Breaking Modality Heterogeneity in Low-Bit Quantization for Large Vision-Language Models

Yi Zhong[1], **Haotong Qin**[3], **Xindong Zhang**[2], **Lei Zhang**[2,3], **Guolei Sun**[1*]
[1] VCIP, College of Computer Science, Nankai University
[2] OPPO Research Institute
[3] Department of Computing, Hong Kong Polytechnic University

Abstract

Low-bit post-training quantization (PTQ) is a pivotal technique for deploying Vision-Language Models (VLMs) on resource-constrained devices. However, existing PTQ methods often degrade VLMs' accuracy due to the heterogeneous activation distributions of text and vision modalities during quantization. We find that this cross-modal heterogeneity is distributed unevenly across channels: a small subset of channels contains most modality-specific outliers, and these outliers typically reside in different channels for each modality. Motivated by this, we propose **SplitQ**, a channel-**Split**ting-driven post-training **Q**uantization framework. At its core, SplitQ introduces a novel *Modality-specific Outlier Channel Decoupling* (MOCD) module that effectively isolates salient modality-specific outlier channels with minimal overhead. To further address the remaining cross-modal distribution discrepancies, we design an *Adaptive Cross-Modal Calibration* (ACC) module that employs dual lightweight learnable branches to dynamically mitigate modality-induced quantization errors. Extensive experiments on popular VLMs demonstrate that SplitQ significantly outperforms existing approaches across 6 popular multi-modal datasets under all evaluated quantization settings, including W4A8, W4A4, W3A3, and W3A2. Notably, SplitQ preserves 93.5% of FP16 performance under the challenging W3A3 setting (69.5 vs. 74.3), pushing the efficiency frontier for deploying advanced VLMs. Our code is available at `https://github.com/EMVision-NK/SplitQ`.

1 Introduction

Vision-Language Models (VLMs) [3, 4, 31, 30] have emerged as foundation models for cross-modal understanding, enabling diverse applications such as visual question answering [1, 26], image captioning [57, 15], and multimodal reasoning [35, 54]. However, the massive scale and high computational demands of VLMs pose significant challenges for deployment on resource-constrained devices. Post-Training Quantization (PTQ) [27, 17, 34, 38, 55] has become a popular approach to reduce memory footprints and computational FLOPs. In particular, transformation-based PTQ methods adopt scaling or orthogonal transformations to redistribute weight and activation values, thereby improving quantization accuracy [2, 43, 33, 50, 36].

While transformation-based PTQ methods are well-established for pure Large Language Model (LLM) quantization, extending them to VLMs remains challenging due to the heterogeneous activation distributions across text and vision modalities. Our empirical analysis reveals that this cross-modal heterogeneity is uneven across channels and exhibits a clear channel-wise structural distribution. Specifically, text and vision activations exhibit severe modality-specific outliers in a

*Corresponding author: Guolei Sun, sunguolei.kaust@gmail.com.

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

small number of channels. What's more, the outlier channels of the two modalities are different. Existing approaches [27] typically learn a single transformation shared by both modalities across all channels, where the cross-modal heterogeneity can severely distort the optimization objective. Some methods [58, 17] also attempt to handle the two modalities separately, but such treatment does not fully consider the modality-specific channel outliers.

Motivated by our observations, we propose **SplitQ**, a novel modality-aware framework for VLM quantization. At its core, SplitQ introduces a novel *Modality-specific Outlier Channel Decoupling* (MOCD), which analyzes unique activation patterns (magnitude saliency for vision tokens and response consistency for text tokens) to explicitly divide channels into three disjoint sets: modality-compatible main channels, text-specific outlier channels, and vision-specific outlier channels. The activation and weight matrices are subsequently split according to these channel sets and processed through different quantization paths. Although the modality-compatible main channels exhibit significantly reduced heterogeneity, residual cross-modal distribution discrepancies persist. To address this, SplitQ is further equipped with an *Adaptive Cross-Modal Calibration* (ACC) module which introduces two lightweight learnable branches to mitigate the residual cross-modal quantization errors: one compensates for weight-side errors amplified by cross-modal shifts, while the other specifically recovers activation-side deviations for text tokens.

Extensive experiments on mainstream VLMs demonstrate that SplitQ consistently achieves state-of-the-art performance across various quantization configurations (W4A8, W4A4, W3A3 and W3A2), significantly outperforming existing approaches. Our main contributions are summarized as follows:

- We investigate modality heterogeneity in VLM quantization and reveal that cross-modal distribution shifts are highly channel-dependent. Motivated by this, we propose SplitQ with a novel MOCD module to explicitly separate channels into three groups: modality-compatible main channels, text-specific channels, and vision-specific channels.
- To further reduce cross-modal quantization errors in the main channels, we equip SplitQ with an ACC module which exploits two lightweight learnable branches for weight smoothing and activation compensation.
- We achieve SOTA quantization results on six benchmarks across Qwen2.5-VL-7B and Qwen2.5-VL-3B. SplitQ nearly preserves FP16 performance at W4A4, achieving average scores of 73.5 and 69.6 on the two models, compared with 74.3 and 70.0 for FP16, respectively, and remains effective even at W3A3 and W3A2.

2 Related Work

2.1 LLM Quantization

Current LLM quantization methods can be broadly categorized according to the quantization stage into quantization-aware training (QAT) [5, 7, 32, 37] and post-training quantization (PTQ) [42, 46, 40, 55, 11]. Although QAT usually achieves better accuracy, it requires additional training and substantial computational resources, making PTQ a more practical and widely studied paradigm for efficient LLM deployment. Existing PTQ methods can be further divided into weight-only quantization [29, 12, 63, 19, 23, 6, 52, 10, 21] and weight-activation quantization [36, 62, 43, 25, 18, 39, 20, 49, 61, 2, 50, 33, 56]. Weight-only quantization reduces the precision of model weights while keeping activations in high precision, thereby significantly reducing memory. In contrast, weight-activation quantization quantizes both weights and activations, which can reduce memory consumption and improve inference efficiency. Among weight-only methods, GPTQ [12] leverages approximate second-order information to perform accurate one-shot weight quantization, while AWQ [29] identifies salient weight channels by observing activation distributions and searches for optimal per-channel scaling to protect them during quantization. LLM.int8() [8] isolates large-magnitude outlier feature dimensions for FP16 computation while processing the remaining dimensions in INT8. For weight-activation quantization, SmoothQuant [50] smooths activation outliers by migrating quantization difficulty from activations to weights, enabling effective weight-activation quantization. OmniQuant [39] introduces learnable weight clipping and equivalent transformation to handle extreme values of weights and activation outliers, and SpinQuant [33] learns rotation matrices to reduce outliers in weights and activations.

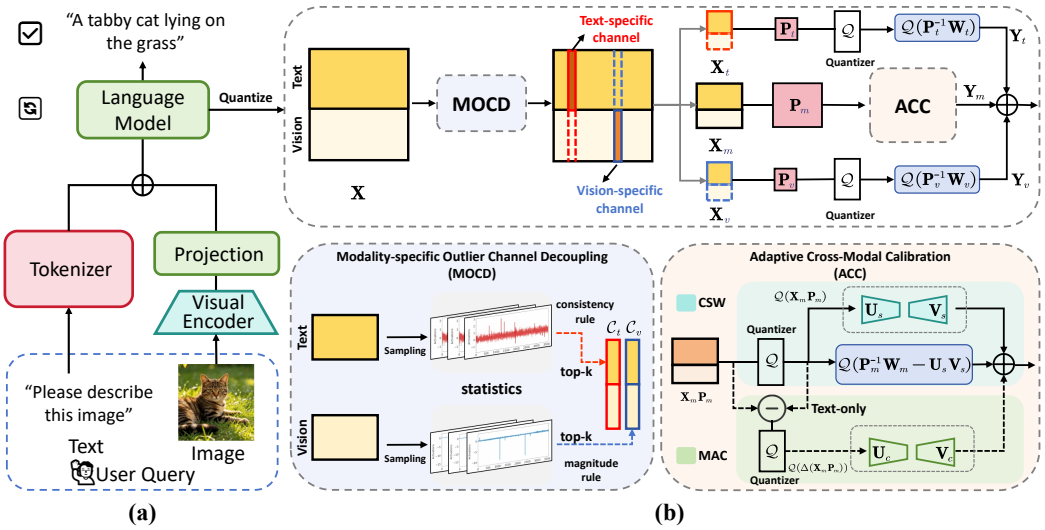

Figure 1: (a) VLM inference pipeline. (b) An overview of SplitQ framework. The two key components of SplitQ include modality-specific outlier channel decoupling (MOCD) module to separate modality-specific outlier channels ($\mathcal{C}_t$ and $\mathcal{C}_v$), and adaptive cross-modal calibration module to further reduce quantization errors caused by cross-modal distribution discrepancies.

2.2 VLM Quantization

As quantization techniques for LLMs have become increasingly mature, recent studies have started to extend quantization techniques to VLMs, where multimodal inputs introduce additional challenges such as modality-dependent activation distributions, redundant visual tokens, and large attention-cache overhead. To address these challenges, several VLM-oriented quantization methods have been proposed. QSLAW [51] uses quantization-aware scale learning to mitigate quantization errors caused by multimodal activation outliers; Q-VLM [44] uses activation entropy as a proxy for block partitioning to balance discretization error and search cost under cross-layer error dependency. MBQ [27] reweights modalities in the calibration loss to account for unequal modality sensitivity when optimizing shared quantization parameters; MQuant [58] assigns modality-specific static scaling factors to visual and textual tokens; VLMQ [53] uses gradient-driven token importance to preserve salient tokens and suppress redundant visual tokens; MASQuant [17] combines modality-specific smoothing with low-rank cross-modal compensation while retaining a shared quantized weight representation. QSVD [48] applies SVD to the combined weight matrices of the query, key, and value, reducing KV-cache size and attention computation overhead; WSVD [45] performs each head SVD with local weighted finetuning to achieve practical decoding latency reduction while preserving accuracy. Bi-VLM [47] further explores saliency-aware hybrid quantization for ultra-low-bit precision. Our work explicitly decouples vision- and text-specific outlier channels using distinct selection criteria and applies independent low-bit transformations to both sets before calibrating the remaining main channels.

3 Transformation-based PTQ

PTQ reduces the computational and memory footprint of models by mapping high-precision floating-point tensor $\mathbf{x}$ to low-precision N-bit integer tensor $\hat{\mathbf{x}}_N$:

$$\hat{\mathbf{x}}_N = Q(\mathbf{x}) = \left(\text{clamp}\left(\left\lfloor \frac{\mathbf{x}}{S_x} \right\rceil + z, q_{\min}, q_{\max}\right) - z\right) \cdot S_x, \tag{1}$$

where S_x and z denote the scale factor and zero-point, respectively. $\lfloor \cdot \rceil$ is the rounding-to-nearest operator, and clamp clips values outside the range $[q_{\min}, q_{\max}]$. We use WxAy to represent the weight-activation quantization configuration of x-bit weights and y-bit activations (e.g., W4A4).

Due to severe outliers in LLM or VLM activations, recent approaches [43, 33, 50, 2] introduce Transformation-based PTQ (TPTQ), where a learnable transformation matrix (e.g., diagonal or

Algorithm 1 SplitQ linear layer.

Require: Activation $\mathbf{X}$, weight $\mathbf{W}$, quantizer $\mathcal{Q}(\cdot)$, channel sets $\mathcal{C}_m, \mathcal{C}_t, \mathcal{C}_v$, transforms $\mathbf{P}_m, \mathbf{P}_t, \mathbf{P}_v$, learnable low-rank branches $(\mathbf{U}_s, \mathbf{V}_s)$ and $(\mathbf{U}_c, \mathbf{V}_c)$.
Ensure: Quantized output $\mathbf{Y}$.

Split $\mathbf{X}$ and $\mathbf{W}$ by $\mathcal{C}_m, \mathcal{C}_t, \mathcal{C}_v$: $(\mathbf{X}_m, \mathbf{X}_t, \mathbf{X}_v)$, $(\mathbf{W}_m, \mathbf{W}_t, \mathbf{W}_v)$.
Compute text-specific path: $\mathbf{Y}_t = \mathcal{Q}(\mathbf{X}_t\mathbf{P}_t)\mathcal{Q}(\mathbf{P}_t^{-1}\mathbf{W}_t)$.
Compute vision-specific path: $\mathbf{Y}_v = \mathcal{Q}(\mathbf{X}_v\mathbf{P}_v)\mathcal{Q}(\mathbf{P}_v^{-1}\mathbf{W}_v)$.
Quantize the main activation: $\widehat{\mathbf{X}}_m = \mathcal{Q}(\mathbf{X}_m\mathbf{P}_m)$.
if Cross-modal Weight Smoothing is enabled **then**
    Compute main path: $\mathbf{Y}_m = \widehat{\mathbf{X}}_m\mathcal{Q}(\mathbf{P}_m^{-1}\mathbf{W}_m - \mathbf{U}_s\mathbf{V}_s) + \widehat{\mathbf{X}}_m\mathcal{Q}(\mathbf{U}_s)\mathcal{Q}(\mathbf{V}_s)$.
else
    Compute main path: $\mathbf{Y}_m = \widehat{\mathbf{X}}_m\mathcal{Q}(\mathbf{P}_m^{-1}\mathbf{W}_m)$.
end if
if Modality-specific Activation Compensation is enabled **then**
    Compensate text tokens: $\mathbf{Y}_m^{\text{text}} \leftarrow \mathbf{Y}_m^{\text{text}} + \mathcal{Q}\left(\mathbf{X}_m\mathbf{P}_m - \widehat{\mathbf{X}}_m\right)^{\text{text}}\mathcal{Q}(\mathbf{U}_c)\mathcal{Q}(\mathbf{V}_c)$.
end if
Merge three paths: $\mathbf{Y} = \mathbf{Y}_m + \mathbf{Y}_t + \mathbf{Y}_v$.
return $\mathbf{Y}$.

orthogonal) is adopted to transfer these outliers from activations to weights, thereby improving quantization friendliness. Specifically, for a linear layer $\mathbf{Y} = \mathbf{X}\mathbf{W}$, where $\mathbf{X} \in \mathbb{R}^{T\times D_{\text{in}}}$ and $\mathbf{W} \in \mathbb{R}^{D_{\text{in}}\times D_{\text{out}}}$, the layer can be reformulated based on computational invariance as:

$$\mathbf{Y} = (\mathbf{X}\mathbf{P}) \cdot (\mathbf{P}^{-1}\mathbf{W}), \tag{2}$$

where $\mathbf{P}$ is a linear optimization matrix that reshapes feature distributions to facilitate adaptive quantization. To optimize learnable parameters, the objective is defined as the reconstruction loss:

$$L = \mathcal{L}\left(Q(\mathbf{X}\mathbf{P}) \cdot Q(\mathbf{P}^{-1}\mathbf{W}), \mathbf{X}\mathbf{W}\right). \tag{3}$$

4 Methodology

In this section, we present our SplitQ framework, a PTQ approach for VLMs. Fig. 1 provides an overview of the framework, while the core quantized linear layer is detailed in Alg. 1. We first introduce our motivations in §4.1, followed by modality-specific outlier channel decoupling in §4.2. Finally, we explain our adaptive cross-modal calibration module in §4.3.

4.1 Motivations

Modality-specific Outlier Channels. We visualize the distributions of text and vision tokens across channels in Fig. 2 to investigate the fundamental challenges of quantizing heterogeneous modalities. Our key observations are as follows. ① The activation distributions of different modalities exhibit fundamental differences. Vision activations usually follow long-tailed distributions where only a few channels contain extremely large magnitudes, i.e., outliers that need to be handled carefully. In contrast, text activation outliers are distributed more evenly across channels. ② For both modalities, outlier channels account for only a small fraction of all channels. ③ More importantly, the activation outliers of text and vision modalities are located in different channels. This separation provides us an opportunity to disentangle the channel outliers for different modalities.

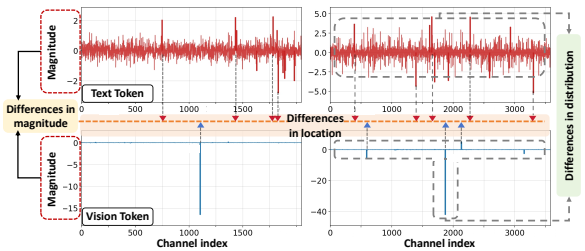

Figure 2: Distributions of text and vision activations across channels. *Left*: layer 5 of Qwen2.5-VL 3B; *Right*: layer 10 of Qwen2.5-VL 7B. Notable divergence in distributions underscores the modality heterogeneity.

From observation ①, we can identify a key reason why PTQ for VLM is particularly difficult. In existing transformation-based PTQ methods where the activations of all channels are optimized with

a shared transformation matrix $\mathbf{P}$, the modality-specific outlier channels significantly interfere with each other, leading to suboptimal transformations for both modalities.

Inspired by observations ② and ③, we propose a Modality-specific Outlier Channel Decoupling (MOCD) technique to disentangle the interference of modality-specific outliers by separating the text-specific and vision-specific outlier channels out of all channels, resulting in three groups of channels: vision-specific, text-specific and modality-compatible channels. Specifically, let $\mathcal{C} = \{0, 1, ..., D_{\text{in}}\}$ denote all channel indexes. MOCD partitions the input channels into three disjoint sets:

$$\mathcal{C} = \mathcal{C}_m \cup \mathcal{C}_t \cup \mathcal{C}_v, \qquad \mathcal{C}_m \cap \mathcal{C}_t = \emptyset, \qquad \mathcal{C}_m \cap \mathcal{C}_v = \emptyset, \tag{4}$$

where $\mathcal{C}_m$, $\mathcal{C}_t$, and $\mathcal{C}_v$ denote modality-compatible, text and vision channels, respectively. They also satisfy the following criterion since the number of outlier channels is much smaller than D_{in}:

$$|\mathcal{C}_m| \gg |\mathcal{C}_t|, \qquad |\mathcal{C}_m| \gg |\mathcal{C}_v|, \tag{5}$$

where $|\text{set}|$ denotes the cardinality of the set. Accordingly, the activation/weight matrices are split as:

$$\mathbf{X} \to \{\mathbf{X}_m, \mathbf{X}_t, \mathbf{X}_v\}, \qquad \mathbf{W} \to \{\mathbf{W}_m, \mathbf{W}_t, \mathbf{W}_v\}. \tag{6}$$

Three matrix pairs $\{\mathbf{X}_m, \mathbf{W}_m\}$, $\{\mathbf{X}_t, \mathbf{W}_t\}$, $\{\mathbf{X}_v, \mathbf{W}_v\}$ are processed differently by learning different transformations. The details of MOCD are further elaborated in §4.2.

Cross-modal Low-bit Quantization Difficulty. The heterogeneity between vision and text persists beyond the obvious outlier channels. Even after decoupling these outliers, the remaining channels still exhibit distribution differences across modalities. More specifically, the activation $\mathbf{X}_m$ and weight $\mathbf{W}_m$ still contain mixed distributions of text and vision modalities.

While high-precision quantization can tolerate such variations, low-bit quantization is highly sensitive to them, as demonstrated in our experiments (§5). Starting from the baseline FlatQuant [43], we add the proposed MOCD and use an affine transformation matrix to quantize $\{\mathbf{X}_m, \mathbf{W}_m\}$. This approach (FlatQuant+MOCD) gives excellent results at W4A4, but its performance is unsatisfactory at challenging W3A3 and W3A2. To mitigate the subtle distribution differences among modalities within $\mathbf{X}_m$, we design an Adaptive Cross-modal Calibration (ACC) module to further improve the quantization of $\{\mathbf{X}_m, \mathbf{W}_m\}$. Details of the ACC module are provided in §4.3.

4.2 Modality-specific Outlier Channel Decoupling

As discussed above, the input channels $\mathcal{C}$ is split into $\mathcal{C}_m$, $\mathcal{C}_t$, and $\mathcal{C}_v$. In this part, we describe the process of selecting text- and vision- specific outlier channels in detail.

Modality-aware Channel Selection. We first select vision-specific outlier channels by magnitude, followed by identifying text-specific outlier channels via a consistency-based calibration proxy. Let $\mathcal{T}_v$ and $\mathcal{T}_t$ denote the sampled vision and text tokens, respectively, and let $\mathcal{C}$ be the full channel set. For vision tokens, we score each channel by its maximum absolute activation:

$$s_v(c) = \max_{i \in \mathcal{T}_v} |X_{i,c}|. \tag{7}$$

The vision-specific outlier set is obtained by selecting the top-K_v channels:

$$\mathcal{C}_v = \text{TopK}_{c \in \mathcal{C}} \left(s_v(c), K_v \right). \tag{8}$$

After this, we denote the remaining channels as $\mathcal{C}' = \mathcal{C} \backslash \mathcal{C}_v$. For text tokens, absolute magnitude is less indicative, as outlier channels are related to unstable relative responses across tokens. We therefore use the percentile rank within each token as a scale-insensitive measure of channel importance:

$$r_{i,c} = \frac{1}{|\mathcal{C}'|} \sum_{j \in \mathcal{C}'} \mathbb{I} \left(|X_{i,j}| \le |X_{i,c}| \right), \quad i \in \mathcal{T}_t,\ c \in \mathcal{C}'. \tag{9}$$

This rank-based measure suppresses token-wise scale variation and makes channel responses comparable across text tokens. For each linear layer, we sample $|\mathcal{T}_t| = 3{,}000$ text tokens from calibration samples. For each channel $c \in \mathcal{C}'$, we independently apply one-dimensional K-means ($K = 3$) to its percentile-rank sequence $\{r_{i,c}\}_{i \in \mathcal{T}_t}$, using random initialization controlled by the global seed, a maximum of 300 iterations, and a tolerance of 10^{-4}. We characterize the cross-token variability of each channel using its cluster centers. Let $\mu_{c,k}$ denote the center of the k-th cluster for channel c, and let $\bar{\mu}_c = \frac{1}{K} \sum_{k=1}^{K} \mu_{c,k}$. We define the text-channel score as

$$s_t(c) = \frac{1}{K} \sum_{k=1}^{K} \left(\mu_{c,k} - \bar{\mu}_c \right)^2. \tag{10}$$

A larger $s_t(c)$ indicates more significant differences among the channel's characteristic relative-response levels across text tokens. We prioritize such channels for separate calibration to reduce their interference with the remaining channels. We select the top-K_t channels as text-specific channels:

$$\mathcal{C}_t = \mathrm{TopK}_{c \in \mathcal{C}'} \left(s_t(c), K_t \right). \tag{11}$$

The remaining channels form the modality-compatible main channel set:

$$\mathcal{C}_m = \mathcal{C} \setminus (\mathcal{C}_v \cup \mathcal{C}_t). \tag{12}$$

Quantization Calibration. Following channel separation, activation $\mathbf{X}$ and weight $\mathbf{W}$ are split into $\{\mathbf{X}_m, \mathbf{X}_t, \mathbf{X}_v\}$ and $\{\mathbf{W}_m, \mathbf{W}_t, \mathbf{W}_v\}$, respectively. The two matrix groups $\{\mathbf{X}_t, \mathbf{W}_t\}$ and $\{\mathbf{X}_v, \mathbf{W}_v\}$ correspond to text- and vision- specific outlier channels. They are processed with independent modality-aware transformations:

$$\mathbf{Y}_t = Q(\mathbf{X}_t \mathbf{P}_t) Q(\mathbf{P}_t^{-1} \mathbf{W}_t), \tag{13}$$

$$\mathbf{Y}_v = Q(\mathbf{X}_v \mathbf{P}_v) Q(\mathbf{P}_v^{-1} \mathbf{W}_v). \tag{14}$$

Here, $\mathbf{P}_t$ and $\mathbf{P}_v$ are optimized for the text- and vision-specific outlier channels, respectively. The main matrix group $\{\mathbf{X}_m, \mathbf{W}_m\}$ corresponding to modality-compatible channels is processed by our adaptive cross-modal calibration module in §4.3.

4.3 Adaptive Cross-modal Calibration

As detailed above, MOCD isolates modality-specific outliers, which are then processed through independent paths (Eq. 13, 14). The remaining channels constitute the main activation-weight pair $\{\mathbf{X}_m, \mathbf{W}_m\}$. We can adopt a shared transformation matrix $\mathbf{P}_m$ for quantization as follows:

$$\mathbf{Y}_m = Q(\mathbf{X}_m \mathbf{P}_m) Q(\mathbf{P}_m^{-1} \mathbf{W}_m). \tag{15}$$

$\{\mathbf{X}_m, \mathbf{W}_m\}$ encompass the majority channels of original $\{\mathbf{X}, \mathbf{W}\}$. Although these channels are relatively compatible after outlier decoupling, their underlying cross-modal distribution differences persist. Forcing a unified transformation $\mathbf{P}_m$ over such a large-scale channel space inevitably leads to sub-optimal smoothing for both modalities. Consequently, this amplifies the weight-side quantization error $\Delta(\mathbf{P}_m^{-1}\mathbf{W}_m)$ and activation-side quantization error $\Delta(\mathbf{X}_m\mathbf{P}_m)$. Here, we denote $\Delta(\mathbf{M}) = \mathbf{M} - Q(\mathbf{M})$ to represent the quantization error of $\mathbf{M}$. To solve this issue, we introduce the ACC module, achieving fine-grained calibration.

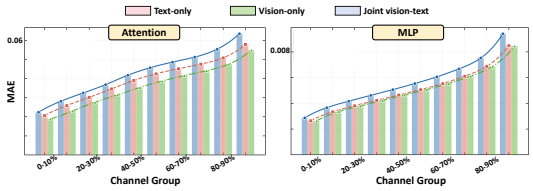

Figure 3: Comparison of weight quantization error $\Delta(\mathbf{P}_m^{-1}\mathbf{W}_m)$ across channels using activations of different modalities: text-only, vision-only, and joint vision-text. The x-axis (Channel Group) partitions all channels, sorted in ascending order of quantization error MAE, into 10 equal-sized bins (each containing 10% of channels). The results show that vision-text activations consistently amplify this error term.

Cross-modal Weight Smoothing (CWS). As shown in Fig. 3, the main path still exhibits multi-modal sensitivity. By comparing PTQ results calibrated on single-modal versus multi-modal tokens, we observe that the weight-side error $\Delta(\mathbf{P}_m^{-1}\mathbf{W}_m)$ is significantly amplified by the distributional differences between vision and text activations. To mitigate this cross-modal conflict, instead of quantizing the weight $\mathbf{P}_m^{-1}\mathbf{W}_m$ directly, we extract a low-rank adaptive component $\mathbf{U}_s\mathbf{V}_s$ to absorb these sensitive patterns:

$$\mathbf{P}_m^{-1}\mathbf{W}_m = \left(\mathbf{P}_m^{-1}\mathbf{W}_m - \mathbf{U}_s\mathbf{V}_s\right) + \mathbf{U}_s\mathbf{V}_s, \tag{16}$$

where $\mathbf{U}_s \in \mathbb{R}^{D_m \times r}$ and $\mathbf{V}_s \in \mathbb{R}^{r \times D_{\text{out}}}$ are learnable low-rank matrices. Let $\widehat{\mathbf{X}}_m = Q(\mathbf{X}_m\mathbf{P}_m)$ denote the quantized input. According to Eq. 16, we explicitly decouple $\mathbf{P}_m^{-1}\mathbf{W}_m$ into a main residual part $(\mathbf{P}_m^{-1}\mathbf{W}_m - \mathbf{U}_s\mathbf{V}_s)$ and a low-rank part $\mathbf{U}_s\mathbf{V}_s$. By quantizing and computing these components independently, the final output of the main path is formulated as:

$$\mathbf{Y}_m = \widehat{\mathbf{X}}_m Q\left(\mathbf{P}_m^{-1}\mathbf{W}_m - \mathbf{U}_s\mathbf{V}_s\right) + \widehat{\mathbf{X}}_m Q(\mathbf{U}_s) Q(\mathbf{V}_s). \tag{17}$$

While [24] adopts a similar low-rank architecture to absorb main energy or outliers in single-modal scenarios, our structural intent is distinct. We utilize the low-rank branch $\mathbf{U}_s\mathbf{V}_s$ adaptively to absorb the weight variations triggered by cross-modal activation shifts. By extracting these sensitive cross-modal components, the remaining main weight $(\mathbf{P}_m^{-1}\mathbf{W}_m - \mathbf{U}_s\mathbf{V}_s)$ exhibits a smoother distribution, allowing for low-error quantization.

Modality-specific Activation Compensation (MAC). While CWS reduces the weight-side quantization error caused by cross-modal heterogeneity, the activation-side remains more challenging. Since the shared transformation $\mathbf{P}_m$ is optimized under conflicting modality distributions, its smoothing effect is inevitably compromised, leaving a non-negligible activation residual $\Delta(\mathbf{X}_m\mathbf{P}_m)$. Unlike static weights, activations are input-dependent and dynamically vary across modalities, making it difficult to reformulate their quantization-sensitive components through structural decomposition (as we did in Eq. 17). We therefore introduce a direct compensation branch to recover the output deviation caused by activation quantization:

$$\Delta\mathbf{Y}_m^{\text{act}} = \Delta(\mathbf{X}_m\mathbf{P}_m)\mathbf{P}_m^{-1}\mathbf{W}_m, \tag{18}$$

where $\Delta\mathbf{Y}_m^{\text{act}}$ is the output deviation induced by the activation residual.

As revealed by prior observations [58], text activations are more sensitive to quantization noise due to their dense semantic nature, whereas vision activations are relatively redundant. Therefore, text compensation usually captures the more critical activation-side errors. For efficient implementation, we use learnable low-rank matrices $\mathbf{U}_c\mathbf{V}_c$ to approximate the weight mapping $\mathbf{P}_m^{-1}\mathbf{W}_m$ in Eq. 18. Furthermore, all factors in this branch are fully quantized, and we apply this compensation only to the text-side output $\mathbf{Y}_m^{\text{text}}$:

$$\mathbf{Y}_m^{\text{text}} \leftarrow \mathbf{Y}_m^{\text{text}} + Q\big(\Delta(\mathbf{X}_m\mathbf{P}_m)^{\text{text}}\big)Q(\mathbf{U}_c)Q(\mathbf{V}_c), \tag{19}$$

where $\mathbf{Y}_m^{\text{text}}$ denotes the output for text activations, and the added term is the quantized compensation for the activation-induced output deviation.

Learnable Constrained Low-rank Matrix. Both CWS and MAC rely on low-rank branches. However, freely learned parameters, such as LoRA [16] or QLoRA [9], easily overfit on small calibration sets, while fixed SVD-based components [24, 17] lack the flexibility needed to absorb quantization errors arising from modality heterogeneity. To balance these two choices, we propose a learnable constrained low-rank parameterization, as shown in Fig. 4. The low-rank components in both branches are closely related to the transformed weight $\mathbf{P}_m^{-1}\mathbf{W}_m$: the smoothing branch is derived directly from it, while the compensation branch is designed to approximate it. Therefore, we build the low-rank structure on the shared transformation $\mathbf{P}_m^{-1}$ and the original weight $\mathbf{W}_m$. We first compute its rank-r truncated SVD:

$$\mathbf{W}_m \approx \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^\top, \tag{20}$$

where $\mathbf{U}_r \in \mathbb{R}^{D_{\text{in}}\times r}$ and $\mathbf{V}_r^\top \in \mathbb{R}^{r\times D_{\text{out}}}$. We then construct the low-rank matrices for both branches as:

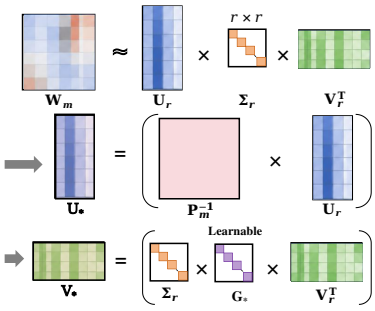

Figure 4: Construction process.

$$\mathbf{U}_* = \mathbf{P}_m^{-1}\mathbf{U}_r, \qquad \mathbf{V}_* = \mathbf{\Sigma}_r\mathbf{G}_*\mathbf{V}_r^\top, \quad *\in\{s,c\}. \tag{21}$$

In this design, the constraint comes from anchoring the matrices to the shared transformation $\mathbf{P}_m^{-1}$ and the dominant SVD subspace of $\mathbf{W}_m$, which ensures structural consistency with the main path. Meanwhile, learnability is confined to a diagonal gating matrix $\mathbf{G}_* \in \mathbb{R}^{r\times r}$, allowing each branch to adaptively reweight the singular directions without introducing significant parameter overhead.

5 Experiments

5.1 Experimental Setups

We evaluate on four VLMs from two representative architectures, including Qwen2.5-VL [4] (3B/7B) and LLaVA-v1.5 [31] (7B/13B). Following recent works [27, 17], we quantize only the LLM component. For each matched bit setting, all methods use the same image-text calibration samples, random seeds, calibration and evaluation prompts, and benchmarks. In line with these prior works [27, 17], we use `lmms-eval` [60] to re-evaluate all FP16 and quantized models within a common pipeline. All reported numerical scores are obtained from complete evaluation runs.

For Qwen2.5-VL, we compare SplitQ with SmoothQuant [50], MBQ [27], and MASQuant [17]. For LLaVA-v1.5, we compare SplitQ with Q-VLM [44], DuQuant [28], QSVD [48] and MQuant [58]. Method-specific hyperparameters follow the recommended settings of each baseline. We use MMMU [59], OCRBench [13], TextVQA [41], SEED-Bench [22], VizWiz [14], and ScienceQA [35] as our benchmarks. For the transformation matrices $\mathbf{P}$ used in our method, we adopt the learnable

Table 1: Quantization results on Qwen2.5-VL-3B and Qwen2.5-VL-7B. Our method clearly outperforms strong baselines under all settings. A score of 0.0 denotes a completed evaluation with no correct answers, whereas '−' indicates that unstable generation prevented a meaningful full-benchmark result. All methods were attempted under the same evaluation protocol.

		Qwen2.5-VL-3B							Qwen2.5-VL-7B						
Method	Bits	MMMU	SEED	OCRBench	VizWiz	ScienceQA	TextVQA	Avg.	MMMU	SEED	OCRBench	VizWiz	ScienceQA	TextVQA	Avg.
FP16	W16A16	42.2	69.9	79.3	69.1	81.9	77.9	70.0	46.7	73.0	83.8	70.8	88.4	82.9	74.3
SQ	W4A8	25.6	55.7	66.9	57.5	72.1	63.9	56.9	37.8	62.7	70.2	61.5	83.3	71.1	64.4
MBQ	W4A8	41.2	58.2	66.9	65.0	76.7	73.4	63.5	43.3	67.7	74.1	64.3	86.0	74.8	68.3
MASQ	W4A8	**46.7**	59.7	67.2	62.7	77.9	69.2	63.9	43.4	69.5	72.8	66.4	85.7	77.0	69.1
SplitQ	W4A8	46.3	**69.7**	**79.1**	**67.7**	**82.4**	**77.4**	**70.4**	**49.1**	**73.2**	**83.5**	**68.7**	**88.1**	**82.6**	**74.2**
SQ	W4A4	23.3	0.0	0.0	0.0	0.0	0.0	3.9	24.8	0.0	0.2	0.0	0.7	0.0	4.3
MBQ	W4A4	25.0	0.0	0.0	0.0	0.0	0.0	4.2	26.7	3.0	0.5	0.0	0.9	0.0	5.2
MASQ	W4A4	26.7	0.0	7.7	0.0	0.0	0.0	5.7	25.0	0.6	13.2	0.0	7.1	0.4	7.7
SplitQ	W4A4	**43.7**	**69.3**	**78.8**	**67.6**	**81.5**	**77.0**	**69.6**	**46.9**	**72.6**	**83.0**	**68.4**	**87.9**	**82.5**	**73.5**
SQ	W3A3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MBQ	W3A3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MASQ	W3A3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SplitQ	W3A3	**39.8**	**67.2**	**74.5**	**63.4**	**70.6**	**71.2**	**64.5**	**43.9**	**71.1**	**79.6**	**63.7**	**80.3**	**78.5**	**69.5**
SplitQ	W3A2	**33.1**	**51.3**	**57.0**	**54.5**	**49.1**	**50.2**	**49.2**	**35.6**	**57.3**	**61.5**	**50.1**	**55.6**	**62.6**	**53.7**

Table 2: Quantization results on LLaVA-v1.5 7B and 13B.

Model	Bits	Duquant				Q-VLM				QSVD				MQuant				SplitQ (Ours)			
		SEED-I	VizWiz	SciQA	Avg	SEED-I	VizWiz	SciQA	Avg	SEED-I	VizWiz	SciQA	Avg	SEED-I	VizWiz	SciQA	Avg	SEED-I	VizWiz	SciQA	Avg
LLaVA-v1.5 7B	FP16	66.2	54.3	70.0	63.5	66.2	54.3	70.0	63.5	66.2	54.3	70.0	63.5	66.2	54.3	70.0	63.5	66.2	54.3	70.0	63.5
	W4A8	54.4	50.6	55.3	53.4	46.1	48.7	53.2	49.3	57.8	53.5	63.6	58.3	62.9	53.3	67.5	61.2	65.3	56.6	70.3	**64.1**
	W4A4	51.5	49.8	54.8	52.0	37.2	48.9	53.1	46.4	55.1	53.6	57.7	55.5	54.9	52.5	60.7	56.0	64.8	54.6	69.4	**62.9**
	W3A3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	61.5	57.2	64.9	**61.2**
	W3A2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	54.4	55.7	52.3	**54.1**
LLaVA-v1.5 13B	FP16	68.3	57.3	74.7	66.8	68.3	57.3	74.7	66.8	68.3	57.3	74.7	66.8	68.3	57.3	74.7	66.8	68.3	57.3	74.7	66.8
	W4A8	66.1	56.5	72.3	65.0	64.2	55.7	68.4	62.8	66.8	56.9	75.0	66.2	68.1	56.9	74.5	66.5	68.3	57.7	74.7	**66.9**
	W4A4	64.6	55.3	67.2	62.4	48.3	50.9	65.0	54.7	67.0	56.8	67.8	63.9	67.2	56.9	70.7	64.9	67.9	57.2	74.0	**66.4**
	W3A3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	64.6	56.6	69.5	**63.6**
	W3A2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	59.2	57.0	61.8	**59.3**

affine form from FlatQuant [43]. The smoothing and compensation branches are optional components. By default, MOCD selects 2% modality-aware outlier channels for vision and text modalities, and the low-rank [16] decomposition ranks of the smooth and compensation branches are set to 2% and 3% of the full weight rank, respectively. To preserve the representation ability of all auxiliary low-rank structures, including modality-specific independent branches as well as CWS and MAC, we apply a 4-bit quantization lower bound to these components.

5.2 Main Results

Table 1 presents results on the Qwen2.5-VL series. At W4A8 and W4A4, SplitQ outperforms all baselines and obtains performance on par with full-precision models, while other methods suffer severe accuracy drops, particularly at W4A4. For challenging W3A3 and W3A2 settings, most existing methods cannot output normal outputs, yet SplitQ still maintains competitive performance.

Following [48], Table 2 further provides results on the LLaVA-v1.5 series across the same evaluation benchmarks. SplitQ still obtains superior results at W4A8 and W4A4. In lower-bit configurations, existing baselines suffer from unstable inference or complete failure, whereas SplitQ remains robust. These findings demonstrate the promising generalization ability of SplitQ across various VLM architectures, and verify that effectively addressing modality heterogeneity through explicit decoupling and refined calibration is important for stable low-bit multimodal quantization.

5.3 Ablation Study

Channel Selection Rule. To verify the efficacy of MOCD, we compare it with various channel selection schemes: no selection ('None'), random selection ('Random'), and heuristic methods based on the maximum or mean activations of text ('Text_max'), vision ('Vision_max'), or all multimodal tokens ('Multi_max', 'Multi_mean'). All approaches select an equal number of channels for fair comparison. As shown in Table 4, activation-driven methods consistently outperform random selection, demonstrating the importance of salient channels. More importantly, MOCD delivers the best overall results across all settings, showing that it effectively selects outlier channels.

Table 4: Ablation of channel selection rules and cluster number (k) within MOCD at W3A3.

Method	Qwen2.5-VL-3B				Qwen2.5-VL-7B			
	MMMU	OCRBench	ScienceQA	Average	MMMU	OCRBench	ScienceQA	Average
FP16	42.2	79.3	81.9	67.8	46.7	83.8	88.4	73.0
None	34.0	66.1	61.6	53.9	35.1	73.2	71.5	59.9
Random	33.8	69.4	61.6	54.9	39.4	76.2	75.1	63.5
Text_max	38.1	71.3	65.6	58.3	38.9	71.7	79.1	63.2
Vision_max	37.0	69.8	68.2	58.3	39.1	77.3	77.0	64.4
Multi_max	34.1	72.7	61.7	56.1	37.1	78.3	77.7	64.3
Multi_mean	36.8	66.0	63.3	55.3	36.8	76.1	75.7	62.8
MOCD (k=1)	**37.7**	70.2	68.2	58.4	**40.3**	76.3	77.0	64.5
MOCD (k=3)	37.2	71.1	**69.0**	**59.1**	40.2	**77.8**	**78.0**	**65.3**
MOCD (k=5)	37.2	**71.5**	68.8	**59.1**	40.0	77.2	77.6	64.9

Table 5: Ablation on the compensation target of MAC at W3A2.

Method	Target	Qwen2.5-VL-3B				Qwen2.5-VL-7B			
		MMMU	OCRBench	ScienceQA	Average	MMMU	OCRBench	ScienceQA	Average
w/o MAC	/	29.2	39.9	20.1	29.7	28.2	47.0	19.1	31.4
w/ MAC	vision	31.5	46.6	28.9	35.7	29.4	50.0	42.6	40.6
w/ MAC	text	32.9	56.8	48.9	46.2	35.6	**61.9**	55.3	**50.9**
w/ MAC	vision & text	**33.0**	**57.2**	**49.6**	**46.6**	**35.6**	61.5	**55.6**	**50.9**

Cluster Number within MOCD. We also conduct ablation experiments on the cluster number k in Table 4. The results suggest that $k = 3$ yields optimal performance, which we adopt by default.

Stability of Outlier Channel Selection in MOCD. Practical post-training quantization (PTQ) relies on limited calibration data. To verify that MOCD remains stable under varying calibration set sizes, we measure the Jaccard similarity between outlier channels selected from calibration set of different sizes and those identified using our default calibration setting (128 samples). As shown in Table 3, the outlier channels selected by MOCD are stable with respect to the size of calibration set.

Table 3: Similarity of MOCD Channel Selection under different calibration set size (N).

Sim. (%)	32	64	256
Qwen2.5-VL 7B	94.1	93.3	94.5
LLaVA-v1.5 7B	89.5	95.5	95.7

Table 6: Ablation study of core components in SplitQ at W3A3 and W3A2. FlatQ: FlatQuant, CWS: Cross-Modal Weight Smoothing, MAC: Modality-Specific Activation Compensation.

Method	bits	Qwen2.5-VL-3B				Qwen2.5-VL-7B			
		MMMU	OCRBench	ScienceQA	Average	MMMU	OCRBench	ScienceQA	Average
FP16	W16A16	42.2	79.3	81.9	67.8	46.7	83.8	88.4	73.0
Base (FlatQ)	W3A3	34.0	66.1	61.6	53.9	35.1	73.2	71.5	59.9
MOCD		37.2	71.1	69.0	59.1	40.2	77.8	78.0	65.3
MOCD + CWS		37.7	72.7	69.6	60.0	41.1	78.2	78.8	66.0
MOCD + CWS + MAC		**39.8**	**74.5**	**70.6**	**61.7**	**43.9**	**79.6**	**80.3**	**68.0**
Base (FlatQ)	W3A2	25.5	0.0	0.0	8.5	26.0	2.5	0.04	9.6
MOCD		26.6	21.7	3.5	17.2	24.9	40.2	2.4	22.5
MOCD + CWS		29.2	39.9	20.1	29.7	28.2	47.0	19.1	31.4
MOCD + CWS + MAC		**32.9**	**56.8**	**48.9**	**46.2**	**35.6**	**61.9**	**55.3**	**50.9**

Ablation on Compensation Target. We study which modality should be compensated in MAC by comparing four settings: without MAC, compensating only vision activations, only text activations, and both modalities. As shown in Table 5, text-only compensation yields substantially larger gains than vision-only compensation on both Qwen2.5-VL-3B and Qwen2.5-VL-7B. Moreover, compensating both modalities brings only marginal or no improvement over text-only compensation. This shows that text activations are more sensitive to quantization noise, whereas vision activations are relatively more redundant. Therefore, we adopt text-only compensation in the final design for a better trade-off between performance and inference efficiency.

Core Component Ablation Study. Table 6 ablates the core components of SplitQ at W3A3 and W3A2. "Base", i.e., FlatQuant, only uses the affine transformation matrix $\mathbf{P}$. Starting from the baseline, we gradually add our proposed techniques: MOCD, CWS, and MAC. The results verify the effectiveness of each component. MOCD consistently improves over the baseline, indicating the necessity of modality-specific channel separation. Furthermore, the significance of the weight smoothing and activation compensation modules is highlighted under the more challenging W3A2 setting. While MOCD successfully isolates heterogeneous outliers, both CWS and MAC effectively ease the quantization difficulty of the remaining activations. These results support our design insight: while decoupling outliers is crucial, effectively smoothing the weights and compensating for activations are indispensable for preventing severe degradation at ultra-low bits.

5.4 Inference Speedup

Previous works [24, 17] have shown that the Nunchaku[24] efficiently supports low-rank structured quantization. To evaluate the practical efficiency of SplitQ, we implement a customized W4A4 CUDA kernel that fuses projection and quantization operators to reduce memory access. Table 7 presents efficiency comparisons on Qwen2.5-VL-7B under batch size of 8. SplitQ achieves a 2.89× speedup over FP16 and reduces memory usage by 1.82×. Compared with MBQ and MASQuant, SplitQ introduces only moderate latency overhead, while maintaining similar memory consumption. These results demonstrate that SplitQ remains practical for efficient VLM deployment at W4A4.

Table 7: Efficiency comparisons of different methods on Qwen2.5-VL-7B on RTX 4090 (seq. len=2048) at W4A4. MASQ: MASQuant.

Method	Prefill (ms)	Speedup	Mem (GB)	Mem Saving
FP16	2146.01	-	17.27	-
MBQ	643.68	3.33×	8.89	1.94×
MASQ	696.44	3.07×	9.42	1.83×
SplitQ	742.17	2.89×	9.48	1.82×

6 Conclusion

In this study, we identify the modality heterogeneity barrier as an important challenge for high-precision VLM quantization. To address this, we propose SplitQ, which mitigates the adverse effects of differing modal distributions through channel-level decoupling and cross-modal calibration. SplitQ achieves state-of-the-art performance across multiple benchmarks. Furthermore, our exploration into ultra-low-bit scenarios demonstrates the potential for significantly higher compression ratios, providing insights for future multimodal intelligence research.

7 Limitations

Despite these promising results, our work has two main limitations.

First, the multi-branch design introduces additional runtime overhead. The CWS and MAC modules bring extra computation and memory costs. The full SplitQ pipeline yields 14.5–15.2% prefill overhead compared with the MOCD-only baseline at batch size 1, which motivates further hardware-aware optimization and operator fusion to improve practical deployment efficiency.

Second, the current design adopts a fixed modality-specific channel ratio shared across all quantized layers. This uniform setting lacks flexibility to adapt to varied architectures, layer properties, and data distributions. Exploring adaptive ratio adjustment strategies has the potential to further enhance the flexibility of our method, which we leave as future work.

Acknowledgments and Disclosure of Funding

Funding. This work is supported in part by the Fundamental Research Funds for the Central Universities (Nankai University, No. 070-63263249), and in part by NSFC (No. 62676206). The computational resources are supported by the Supercomputing Center of Nankai University (NKSC).

Competing Interests. The authors declare no competing interests.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015.

[2] Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. In *NeurIPS*, 2024.

[3] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

[5] Yelysei Bondarenko, Riccardo Del Chiaro, and Markus Nagel. Low-rank quantization-aware training for llms. *arXiv preprint arXiv:2406.06385*, 2024.

[6] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. Quip: 2-bit quantization of large language models with guarantees. In *NeurIPS*, 2023.

[7] Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. Efficientqat: Efficient quantization-aware training for large language models. In *ACL*, 2025.

[8] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35: 30318–30332, 2022.

[9] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In *NeurIPS*, 2023.

[10] Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2306.03078*, 2023.

[11] Xin Ding, Xiaoyu Liu, Zhijun Tu, Yun Zhang, Wei Li, Jie Hu, Hanting Chen, Yehui Tang, Zhiwei Xiong, Baoqun Yin, et al. Cbq: Cross-block quantization for large language models. *arXiv preprint arXiv:2312.07950*, 2023.

[12] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

[13] Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, et al. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. *arXiv preprint arXiv:2501.00321*, 2024.

[14] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *CVPR*, 2018.

[15] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. In *NeurIPS*, 2019.

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

[17] Lulu Hu, Wenhu Xiao, Xin Chen, Xinhua Xu, Bowen Xu, Kun Li, and Yongliang Tao. Masquant: Modality-aware smoothing quantization for multimodal large language models. *arXiv preprint arXiv:2603.04800*, 2026.

[18] Xing Hu, Yuan Cheng, Dawei Yang, Zhihang Yuan, Jiangyong Yu, Chen Xu, and Sifan Zhou. I-llm: Efficient integer-only inference for fully-quantized low-bit large language models. *arXiv preprint arXiv:2405.17849*, 2024.

[19] Xing Hu, Zhixuan Chen, Dawei Yang, Zukang Xu, Chen Xu, Zhihang Yuan, Sifan Zhou, and Jiangyong Yu. Moequant: Enhancing quantization for mixture-of-experts large language models via expert-balanced sampling and affinity guidance. *arXiv preprint arXiv:2505.03804*, 2025.

[20] Xing Hu, Yuan Cheng, Dawei Yang, Zukang Xu, Zhihang Yuan, Jiangyong Yu, Chen Xu, Zhe Jiang, and Sifan Zhou. Ostquant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. *arXiv preprint arXiv:2501.13987*, 2025.

[21] Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W Mahoney, and Kurt Keutzer. Squeezellm: Dense-and-sparse quantization. *arXiv preprint arXiv:2306.07629*, 2023.

[22] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

[23] Liang Li, Qingyuan Li, Bo Zhang, and Xiangxiang Chu. Norm tweaking: High-performance low-bit quantization of large language models. In *AAAI*, 2024.

[24] Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. *arXiv preprint arXiv:2411.05007*, 2024.

[25] Qingyuan Li, Yifan Zhang, Liang Li, Peng Yao, Bo Zhang, Xiangxiang Chu, Yerui Sun, Li Du, and Yuchen Xie. Fptq: Fine-grained post-training quantization for large language models. *arXiv preprint arXiv:2308.15987*, 2023.

[26] Ruiyu Li and Jiaya Jia. Visual question answering with question representation update (qru). In *NeurIPS*, 2016.

[27] Shiyao Li, Yingchun Hu, Xuefei Ning, Xihui Liu, Ke Hong, Xiaotao Jia, Xiuhong Li, Yaqi Yan, Pei Ran, Guohao Dai, et al. Mbq: Modality-balanced quantization for large vision-language models. In *CVPR*, 2025.

[28] Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. In *NeurIPS*, 2024.

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

[30] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[31] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024.

[32] Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. In *ACL*, 2024.

[33] Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. *arXiv preprint arXiv:2405.16406*, 2024.

[34] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. In *NeurIPS*, 2021.

[35] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022.

[36] Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. Affinequant: Affine transformation quantization for large language models. *arXiv preprint arXiv:2403.12544*, 2024.

[37] Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In *ICML*, 2022.

[38] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *CVPR*, 2023.

[39] Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. *arXiv preprint arXiv:2308.13137*, 2023.

[40] Sayeh Sharify, Utkarsh Saxena, Zifei Xu, Ilya Soloveychik, Xin Wang, et al. Post training quantization of large language models with microscaling formats. *arXiv preprint arXiv:2405.07135*, 2024.

[41] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, 2019.

[42] Siqing Song, Chuang Wang, Rui-Qi Wang, Yi Yang, and Xu-Yao Zhang. Achieving binary weight and activation for llms using post-training quantization. In *ACL*, 2025.

[43] Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, et al. Flatquant: Flatness matters for llm quantization. *arXiv preprint arXiv:2410.09426*, 2024.

[44] Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Q-vlm: Post-training quantization for large vision-language models. In *NeurIPS*, 2024.

[45] Haiyu Wang, Yutong Wang, Jack Jiang, and Sai Qian Zhang. Wsvd: Weighted low-rank approximation for fast and efficient execution of low-precision vision-language models. *arXiv preprint arXiv:2604.02570*, 2026.

[46] Shigeng Wang, Chao Li, Yangyuxuan Kang, Jiawei Fan, Zhonghong Ou, and Anbang Yao. Sliderquant: Accurate post-training quantization for llms. *arXiv preprint arXiv:2603.25284*, 2026.

[47] Xijun Wang, Rayyan Abdalla, Junyun Huang, Chengyuan Zhang, Ruiqi Xian, and Dinesh Manocha. Bi-vlm: Binary post-training quantization for vision-language models. In *AAAI*, 2026.

[48] Yutong Wang, Haiyu Wang, and Sai Qian Zhang. Qsvd: Efficient low-rank approximation for unified query-key-value weight compression in low-precision vision-language models. *arXiv preprint arXiv:2510.16292*, 2025.

[49] Xiuying Wei, Yunchen Zhang, Xiangguo Zhang, Ruihao Gong, Shanghang Zhang, Qi Zhang, Fengwei Yu, and Xianglong Liu. Outlier suppression: Pushing the limit of low-bit transformer language models. In *NeurIPS*, 2022.

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

[51] Jingjing Xie, Yuxin Zhang, Mingbao Lin, Liujuan Cao, and Rongrong Ji. Advancing multimodal large language models with quantization-aware scale learning for efficient adaptation. In *ACM MM*, 2024.

[52] Chen Xu, Yuxuan Yue, Zukang Xu, Xing Hu, Jiangyong Yu, Zhixuan Chen, Sifan Zhou, Zhihang Yuan, and Dawei Yang. Rwkvquant: Quantizing the rwkv family with proxy guided hybrid of scalar and vector quantization. *arXiv preprint arXiv:2505.03803*, 2025.

[53] Yufei Xue, Yushi Huang, Jiawei Shao, and Jun Zhang. Vlmq: Efficient post-training quantization for large vision-language models via hessian augmentation. *arXiv preprint arXiv:2508.03351*, 2025.

[54] Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. In *ICCV*, 2025.

[55] Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. In *NeurIPS*, 2022.

[56] Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. Zeroquant-v2: Exploring post-training quantization in llms from comprehensive study to low rank compensation. *arXiv preprint arXiv:2303.08302*, 2023.

[57] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, 2016.

[58] JiangYong Yu, Sifan Zhou, Dawei Yang, Shuoyu Li, Shuo Wang, Xing Hu, Chen Xu, Zukang Xu, Changyong Shu, and Zhihang Yuan. Mquant: Unleashing the inference potential of multimodal large language models via static quantization. In *ACM MM*, 2025.

[59] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *CVPR*, 2024.

[60] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 881–916, 2025.

[61] Ying Zhang, Peng Zhang, Mincong Huang, Jingyang Xiang, Yujie Wang, Chao Wang, Yineng Zhang, Lei Yu, Chuan Liu, and Wei Lin. Qqq: Quality quattuor-bit quantization for large language models. *arXiv preprint arXiv:2406.09904*, 2024.

[62] Weibo Zhao, Yubin Shi, Xinyu Lyu, Wanchen Sui, Shen Li, and Yong Li. Aser: activation smoothing and error reconstruction for large language model quantization. In *AAAI*, 2025.

[63] Zhen Zheng, Xiaonan Song, and Chuanjie Liu. Mixllm: Llm quantization with global mixed-precision between output-features and highly-efficient system design. *arXiv preprint arXiv:2412.14590*, 2024.

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: Our claims in the abstract and introduction are clear and supported by our experiments.

   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 the paper.

   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: [Yes]

Justification: When we introduce our method, we provide necessary assumptions.

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 information required to reproduce the main results is disclosed in experimental section and Appendix.

   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: Our implementation is publicly available at an anonymous GitHub repository. All the data used in this paper is publicly available.

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 necessary details are released in experiment section and Appendix.

   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: [Yes]

   Justification: Experimental statistical significance is considered in the experiment section.

   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: The necessary computation resources are provided in our paper.

   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: Our research conforms with this rule.

   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: [Yes]

    Justification: This information is provided in the Appendix.

    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: The models considered in the paper are all publicly available, so there are no such risks.

    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: [Yes]

    Justification: All datasets and models are properly credited in our paper.

    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: We do not release new assets in this paper.

    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 paper does not involve 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 paper does not involve with human subjects.

    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: The core method development does not involve LLMs.

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.