}}, \ldots, a_m^{\text{sal}}). \tag{11}$$

Thus, the quantization problem is formulated as:

$$\|W_l^{\text{sal}} - \mathbf{a} \odot B_l^{\text{sal}}\|_F^2. \tag{12}$$

We analyze the convexity of Equation (12) by reformulating it as a row-wise optimization:

$$\|w_{l,ij}^{\text{sal}} - a_i^{\text{sal}} b_{l,ij}^{\text{sal}}\|_2^2, \quad \forall\, 1 \le j \le n. \tag{13}$$

Since the ℓ_2-norm is convex, the optimization formulation is convex with respect to the scaling parameter a_i^{sal}. However, the discrete nature of B_l^{sal}, constrained to 2^{N_b} values, results in a discrete optimization problem where gradient-based solvers fail due to the discontinuous solution space. To address this, we relax $b_{l,ij}^{\text{sal}}$ to the continuous interval $[-1, 1]$, enabling an iterative quadratic programming approach that alternates between optimizing a_i^{sal} and updating B_l^{sal}, ensuring a feasible quantization solution. After convergence, B_l^{sal} is mapped back to a discrete set of 2^{N_b} values, corresponding to the midpoints of $2^{N_b} + 1$ quantization levels, as described in Algorithm 1.

Our empirical analysis reveals that outlier weights contribute unevenly to the tails of the Gaussian distribution, necessitating adaptive quantization resolution. To address this, we apply an exponential adaptation to the $2^{N_b} + 1$ linearly spaced levels $d \in [-1, 1]$ after convex optimization:

$$r_{\text{levels}}(d) = \mu_{B_l^{\text{sal}}}^{\text{optimal}} + \sigma_{B_l^{\text{sal}}}^{\text{optimal}} \cdot \text{sign}(d) \cdot (\alpha \times \exp(|d|) - 1), \tag{14}$$

where r_{levels} is the adaptively defined quantization level corresponding to the linearly defined level d. Quantization centers, r_{centers}, representing the midpoints between each subsequent level, encode mapped values from B_l^{sal} as a binary value of length $N_b = 2$. The parameters $\mu_{B_l^{\text{sal}}}^{\text{optimal}}$ and $\sigma_{B_l^{\text{sal}}}^{\text{optimal}}$ denote the mean and standard deviation of the optimized B_l^{sal}, respectively. The term $\text{sign}(d)$ ensures symmetric quantization. The parameter α, empirically set to 1.4, adjusts the mapping to provide finer resolution for small weights and coarser representation for larger values.

Algorithm 2: Unsalient Weights Binarization

Require: Weight matrix $W_l^{\text{uns}(k)} \in \mathbb{R}^{m \times n}$

Ensure: Binary matrix $B_l^{\text{uns}(k)} \in \{-1, 1\}^{m \times n}$ and scaling factor $a_l^{\text{uns}(k)} \in \mathbb{R}$

1: **Step 1: Compute Binary Matrix**
2: Compute $B_l^{\text{uns}(k)}$ using Equation (19)
3: **Step 2: Compute Scaling Factor**
4: Compute $a_l^{\text{uns}(k)}$ using Equation (18)
5: **return** $B_l^{\text{uns}(k)}, a_l^{\text{uns}(k)}$

Unsalient Weights Binarization: With salient weights quantized and separated, the remaining components of the weight matrix in layer l correspond to the unsalient weights. We apply strict binarization by mapping each matrix $W_l^{\text{uns}(k)}$ to a binary matrix $B_l^{\text{uns}(k)}$ and a scalar factor $a_l^{\text{uns}(k)}$, such that:

$$W_l^{\text{uns,quantized}(k)} = a_l^{\text{uns}(k)} \times B_l^{\text{uns}(k)}. \tag{15}$$

Reformulating the optimization problem in (1), we set $B = B_l^{\text{uns}(k)}$ and $A = a_l^{\text{uns}(k)} I_{m \times m}$, where $B_l^{\text{uns}(k)} \in \{-1, 1\}$. The resulting binarization problem can be expressed as:

$$\|W_l^{\text{uns}(k)} - a_l^{\text{uns}(k)} B_l^{\text{uns}(k)}\|_F^2. \tag{16}$$

Since $b_{l,ij}^{\text{uns}(k)}$ is restricted to $\{-1, 1\}$, an element-wise thresholding strategy minimizes the squared error, yielding the optimal binary values:

$$b_{l,ij}^{\text{uns}(k)*} = \begin{cases} \text{sign}(w_{l,ij}^{\text{uns}(k)}) & \text{if } a_l^{\text{uns}(k)} > 0, \\ -\text{sign}(w_{l,ij}^{\text{uns}(k)}) & \text{if } a_l^{\text{uns}(k)} < 0. \end{cases} \tag{17}$$

With $B_l^{\text{uns}(k)}$ now predetermined, a solution to the convex optimization formulation of the quadratic program in (12) with respect to $a_l^{\text{uns}(k)}$ gives an optimal solution:

$$a_{l,ij}^{\text{uns}(k)*} = \frac{\langle W_l^{\text{uns}(k)}, B_l^{\text{uns}(k)} \rangle}{\|B_l^{\text{uns}(k)}\|_F^2}, \tag{18}$$

where the inner product is $\langle W_l^{\text{uns}(k)}, B_l^{\text{uns}(k)} \rangle = \text{tr}\left({W_l^{\text{uns}(k)}}^\top B_l^{\text{uns}(k)} \right)$. This guarantees that $a_{l,ij}^{\text{uns}(k)}$ is always positive, ensuring optimal binary matrix:

$$B_{l,ij}^{\text{uns}(k)*} = \text{sign}(W_{l,ij}^{\text{uns}(k)}). \tag{19}$$

Algorithm 3: Our Overall Method

Require: $W_l \in \mathbb{R}^{m \times n}$, N_{uns}, N_b, $p_l^{\text{sal,max}}$

Ensure: $W_l^{\text{quantized}}$

1: **Step 1: Initialize** $p_l^{\text{sal}} = p_l^{\text{sal,max}}$
2: **Step 2: Optimize** Minimize $\mathcal{J}$: **Hybrid Quantization** $(W_l, N_{\text{uns}}, N_b, p_l^{\text{sal}})$ over $p_l^{\text{sal}} \in [0, p_l^{\text{sal,max}}]$, find $p_l^{\text{sal, opt}}$
3: **Step 3: Quantize** Run **Hybrid Quantization** $(W_l, N_{\text{uns}}, N_b, p_l^{\text{sal, opt}})$
4: **Step 4: Reconstruct**

$$W_l^{\text{quantized}} = W_l^{\text{sal,quantized}} + \sum_{k=1}^{N_{\text{uns}}} W_l^{\text{uns,quantized}(k)}$$

5: **return** $W_l^{\text{quantized}}$

Hybrid Quantization $(W_l, N_{\text{uns}}, N_b, p_l^{\text{sal}})$

6: Partition W_l into W_l^{sal} and $W_l^{\text{uns}(k)}$ (Eqs. (8), (9))
7: Quantize W_l^{sal} into $B_l^{\text{sal}}, \mathbf{a}$ (Algorithm 1)
8: **for** $k = 1$ to N_{uns} **do**
9: Quantize $W_l^{\text{uns}(k)}$ into $B_l^{\text{uns}(k)}, a_l^{\text{uns}(k)}$ (Algorithm 2)
10: **end for**
11: **return** $B_l^{\text{sal}}, \mathbf{a}, \{(a_l^{\text{uns}(k)}, B_l^{\text{uns}(k)})\}_{k=1}^{N_{\text{uns}}}$

Benchmark	FP	AWQ-L	BiLLM-L	Bi-VLM-L (Ours)	AWQ-all	BiLLM-all	Bi-VLM-all (Ours)
MME Perception	1446.81	0.00	1096.67	1315.84 (**219.17**↑)	-	781.51	1308.40 (**526.89**↑)
MME Cognition	341.42	0.00	248.21	171.43 (**76.78**↓)	-	222.85	196.07 (**26.78**↓)
ScienceQA-IMG	85.82	0.00	11.01	58.75 (**47.74**↑)	-	17.40	58.35 (**45.95**↑)
MMMU	42.78	24.33	25.56	36.42 (**10.86**↑)	-	24.56	33.27 (**8.71**↑)
VizWiz-VQA	59.72	0.00	35.36	39.33 (**3.97**↑)	-	22.93	47.18 (**24.25**↑)

Table 1: **SOTA comparison on Llama 3.2-Vision instruction 11B with weight 1 to 1.1 bit**. For the language model part, our Bi-VLM outperforms the SOTA by 4%-47%. For the overall VLM, our Bi-VLM outperforms the SOTA by 8%-45%. FP: Full precision. L: Language model, all: the whole VLM model.

Benchmark	FP	AWQ-L	BiLLM-L	Bi-VLM-L (Ours)	AWQ-all	BiLLM-all	Bi-VLM-all (Ours)
MME Perception	1578.39	0.00	1024.15	1457.68 (**433.53**↑)	-	813.89	1063.07 (**249.18**↑)
MME Cognition	418.21	0.00	150.36	340.71 (**190.35**↑)	-	178.57	272.21 (**93.64**↑)
ScienceQA-IMG	95.84	0.00	72.83	93.55 (**20.72**↑)	-	63.81	83.43 (**19.62**↑)
MMMU	49.56	25.33	31.11	44.33 (**13.22**↑)	-	28.78	39.12 (**10.34**↑)
VizWiz-VQA	60.38	0.00	56.91	60.10 (**3.19**↑)	-	52.81	57.36 (**4.55**↑)

Table 2: **SOTA comparison on Llava-One-Vision 7B with weight 1 to 1.1 bit**. For the language model part, our Bi-VLM outperforms the SOTA by 3%-20%. For the overall VLM, our Bi-VLM outperforms the SOTA by 4%-19%. FP: Full precision. L: Language model, all: the whole VLM model.

Adaptive Saliency Search via Optimization Salient weights deviate significantly from the typical weight distribution, making them unsuitable for binarization. Their inclusion in the unsalient subset $\mathcal{S}_l^c$ leads to rectification, degrading model performance (Shang et al. 2023), while selecting too many salient weights in $\mathcal{S}_l$ increases storage requirements, reducing compression efficiency.

In order to balance the performance and compression, we formulate salient weight selection as a numerical optimization problem that determines the optimal salient percentile p_l^{sal}. The optimized saliency region is constrained by a compression threshold set by the maximum proportion of salient weights quantized with N_b bits and the number of binarized unsalient quantiles N_{uns}. Given a specific layer l with weight matrix W_l, a predefined unsalient region partition count N_{uns}, and a maximum allowable salient percentile $p_l^{\text{sal,max}}$, we minimize the following objective function based on normalized reconstruction error:

$$\min_{p_l^{\text{sal}}} \mathcal{J}(p_l^{\text{sal}}; W_l, N_{\text{uns}}, N_b) = \tag{20}$$

$$\frac{\|W_l^{\text{sal}} - \mathbf{a} \odot B_l^{\text{sal}}\|_F^2 + \sum_{k=1}^{N_{\text{uns}}} \|W_l^{\text{uns}(k)} - a_l^{\text{uns}(k)} B_l^{\text{uns}(k)}\|_F^2}{\|W_l\|_F^2},$$

$$\text{s.t.} \quad p_l^{\text{sal}} \in [0, p_l^{\text{sal,max}}].$$

Here, W_l^{sal} and $W_l^{\text{uns}(k)}$ are obtained from Equations (8) and (9), respectively, while the pairs $\{\mathbf{a}, B_l^{\text{sal}}\}$ and $\{a_l^{\text{uns}(k)}, B_l^{\text{uns}(k)}\}$ are computed using Algorithms 1 and 2. Since the Frobenius norm $\|\cdot\|_F^2$ is convex and our hybrid quantization approach relaxes the discrete space of B_l^{sal} and $B_l^{\text{uns}(k)}$, the objective function $\mathcal{J}$ ensures a global minimum for given W_l, N_{uns}, and N_b. For computational efficiency, we employ bounded numerical optimization using Brent's method, a gradient-free approach that combines golden-section search for robustness and parabolic interpolation for fast convergence (Brent 2013). Once the optimal salient percentile $p_l^{\text{sal,opt}}$ is determined, we apply the full quantization process to layer W_l, as outlined in Algorithm 3.

Results and Comparisons

Baseline Model and Datasets

To demonstrate the effectiveness of our proposed method, we compare our algorithm with the state-of-the-art (SOTA) methods on 4 dataset benchmarks and 3 models. Datasets include: MME (Fu et al. 2024) that focuses on perception and cognition abilities; MMMU (Yue et al. 2024) that focuses on college-level subject knowledge and deliberate reasoning, Science Question Answering (ScienceQA) (Lu et al. 2022) that focuses on science topics, and VizWiz-VQA (Gurari et al. 2018) that on predicting the answer to a visual question and predict whether a visual question cannot be answered. Models include Llama 3.2-Vision instruction 11B, Llava-One-Vision 7B (Liu et al. 2024a; Li et al. 2024), and Qwen2.5-VL-7B-Instruct (Bai et al. 2025). For all the experiments, we use 64 samples for calibration. We use 2 bits for salient weights and 1 bit for non-salient weights. Salient weights account up to 5% of the total weights. Quantized weights storage and bitwidth calculation, saliency threshold determination, please refer to the Supplementary.

Benchmark	FP	AWQ-L	BiLLM-L	Bi-VLM-L (Ours)	AWQ-all	BiLLM-all	Bi-VLM-all (Ours)
MME Perception	1683.88	0.00	1204.31	1690.67 (**486.36**↑)	-	828.54	1458.70 (**630.16**↑)
MME Cognition	653.21	0.00	341.07	621.79 (**280.72**↑)	-	195.36	559.64 (**364.28**↑)
ScienceQA-IMG	77.29	0.00	62.62	68.32 (**5.70**↑)	-	59.49	64.01 (**4.52**↑)
MMMU	51.00	25.44	35.33	46.00 (**10.67**↑)	-	30.89	42.89 (**12.00**↑)
VizWiz-VQA	70.43	0.00	59.85	66.20 (**6.35**↑)	-	52.72	62.96 (**10.24**↑)

Table 3: **SOTA comparison on Qwen2.5-VL-7B-Instruct with weight 1.1 bit**. For the language model part, our Bi-VLM outperforms the SOTA by 5%-10%. For the overall VLM, our Bi-VLM outperforms the SOTA by 4%-12%. FP: Full precision. L: Language model, all: the whole VLM model.

Comparison with SOTA Methods

For Llama 3.2-Vision instruction 11B, Table 1 summarizes the quantization performance of our Bi-VLM method (under both language-only and whole-model quantization) compared to the state-of-the-art AWQ and BiLLM approaches. Notably, our Bi-VLM outperforms SOTA by 4%-47% on all settings. Although Bi-VLM shows somewhat lower performance on MME Cognition compared with BiLLM, the overall results confirm that Bi-VLM consistently preserves most of the full-precision accuracy across diverse benchmarks—frequently improving upon competing low-bit methods—while substantially reducing model size and computational overhead.

For Llava-One-Vision 7B (Liu et al. 2024a; Li et al. 2024), as shown in Table 2, our Bi-VLM outperforms SOTA by 3%-20% on all settings, these results underscore that our Bi-VLM quantization strategy substantially narrows the gap to full-precision performance—often beating previous low-bit methods by a large margin—across diverse vision-language tasks.

For Qwen2.5-VL-7B-Instruct (Bai et al. 2025), as shown in Table 3, our Bi-VLM outperforms SOTA by 4.5%-12% on all settings. Overall, these results demonstrate that our Bi-VLM quantization strategy consistently outperforms prior low-bit methods across diverse vision-language tasks based on aggressive bitwidth reduction.

Method	Performance	Average bit width	Quant Speed
BiLLM	17.4	1.11	139.8 Mins
Bi-VLM (Ours)	58.35	1.014	9.6 Mins

Table 4: More comparisons w.r.t. different aspects. With model Llama 3.2-Vision-instruction-11B on ScienceQA-IMG dataset. We get considerable improvements and our quantization speed is much more efficient than BiLLM.

Benchmark	Full precision	Vision	adaptor	Language
BiLLM	85.82	71.44	85.87	12.99
Bi-VLM (Ours)	85.82	85.23	85.86	58.75

Table 5: Llama 3.2-Vision 11B, 1 to 1.1 bit, on ScienceQA-IMG dataset. The language model has the highest sensitivity.

Components	B	B+HQ	B+HQ+AS	B+HQ+AS+MUS
Accuracy	6.33	13.74	49.51	58.35

Table 6: Ablation study on Bi-VLM Components on Llama 3.2-Vision 11B, ScienceQA-IMG. B: Fixed binary only. HQ: Hybrid precision quantization. AS: Adaptive saliency search. MUS: Multiple unsalient search.

Ablation Study

We conducted more comparisons across various methodological aspects using the LLaMA 3.2-Vision-Instruction-11B model on the ScienceQA-IMG dataset. Our proposed Bi-VLM framework demonstrates substantial performance improvements over BiLLM. Notably, our quantization process is significantly more efficient than that of BiLLM. As reported in Table 4, Bi-VLM achieves superior accuracy with lower bit precision and exhibits a 14.6× speedup in quantization.

To further investigate ablation studies on quantization sensitivity of different VLM components, we performed ablation studies on Vision model, Adaptor module, and Language model, as presented in Table 5. The results indicate moderate sensitivity to the vision encoder, minimal sensitivity to the adaptor/projector, and considerable sensitivity to the language model.

To explore the contributions of different components to quantization performance, we conducted ablations on the internal components of Bi-VLM, as shown in Table 6. Each module contributes meaningfully to the overall system performance, validating the design choices of our architecture.

Conclusions, Limitations, and Future Work

We present a novel ultra-low-bit quantization method for VLMs. Our approach successfully bridges the gap between the computational demands of vision-language models and the practical limitations of ultra-low-bit precision. Our results demonstrate that Bi-VLM outperforms the state-of-the-art methods on both language and overall vision-language models. Our approach has some limitations. We did not explore our method on large bitwidth, like 4 bits or 8 bits. It may be useful to explore combinations of different bitwidths. We would like to evaluate the performance on more tasks and in hardware-constrained environments.

References

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; Zhong, H.; Zhu, Y.; Yang, M.; Li, Z.; Wan, J.; Wang, P.; Ding, W.; Fu, Z.; Xu, Y.; Ye, J.; Zhang, X.; Xie, T.; Cheng, Z.; Zhang, H.; Yang, Z.; Xu, H.; and Lin, J. 2025. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*.

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Brent, R. P. 2013. *Algorithms for minimization without derivatives*. Courier Corporation.

Cholakov, R.; Guo, H.; and Kim, Y. 2023. Distributional Quantization of Large Language Models.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Dettmers, T.; Svirschevski, R.; Egiazarian, V.; Kuznedelev, D.; Frantar, E.; Ashkboos, S.; Borzunov, A.; Hoefler, T.; and Alistarh, D. 2023. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2306.03078*.

Fang, J.; Shafiee, A.; Abdel-Aziz, H.; Thorsley, D.; Georgiadis, G.; and Hassoun, J. H. 2020. Post-training piecewise linear quantization for deep neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 69–86. Springer.

Faraone, J.; Fraser, N.; Blott, M.; and Leong, P. H. 2018. Syq: Learning symmetric quantization for efficient deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4300–4309.

Frantar, E.; Ashkboos, S.; Hoefler, T.; and Alistarh, D. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

Fu, C.; Chen, P.; Shen, Y.; Qin, Y.; Zhang, M.; Lin, X.; Yang, J.; Zheng, X.; Li, K.; Sun, X.; Wu, Y.; and Ji, R. 2024. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. arXiv:2306.13394.

Gupta, S.; Agrawal, A.; Gopalakrishnan, K.; and Narayanan, P. 2015. Deep learning with limited numerical precision. In *International conference on machine learning*, 1737–1746. PMLR.

Gurari, D.; Li, Q.; Stangl, A. J.; Guo, A.; Lin, C.; Grauman, K.; Luo, J.; and Bigham, J. P. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3608–3617.

Helwegen, K.; Widdicombe, J.; Geiger, L.; Liu, Z.; Cheng, K.-T.; and Nusselder, R. 2019. Latent weights do not exist: Rethinking binarized neural network optimization. *Advances in neural information processing systems*, 32.

Huang, W.; Liu, Y.; Qin, H.; Li, Y.; Zhang, S.; Liu, X.; Magno, M.; and Qi, X. 2024. Billm: Pushing the limit of post-training quantization for llms. *arXiv preprint arXiv:2402.04291*.

Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A. G.; Adam, H.; and Kalenichenko, D. 2017. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2704–2713.

Krishnamoorthi, R. 2018. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*.

Lee, C.; Jin, J.; Kim, T.; Kim, H.; and Park, E. 2023. OWQ: Outlier-Aware Weight Quantization for Efficient Fine-Tuning and Inference of Large Language Models. In *AAAI Conference on Artificial Intelligence*.

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

Lin, J.; Tang, J.; Tang, H.; Yang, S.; Chen, W.-M.; Wang, W.-C.; Xiao, G.; Dang, X.; Gan, C.; and Han, S. 2024. AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration. *Proceedings of Machine Learning and Systems*, 6: 87–100.

Liu, H.; Li, C.; Li, Y.; Li, B.; Zhang, Y.; Shen, S.; and Lee, Y. J. 2024a. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2024b. Visual instruction tuning. *Advances in neural information processing systems*, 36.

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

Nagel, M.; Amjad, R. A.; Van Baalen, M.; Louizos, C.; and Blankevoort, T. 2020. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, 7197–7206. PMLR.

Nagel, M.; Baalen, M. v.; Blankevoort, T.; and Welling, M. 2019. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1325–1334.

Qin, H.; Ding, Y.; Zhang, M.; Yan, Q.; Liu, A.; Dang, Q.; Liu, Z.; and Liu, X. 2022. Bibert: Accurate fully binarized bert. *arXiv preprint arXiv:2203.06390*.

Qin, H.; Gong, R.; Liu, X.; Shen, M.; Wei, Z.; Yu, F.; and Song, J. 2020. Forward and backward information retention for accurate binary neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2250–2259.

Qin, H.; Zhang, M.; Ding, Y.; Li, A.; Cai, Z.; Liu, Z.; Yu, F.; and Liu, X. 2023. Bibench: Benchmarking and analyzing network binarization. In *International Conference on Machine Learning*, 28351–28388. PMLR.

Rastegari, M.; Ordonez, V.; Redmon, J.; and Farhadi, A. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, 525–542. Springer.

Shang, Y.; Yuan, Z.; Wu, Q.; and Dong, Z. 2023. Pb-llm: Partially binarized large language models. *arXiv preprint arXiv:2310.00034*.

Wang, C.; Wang, Z.; Xu, X.; Tang, Y.; Zhou, J.; and Lu, J. 2024. Q-VLM: Post-training Quantization for Large Vision-Language Models. *arXiv preprint arXiv:2410.08119*.

Wang, H.; Ma, S.; Dong, L.; Huang, S.; Wang, H.; Ma, L.; Yang, F.; Wang, R.; Wu, Y.; and Wei, F. 2023. Bitnet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*.

Xiao, G.; Lin, J.; Seznec, M.; Wu, H.; Demouth, J.; and Han, S. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, 38087–38099. PMLR.

Yu, H.; Wen, T.; Cheng, G.; Sun, J.; Han, Q.; and Shi, J. 2020. Low-bit quantization needs good distribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 680–681.

Yue, X.; Ni, Y.; Zhang, K.; Zheng, T.; Liu, R.; Zhang, G.; Stevens, S.; Jiang, D.; Ren, W.; Sun, Y.; Wei, C.; Yu, B.; Yuan, R.; Sun, R.; Yin, M.; Zheng, B.; Yang, Z.; Liu, Y.; Huang, W.; Sun, H.; Su, Y.; and Chen, W. 2024. MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI. In *Proceedings of CVPR*.

Zeng, S.; Liu, J.; Dai, G.; Yang, X.; Fu, T.; Wang, H.; Ma, W.; Sun, H.; Li, S.; Huang, Z.; et al. 2024. Flightllm: Efficient large language model inference with a complete mapping flow on fpgas. In *Proceedings of the 2024 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 223–234.

Zhou, S.; Wu, Y.; Ni, Z.; Zhou, X.; Wen, H.; and Zou, Y. 2016. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*.