

# P4Q: Learning to Prompt for Quantization in Low-Bit CLIP

HUIXIN SUN, EIE, Beihang University, China

RUNQI WANG, CST, Beijing Jiaotong University, China

YANJING LI, EIE, Beihang University, China

CHANG GAO and LIPING JING, CST, Beijing Jiaotong University, China

XIAOLONG JIANG and YAO HU, Xiaohongshu Inc, China

BAOCHANG ZHANG, ASEE, Beihang University, China

XIANBIN CAO, EIE, Beihang University, China

Large-scale pre-trained Vision-Language Models (VLMs) have demonstrated remarkable performance across various visual and multimodal tasks. However, deploying these models on downstream application platforms remains challenging due to computational demands and domain gaps. Quantization offers a promising solution by significantly reducing these costs, making VLMs more feasible for deployment in such environments. There are two prevailing paradigms: Quantization-Aware Training (QAT), which preserves model performance but incurs substantial training costs; and Post-Training Quantization (PTQ), which offers greater efficiency but introduces multimodal gaps and leads to performance degradation on downstream tasks. To reduce computational costs and bridge domain gaps, we propose the “Prompt for Quantization” (P4Q) by integrating PTQ with Parameter-Efficient Fine-Tuning (PEFT) techniques. P4Q compresses model parameters and activations via PTQ, introducing learnable prompts and low-bit adapters to enhance performance on downstream tasks. The learnable prompts embed downstream knowledge to mitigate domain gaps, while the low-bit adapters realign the distributions of image and text features, thereby mitigating multimodal gaps. We also introduce a distillation loss based on cosine similarity predictions to distill the quantized model using a full-precision teacher model. Extensive experiments on thirteen datasets demonstrate that P4Q significantly enhances the performance of low-bit CLIP while reducing deployment costs. For instance, an 8-bit P4Q compressed CLIP-ViT/B-32 achieves 66.94% Top-1 accuracy on ImageNet, surpassing the prompt fine-tuned full-precision counterpart by 2.24% while reducing model size by 4 ×.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Computer vision**.

Additional Key Words and Phrases: Vision-Language Models (VLMs), CLIP, Parameter-Efficient Fine-Tuning (PEFT), Post-Training Quantization (PTQ), Prompt Learning

## ACM Reference Format:

Huixin Sun, Runqi Wang, Yanjing Li, Chang Gao, Liping Jing, Xiaolong Jiang, Yao Hu, Baochang Zhang, and Xianbin Cao. 2026. P4Q: Learning to Prompt for Quantization in Low-Bit CLIP. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1 (June 2026), 19 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

---

Authors' Contact Information: Huixin Sun, [sunhuixin@buaa.edu.cn](mailto:sunhuixin@buaa.edu.cn), EIE, Beihang University, Beijing, China; Runqi Wang, [runqiwang@buaa.edu.cn](mailto:runqiwang@buaa.edu.cn), CST, Beijing Jiaotong University, Beijing, China; Yanjing Li, [liyanjing@buaa.edu.cn](mailto:liyanjing@buaa.edu.cn), EIE, Beihang University, Beijing, China; Chang Gao, [22110098@bjtu.edu.cn](mailto:22110098@bjtu.edu.cn); Liping Jing, [lpjing@bjtu.edu.cn](mailto:lpjing@bjtu.edu.cn), CST, Beijing Jiaotong University, Beijing, China; Xiaolong Jiang, [laige@xiaohongshu.com](mailto:laige@xiaohongshu.com); Yao Hu, [xiahou@xiaohongshu.com](mailto:xiahou@xiaohongshu.com), Xiaohongshu Inc, Shanghai, China; Baochang Zhang, [bczhang@buaa.edu.cn](mailto:bczhang@buaa.edu.cn), ASEE, Beihang University, Beijing, China; Xianbin Cao, [xbcao@buaa.edu.cn](mailto:xbcao@buaa.edu.cn), EIE, Beihang University, Beijing, China.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 1551-6865/2026/6-ART

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

## 1 Introduction

Recent research in large-scale pre-trained Vision-Language Models (VLMs), such as Contrastive Language-Image Pre-Training (CLIP) [41], have achieved exceptional performance in both computer vision and natural language processing [53, 44], showcasing their ability to learn open-world concepts [23, 4]. However, deploying CLIP on downstream applications remains challenging due to limited computational resources and substantial domain gaps. For example, the CLIP-ViT/B-32 [41] contains 152M parameters and demands 53.188G FLOPs for inference under the same accounting as Table 2. Moreover, downstream applications such as transportation and medicine require the offline deployment of expert models [52, 57], which demands both high task accuracy and efficient deployment, complicating the challenge. Therefore, our goal is to achieve model compression while enabling the efficient integration of downstream task knowledge to ensure robust performance in resource-constrained and domain-shifted settings.

Quantization offers a promising solution by significantly reducing computational costs, making VLMs more feasible for deployment in resource-constrained environments. There are two prevailing paradigms: Quantization-Aware Training (QAT), which preserves model performance by simulating quantization effects during training, but incurs substantial computational overhead and requires access to large-scale training data [12, 34]. This makes QAT impractical for VLMs, which are typically pre-trained on massive, often inaccessible datasets [33, 27, 56]. On the other hand, Post-Training Quantization (PTQ) provides a more efficient alternative by avoiding full retraining and directly applying quantization to a pre-trained model [46, 24, 12]. However, PTQ introduces quantization-induced noise and fails to adapt to domain-specific features, leading to multimodal misalignment and severe accuracy degradation. For instance, a 4-bit CLIP-ViT/B-32 [41] using OMSE [9] achieves only 46.05% Top-1 accuracy on CIFAR100 [26], resulting in a 19.29% performance drop compared to the real-valued counterpart. While Parameter-Efficient Fine-Tuning (PEFT) methods [21, 55] adapt CLIP to downstream tasks with limited data and less memory usage, they retain the original model size, leaving deployment bottlenecks unresolved. Therefore, the integration of PTQ with PEFT emerges as a compelling approach to enable efficient and effective deployment of VLMs in downstream applications, particularly under resource-constrained and domain-shifted conditions.

In exploring the design of effective PEFT strategies, we observe that the performance degradation of the quantized CLIP stems from the degraded alignment of cross-modal features. As shown in Fig. 1 (a), full-precision CLIP exhibits strong diagonal similarity between matched image-text pairs. After employing a PTQ baseline [9], the image features are partially mismatched with the descriptive text features. For example, the image of the rocket erroneously matches the description “a cup of coffee on a saucer”, exhibited in Fig. 1 (b). Since there is no backpropagation in the quantization process of PTQ, it is similar to an open-loop system, *i.e.*, there is no effective cross-mode supervision on it. Moreover, PTQ quantizes the image encoder and the text encoder separately, which can cause the cross-modal gap to be further amplified due to the structural differences between CLIP encoders.

To bridge this gap, we propose “Prompt for Quantization” (P4Q), a CLIP compression framework that combines PEFT with quantization for accuracy-preserving edge deployment. As illustrated in Fig. 2, our framework introduces a learnable prompt and a novel Q-Adapter, which jointly operate in the embedding space to realign quantized image and text representations. Inspired by the effectiveness of prompt tuning in reorganizing textual representations for downstream tasks [56, 14], P4Q designs learnable prompts for the quantized CLIP, embedding downstream knowledge to mitigate domain gaps. To mitigate the degraded cross-modal similarity, a low-bit adapter named QAdapter is added after the image encoder that aligns visual features with textual counterparts. Critically, the Q-Adapter serves a dual role: it implicitly learns a distribution alignment function that compensates for quantization-induced errors, while simultaneously adapting features to the

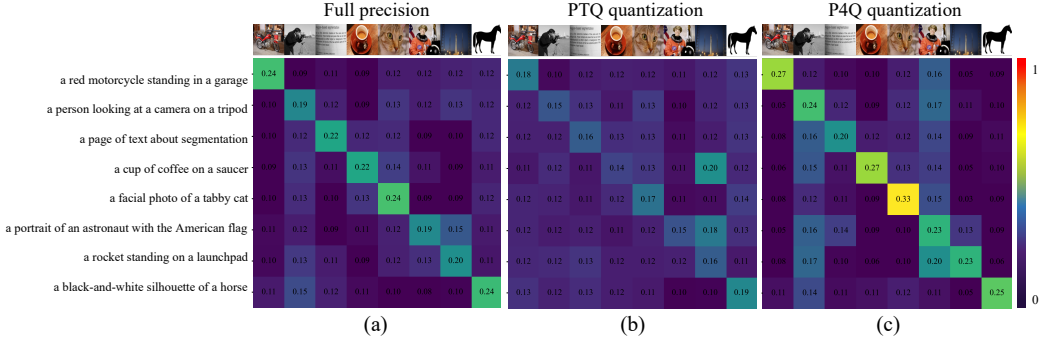


Fig. 1. Cosine similarity predictions between text and image features. The horizontal axis contains 8 images, their descriptive text is vertically given in order. Each value corresponds to the similarity between the related image features and text features. The brighter the grid, the stronger the similarity between the image and the text on the current horizontal and vertical coordinates. (a) Image features and text features are encoded by full-precision CLIP. (b) Image features and text features are encoded by PTQ quantized CLIP. (c) Image features and text features are encoded by P4Q quantized CLIP.

downstream task objective. Both the prompt and the QAdapter are optimized in quantization-awareness using a contrastive loss. Moreover, P4Q incorporates a knowledge distillation scheme to transfer alignment knowledge from full-precision CLIP based on similarity predictions, enhancing the low-bit model’s generalization. Together, these components allow P4Q to effectively bridge the deployment-critical gap between PTQ efficiency and multimodal alignment robustness, providing a principled and lightweight solution for fine-tuning low-bit CLIP without modifying the frozen quantized backbone. As shown in Fig. 1(c), P4Q successfully realigns low-bit image and text features, improving similarity accuracy.

Our contributions are summarized as follows:

- (1) We introduce “Prompt for Quantization” (P4Q), a CLIP compression framework compatible with PEFT and PTQ that enables accuracy-preserving edge deployment on the downstream application domain.
- (2) P4Q adopts a lightweight architecture in which a low-bit Q-Adapter serves dual purposes: compensating for quantization-induced distribution shifts and enabling task-specific adaptation. Together with learnable prompts and similarity-preserving distillation, the framework restores cross-modal alignment within the quantized embedding space.
- (3) Extensive experiments across thirteen datasets demonstrate that P4Q consistently outperforms the evaluated PTQ and quantized linear-probe baselines, significantly improving low-bit CLIP accuracy while reducing deployment costs.

## 2 Related Work

### 2.1 Quantization

Quantization aims to reduce the memory and computational costs of deep neural networks by converting weights and activations to low-bit representations. Two main categories are widely adopted: Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ). QAT methods require fine-tuning, making them resource-intensive. For example, [51] proposed a binarized convolutional network using projection functions and a novel backpropagation rule, while [30] improved quantized vision transformers through information rectification and distillation. Though effective, these methods are often impractical for large-scale pre-trained models due to high

computational demands. PTQ methods offer a training-free alternative. Approaches such as MinMax, EMA [22], Percentile [28], and OMSE [9] compress model weights with minimal tuning. FQ-ViT [32] further proposed a 4-bit transformer quantization using log-int softmax and bit-shift operators to simplify inference. Additionally, data-free PTQ methods generate synthetic data to support quantization without requiring access to real images [7]. Despite their efficiency, standard PTQ approaches often lead to noticeable performance drops in multi-modal and vision-language tasks.

## 2.2 PTQ in Large Language and Multimodal Models

Recent advances in large vision-language models (LVLMs) leverage both massive image-text datasets [33, 8] and the transfer learning capabilities of foundation language models [6, 42] to achieve exceptional cross-modal understanding. While text-focused quantization techniques like GPTQ [16], AWQ [31], and QLoRA [12] demonstrate effective weight compression through dynamic activation analysis, these methods exhibit two key limitations. First, optimizations primarily target textual representations, neglecting visual feature preservation. In addition, they fail to address the critical cross-modal alignment degradation in low-bit regimes. Emerging multimodal quantization approaches like Q-VLM [47] attempt structured post-training quantization through layer dependency modeling and entropy-guided partitioning. However, these methods still prioritize intra-modal accuracy over inter-modal coordination, leading to suboptimal joint embedding quality. Our work bridges this gap by synergizing Parameter-Efficient Fine-Tuning (PEFT) with precision-aware quantization objectives, enabling better preservation of vision-language alignment while maintaining computational efficiency.

## 2.3 Prompt Learning

Prompt learning originates from the NLP. The high-level idea of prompting is to apply a function to modify the input text so that the language model gets additional information about the task. Concretely, given a pre-trained language model, the task is often formulated as a “fill-in-the-blank” cloze test, such as asking the model to predict the masked token in “No reason to watch. It was [MASK]” as either “positive” or “negative” for sentiment classification. The key lies in how to design the underlined part. However, the design of a prompting function is challenging and requires heuristics. Thus, continuous prompt learning methods [54, 29] were developed, which seek to address this problem by applying trainable prompts in a continuous space. A drawback of such methods compared to searching discrete tokens is the lack of a clear way to visualize what “words” are learned for the vectors. Recent works [56, 49] adopt prompt learning to Vision-Language Models (VLMs), such as CoOp [56]. CLIP [41] is a kind of VLM, which is trained under a large number of images and their text descriptions. It adopts a two-stream architecture, consisting of an image encoder and a text encoder that encodes image and text inputs separately and produces individual vision and language representations embedded in a joint space using a contrastive loss. CoOp adopts prompt tuning to CLIP while freezing the encoders. Besides, prompt learning has been applied to many downstream tasks of VLMs such as object detection [14, 17], semantic segmentation [50], and so on. Inspired by the success of prompt learning in other tasks, this paper attempts to calibrate the distribution of the activations in low-bit CLIP by prompt learning. The experiment proves that it can improve the performance of low-bit CLIP effectively and efficiently.

## 3 Methodology

In this section, we first revisit PTQ and CLIP in Sec. 3.1. Then, we describe our “Prompt for Quantization” (P4Q) method in Sec. 3.2. The overview of our framework is given in Fig. 2. In Sec. 3.3, the optimization process is shown.

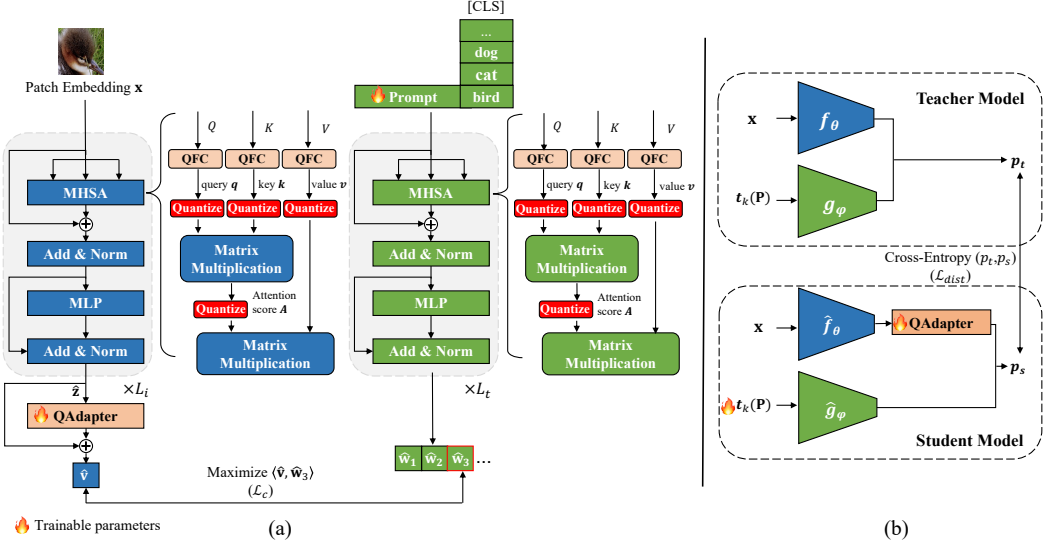


Fig. 2. Overview of P4Q. The blue parts represent the visual stream, and the green parts represent the textual stream. The learnable parameters are marked by the icon of fire. QFC denotes the quantized fully connected layer. Figure (a) shows the structure of quantized CLIP. Figure (b) shows the knowledge distillation module.  $f_\theta$  and  $g_\psi$  represent the full-precision image and text encoders respectively.  $\hat{f}_\theta$  and  $\hat{g}_\psi$  represent the low-bit image and text encoders respectively.

### 3.1 Preliminaries

**Post Training Quantization (PTQ).** We quantize tensors with bit-width  $b$  using uniform scaling; weight tensors use a signed integer grid and activation tensors an unsigned grid [22, 9]:

$$(q_{\min}, q_{\max}) = \begin{cases} (-2^{b-1}, 2^{b-1} - 1) & \text{signed (weights),} \\ (0, 2^b - 1) & \text{unsigned (activations).} \end{cases} \quad (1)$$

Uniform quantization is well supported on most hardware platforms. Given a tensor  $x$  and bit-width  $b$ , we first clip  $x$  into a calibration interval  $[l_x, u_x]$  and then map it to an integer grid  $[q_{\min}, q_{\max}]$ , where  $q_{\min} = 0$  and  $q_{\max} = 2^b - 1$  for unsigned quantization. The scale and zero-point are computed as

$$s_x = \frac{u_x - l_x}{q_{\max} - q_{\min}}, \quad zp_x = \left\lfloor q_{\min} - \frac{l_x}{s_x} \right\rfloor. \quad (2)$$

The quantized tensor is then obtained by

$$Q(x | b) = \text{clip} \left( \left\lfloor \frac{x}{s_x} \right\rfloor + zp_x, q_{\min}, q_{\max} \right). \quad (3)$$

The corresponding dequantized value is

$$\tilde{x} = s_x (Q(x | b) - zp_x). \quad (4)$$

For the OMSE baseline [9], the clipping interval is selected on calibration data by minimizing the reconstruction error between the original tensor and its dequantized counterpart:

$$(l_x^*, u_x^*) = \arg \min_{l_x, u_x} \|x - \tilde{x}(l_x, u_x, b)\|_2^2. \quad (5)$$

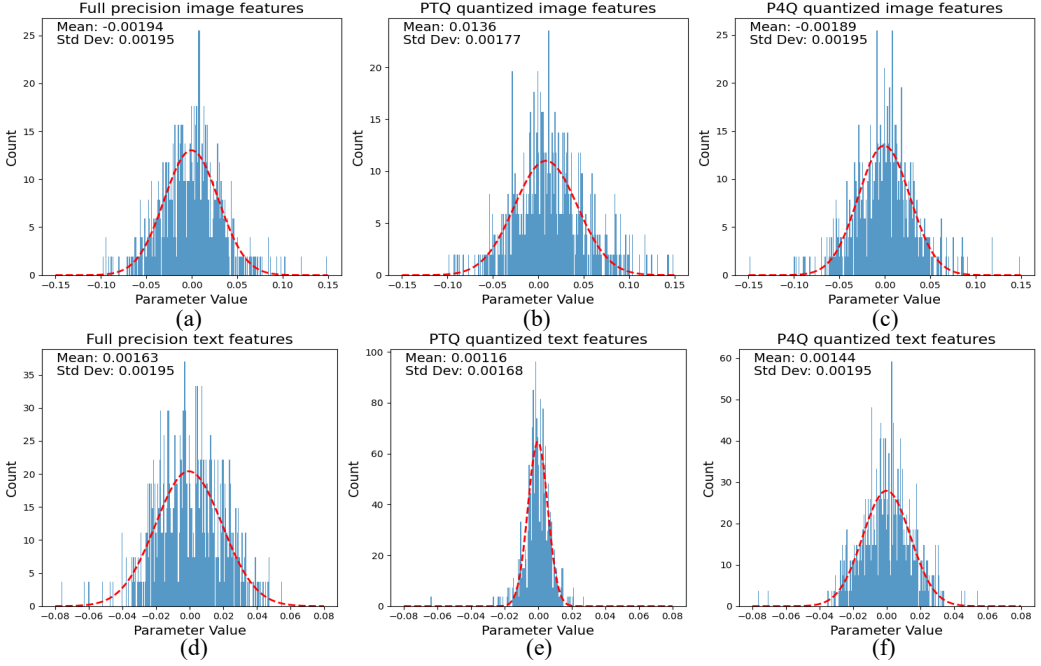


Fig. 3. Histogram of image features and text features of the same class ('wolf') on CIFAR100, where the horizontal axis denotes the value of each element of the feature vectors, and the vertical axis denotes the number of elements. Mean and Ste Dev denote the mean and standard deviation of the distribution, respectively.

After obtaining  $(I_x^*, u_x^*)$ , the corresponding  $s_x$  and  $zp_x$  are computed by the equations above. This separates the uniform quantizer definition from the calibration rule and follows the minimum-MSE principle of OMSE. **CLIP** [41]. CLIP consists of an image encoder  $f_\theta(\cdot)$  and a text encoder  $g_\psi(\cdot)$ . Specifically, the image  $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$  and the text  $\mathbf{t} \in \mathbb{R}^D$  are fed into  $f_\theta(\cdot)$  and  $g_\psi(\cdot)$  respectively to obtain the image feature  $\mathbf{z} \in \mathbb{R}^D$  and the text feature  $\mathbf{w} \in \mathbb{R}^D$ , where  $\mathbf{t}$  is the input word token. In CLIP,  $\mathbf{t}$  is obtained via one of the hand-crafted prompts which have a template like "a photo of a [CLS]", where [CLS] is the class name of the testing image. Thus, the probability of predicting the testing image  $\mathbf{x}$  as the class  $y_i$  can be computed as:

$$p(y_i|\mathbf{x}) = \frac{e^{\langle \mathbf{z}, \mathbf{w}_{y_i} \rangle / \tau}}{\sum_{k=1}^K e^{\langle \mathbf{z}, \mathbf{w}_k \rangle / \tau}}, \quad (6)$$

where  $\tau$  is a temperature parameter learned by CLIP,  $\langle \cdot, \cdot \rangle$  denotes the Cosine similarity,  $\mathbf{w}_k$  is the feature derived from  $\mathbf{t}_k$  of the  $k$ -th class, and  $K$  is the total number of downstream dataset classes.

### 3.2 Prompt for Quantization

CLIP is challenging to quantify by QAT due to its significant training burden and non-public pre-training dataset. Furthermore, naive PTQ causes significant quantization error and performance deterioration of CLIP from full-precision to low-bit. As shown in Fig. 3 (a) and (b), the distribution of image features is disturbed by PTQ. Fig. 3 (d) and (e) indicate the distortion of text features is even more serious. To alleviate the quantization error, we must calibrate the output features of the

low-bit model. Since prompt tuning is effective and efficient, we suggest Prompt for Quantization (P4Q) to calibrate the output features of the low-bit CLIP quantized by PTQ. The P4Q details are introduced following.

Following CLIP, we choose ViT-B/32 [13] and a text transformer as the image encoder and text encoder with the fixed weights. As shown in Fig. 2 (a), the image encoder and text encoder contain  $L_i$  and  $L_t$  blocks respectively. The self-attention module in each block is conducted quantization following PTQ in sec. 3.1. It should be noted that there is no weights training in PTQ, *i.e.*, the pre-trained full-precision weights of the encoders are directly quantized to low-bit weights. To relieve the distribution bias caused by quantization, we replace the hand-crafted prompt templates with a set of continual trainable vectors named learnable prompt  $\mathbf{P}$  in the quantized textual stream. Specifically,  $\mathbf{P}$  are concatenated with the embedding of a class name, formulating the text description  $\mathbf{t}_k(\mathbf{P})$  of the  $k$ -th class as:

$$\mathbf{t}_k(\mathbf{P}) = [\mathbf{p}]_1 [\mathbf{p}]_2 \dots [\mathbf{p}]_M [\text{CLS}]_k, \quad (7)$$

where each  $[\mathbf{p}]_m \in \mathbb{R}^D$ ,  $m \in \{1, \dots, M\}$ , is a learnable token of  $\mathbf{P}$ , and  $\mathbf{P} \in \mathbb{R}^{D \times M}$  is shared among all classes.  $[\text{CLS}]_k$  is the text embedding of the  $k$ -th class name, which can also appear at the start and middle of the prompt. The text description is quantized before inputting the self-attention module.

Then the unsigned quantized text encoder produces a text feature  $\hat{\mathbf{w}}_k$ , which is formulated as:

$$\hat{\mathbf{w}}_k = \text{clip}(\lfloor \frac{\hat{g}_\psi(\mathbf{t}_k(\mathbf{P}))}{s_w} \rfloor + zp_w, 0, 2^b - 1), \quad (8)$$

where  $\hat{g}_\psi$  represents the quantized text encoder.  $s_w$  and  $zp_w$  are determined by full-precision  $\mathbf{w}_k$ , the calculation of which can be refer to sec. 3.1. CLIP employs a two-stream structure, requiring the output feature of the image encoder to align with the output feature of the text encoder. Since the weights of the image encoder are fixed, the feature of the same image remains constant. It is difficult to change the distribution of image feature even if prompt is learnable.

Therefore, learnable parameters are in need in the image stream to jointly calibrate the image feature and the text feature. As shown in Fig. 2 (a), we attach a low-bit module, namely QAdapter, in the image stream. The QAdapter can project the low-bit image feature  $\hat{\mathbf{z}}$  to the appropriate distribution based on the text feature  $\hat{\mathbf{w}}_k$ . The feature  $\hat{\mathbf{z}}$  is formulated as:

$$\hat{\mathbf{z}} = Q(\hat{f}_\theta(\mathbf{x})|b) = \text{clip}(\lfloor \frac{\hat{f}_\theta(\mathbf{x})}{s_z} \rfloor + zp_z, 0, 2^b - 1), \quad (9)$$

where  $\hat{f}_\theta$  represents the image encoder containing the quantization module.  $s_z$  and  $zp_z$  are determined by full-precision  $\mathbf{z}$ .

Fig. 4 sketches QAdapter, including two low-bit fully connected layers (QFC) in a bottleneck with a shortcut. The feature  $\hat{\mathbf{v}}$  calculated by QAdapter is formulated as:

$$\hat{\mathbf{v}} = \alpha \hat{h}_2(Q(\hat{h}_1(\hat{\mathbf{z}}|b))) + (1 - \alpha)\hat{\mathbf{z}} \quad (10)$$

where  $\hat{h}_1$  and  $\hat{h}_2$  represent the two QFC layers and  $\alpha$  denotes the adaptation ratio. To simplify the function, we omit the ReLU activation function. We place QAdapter after the quantized image encoder rather than inside intermediate transformer blocks. This post-encoder placement directly calibrates the final visual feature used for image-text similarity while preserving the frozen PTQ backbone. Inserting adapters into intermediate quantized layers would change the activation distribution of downstream layers and require additional calibration.

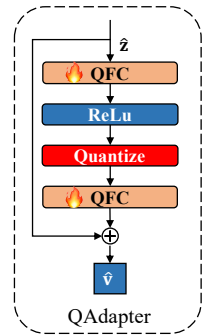


Fig. 4. Structure of QAdapter.

The similarity prediction between the image  $\mathbf{x}_i$  and the text description of class  $y_i$  is finally computed as:

$$p(y_i|\mathbf{x}_i) = \frac{e^{\langle \hat{\mathbf{v}}_i, \hat{\mathbf{w}}_{y_i} \rangle / \tau}}{\sum_{k=1}^K e^{\langle \hat{\mathbf{v}}_i, \hat{\mathbf{w}}_k \rangle / \tau}}. \quad (11)$$

### 3.3 Optimization Process of P4Q

The key to P4Q is to provide an efficient training paradigm for PTQ. It retains the high efficiency of PTQ and introduces a learnable prompt and QAdapter to improve the quantization performance. The purpose of these learnable parameters is to realign the distributions of image and text features. In this subsection, a joint loss function in supervising prompt and adapter training are detailed below.

Following the loss function of CLIP, the Cosine similarity of the quantized image feature and text feature is adopted as the classification loss  $\mathcal{L}_c$ . It can directly affect the distribution alignment of images and text features, which is formulated as:

$$\mathcal{L}_c = \mathbb{E}[-\log \frac{e^{\langle \hat{\mathbf{v}}_i, \hat{\mathbf{w}}_{y_i} \rangle / \tau}}{\sum_{k=1}^K e^{\langle \hat{\mathbf{v}}_i, \hat{\mathbf{w}}_k \rangle / \tau}}]. \quad (12)$$

To address the distributions of image and text features mismatch that occurred in the quantized CLIP, the distributions of image and text features of the full-precision model can be used as a reference. We introduce a knowledge distillation module that uses similarity predictions of the full-precision teacher as soft targets for the quantized student. Knowledge distillation can help reduce the difficulty of optimizing the additional parameters in the quantized model and determine the optimization direction for the cross-modal alignment. The distillation loss is the cross-entropy with teacher probabilities, equivalently the KL term  $\text{KL}(p_t \| p_s)$  up to an additive constant:

$$\mathcal{L}_{\text{dist}} = -\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K p_t(k | \mathbf{x}_i) \log p_s(k | \mathbf{x}_i), \quad (13)$$

where  $p_t(k | \mathbf{x}_i)$  is the teacher softmax over  $K$  classes computed as in Eq. (6) using full-precision image features  $\mathbf{z}_i$  and text prototypes  $\mathbf{w}_k$ , and  $p_s(k | \mathbf{x}_i)$  is the student distribution defined analogously from  $(\hat{\mathbf{v}}_i, \hat{\mathbf{w}}_k)$  as in Eq. (11). The batch size is  $n$ , as in Fig. 2 (b).

Different from prior work [58] that needs to match the outputs from every intermediate block, or further using multi-step progressive structural transition [36], we find that our distillation loss, while much simpler, can yield competitive results. The overall training losses for our approach are:

$$\mathcal{L} = \mathcal{L}_c + \lambda \mathcal{L}_{\text{dist}}, \quad (14)$$

where  $\lambda$  is a trade-off hyper-parameter. The prompt and QAdapter trained by  $\mathcal{L}$ , calibrate the text features and image features effectively. As shown in Fig. 3 (c) and (f), the distributions of P4Q quantized image and text features are moved to their full-precision counterparts and realigned.

## 4 Experiments

In this section, we first evaluate the performance of the proposed P4Q method on the **image classification** task in Secs. 4.2–4.4. We then examine its generalization abilities in **domain generalization** in Sec. 4.5, as explored in previous works [56, 55]. Extensive results in image classification reveal that P4Q surpasses the linear probe fine-tuned low-bit comparisons, forming a lighter model with full-precision comparable performance. Results in domain generalization experiments demonstrate that P4Q achieves higher zero-shot accuracy on target datasets than the quantized baselines, capturing generalizable features that remain effective in unsupervised downstream datasets.

Table 1. Comparison of the accuracy with PTQ models on CIFAR100 dataset. #Bits (W-A-Attention) denotes the bit-width of weights, activations, and attention activations (query, key, value, and attention weights). The best results are bold.

| Backbone | Method                   | #Bits    | Top-1        | Top-5        | FLOPs(G) | Size(MB) |
|----------|--------------------------|----------|--------------|--------------|----------|----------|
| ViT-B/32 | Zero Shot CLIP           | 32-32-32 | 65.34        | 89.00        | 53.188   | 577.08   |
|          | + Prompt                 | 32-32-32 | 76.51        | 94.97        | 53.188   | 577.08   |
|          | + Linear Probe           | 32-32-32 | 80.50        | -            | 53.188   | 577.28   |
|          | MinMax                   | 8-8-8    | 64.70        | 88.85        | 13.297   | 146.95   |
|          | EMA [22]                 | 8-8-8    | 54.45        | 83.56        | 13.297   | 146.95   |
|          | Percentile [28]          | 8-8-8    | 59.51        | 85.88        | 13.297   | 146.95   |
|          | Baseline [9]             | 8-8-8    | 64.48        | 88.88        | 13.297   | 146.95   |
|          | + Quantized Linear Probe | 8-8-8    | 79.03        | -            | 13.297   | 147.45   |
|          | + <b>P4Q</b>             | 8-8-8    | <b>79.42</b> | <b>95.49</b> | 13.297   | 147.08   |
|          | Baseline [9]             | 4-4-8    | 46.05        | 74.47        | 13.271   | 94.76    |
|          | + Quantized Linear Probe | 4-4-8    | 66.11        | -            | 13.271   | 94.78    |
|          | + <b>P4Q</b>             | 4-4-8    | <b>69.20</b> | <b>91.26</b> | 13.271   | 94.89    |
|          | Baseline [9]             | 3-3-8    | 16.79        | 38.68        | 13.266   | 81.96    |
|          | + Quantized Linear Probe | 3-3-8    | 30.53        | -            | 13.266   | 81.98    |
|          | + <b>P4Q</b>             | 3-3-8    | <b>47.83</b> | <b>77.66</b> | 13.266   | 82.09    |
|          | Baseline [9]             | 2-2-8    | 5.89         | 19.48        | 13.260   | 68.67    |
|          | + Quantized Linear Probe | 2-2-8    | 9.20         | -            | 13.260   | 68.68    |
|          | + <b>P4Q</b>             | 2-2-8    | <b>31.09</b> | <b>62.24</b> | 13.260   | 68.80    |
| ViT-L/14 | Zero Shot CLIP           | 32-32-32 | 78.28        | 94.02        | 167.734  | 1631.23  |
|          | + Prompt                 | 32-32-32 | 83.65        | 96.64        | 180.673  | 1631.23  |
|          | + Linear Probe           | 32-32-32 | 89.60        | -            | 170.564  | 1631.23  |
|          | Baseline [9]             | 8-8-8    | 78.16        | 94.14        | 41.933   | 413.94   |
|          | + Quantized Linear Probe | 8-8-8    | <b>85.44</b> | -            | 41.933   | 413.94   |
|          | + <b>P4Q</b>             | 8-8-8    | 85.41        | 97.20        | 41.933   | 413.94   |
|          | Baseline [9]             | 4-4-8    | 69.49        | 89.31        | 41.861   | 272.84   |
|          | + Quantized Linear Probe | 4-4-8    | 78.14        | -            | 41.861   | 272.84   |
|          | + <b>P4Q</b>             | 4-4-8    | <b>80.09</b> | 95.42        | 41.861   | 272.84   |
|          | Baseline [9]             | 3-3-8    | 49.72        | 73.74        | 41.843   | 237.88   |
|          | + Quantized Linear Probe | 3-3-8    | 57.77        | -            | 41.843   | 237.88   |
|          | + <b>P4Q</b>             | 3-3-8    | <b>66.74</b> | 89.49        | 41.843   | 237.88   |
|          | Baseline [9]             | 2-2-8    | 21.71        | 42.05        | 41.824   | 201.87   |
|          | + Quantized Linear Probe | 2-2-8    | 19.69        | -            | 41.824   | 201.87   |
|          | + <b>P4Q</b>             | 2-2-8    | <b>42.81</b> | 72.03        | 41.824   | 201.87   |

#### 4.1 Datasets and Implementation Details

**Datasets.** Main experiments for the image classification task are carried out on the CIFAR100 [26] and ImageNet-1k [11] datasets. The CIFAR-100 dataset consists of 60,000 images of size  $32 \times 32$  from 100 classes. Each class contains 500 training images and 100 testing images, and the 100 fine-grained classes are grouped into 20 superclasses. We use the standard fine-label classification protocol. The ImageNet-1k dataset contains samples sized  $224 \times 224$  from 1000 classes. Each class consists of about 1,300 training and 50 test samples. Following [55], we further utilize ten downstream datasets to examine the robustness of P4Q. These datasets include one repository for generic object classification Caltech101 [15], and five repositories for fine-grained classification: OxfordPets [38],

Table 2. Comparison of the accuracy with PTQ models on the ImageNet dataset. #Bits (W-A-Attention) denotes the bit-width of weights, activations, and attention activations (query, key, value, and attention weights).

| Backbone | Method                   | #Bits    | Top-1        | Top-5        | FLOPs(G) | Size(MB) |
|----------|--------------------------|----------|--------------|--------------|----------|----------|
| ViT-B/32 | Zero Shot CLIP           | 32-32-32 | 63.36        | 88.82        | 53.188   | 577.08   |
|          | + Prompt                 | 32-32-32 | 64.70        | 88.85        | 55.609   | 577.08   |
|          | + Linear Probe           | 32-32-32 | 76.10        | -            | 54.907   | 577.08   |
|          | MinMax                   | 8-8-8    | 63.14        | 88.78        | 13.297   | 146.95   |
|          | EMA [22]                 | 8-8-8    | 41.51        | 71.28        | 13.297   | 146.95   |
|          | Percentile [28]          | 8-8-8    | 46.48        | 73.96        | 13.297   | 146.95   |
|          | Baseline [9]             | 8-8-8    | 64.48        | 88.88        | 13.297   | 146.95   |
|          | + Quantized Linear Probe | 8-8-8    | 66.90        | -            | 13.297   | 147.45   |
|          | + <b>P4Q</b>             | 8-8-8    | <b>66.94</b> | <b>92.50</b> | 13.297   | 147.08   |
|          | Baseline [9]             | 4-4-8    | 56.96        | 82.98        | 13.271   | 94.76    |
|          | + Quantized Linear Probe | 4-4-8    | 58.80        | -            | 13.271   | 94.78    |
|          | + <b>P4Q</b>             | 4-4-8    | <b>61.97</b> | <b>86.22</b> | 13.271   | 94.89    |
|          | Baseline [9]             | 3-3-8    | 45.18        | 73.85        | 13.266   | 81.96    |
|          | + Quantized Linear Probe | 3-3-8    | 49.40        | -            | 13.266   | 81.98    |
|          | + <b>P4Q</b>             | 3-3-8    | <b>53.62</b> | <b>83.17</b> | 13.266   | 82.09    |
|          | Baseline [9]             | 2-2-8    | 15.32        | 33.73        | 13.260   | 68.67    |
|          | + Quantized Linear Probe | 2-2-8    | 19.40        | -            | 13.260   | 68.68    |
|          | + <b>P4Q</b>             | 2-2-8    | <b>25.19</b> | <b>56.12</b> | 13.260   | 68.80    |
| ViT-L/14 | Zero Shot CLIP           | 32-32-32 | 75.54        | 94.57        | 167.734  | 1631.23  |
|          | + Prompt                 | 32-32-32 | 76.90        | 95.40        | 180.673  | 1631.23  |
|          | + Linear Probe           | 32-32-32 | 79.05        | -            | 170.564  | 1631.23  |
|          | Baseline [9]             | 8-8-8    | 73.98        | 93.91        | 41.933   | 413.94   |
|          | + Quantized Linear Probe | 8-8-8    | 74.67        | -            | 41.933   | 413.94   |
|          | + <b>P4Q</b>             | 8-8-8    | <b>77.69</b> | <b>96.05</b> | 41.933   | 414.22   |
|          | Baseline [9]             | 4-4-8    | 69.95        | 91.58        | 41.861   | 272.84   |
|          | + Quantized Linear Probe | 4-4-8    | 70.23        | -            | 41.861   | 272.84   |
|          | + <b>P4Q</b>             | 4-4-8    | <b>75.56</b> | <b>92.85</b> | 41.861   | 273.12   |
|          | Baseline [9]             | 3-3-8    | 66.43        | 90.59        | 41.843   | 237.88   |
|          | + Quantized Linear Probe | 3-3-8    | 68.97        | -            | 41.843   | 237.88   |
|          | + <b>P4Q</b>             | 3-3-8    | <b>74.23</b> | <b>92.52</b> | 41.843   | 238.16   |
|          | Baseline [9]             | 2-2-8    | 35.34        | 61.53        | 41.824   | 201.87   |
|          | + Quantized Linear Probe | 2-2-8    | 33.27        | -            | 41.824   | 201.87   |
|          | + <b>P4Q</b>             | 2-2-8    | <b>42.05</b> | <b>63.40</b> | 41.824   | 202.15   |

StanfordCars [25], Flowers102 [37], Food101 [5], and FGVC Aircraft [35]. Furthermore, we include SUN397 for scene recognition [40], UCF101 for action recognition [45], DTD for texture classification [10], and EuroSAT for satellite imagery recognition [18]. In domain generalization experiments, we train a 4-bit P4Q using all data samples from the ImageNet and test the zero-shot performance on ImageNet-A [19], ImageNet-R [20], ImageNet-V2 [43], ImageNet-Sketch [48].

**Baseline and Comparisons.** To the best of our knowledge, there is no publicly available source code on Post-Training Quantization (PTQ) of CLIP at this point, so we implement a PTQ baseline using OMSE [9]. The baseline employs signed channel-wise quantization for weights and unsigned layer-wise quantization for activations following Eq. 1. We randomly sample 10 iterations of training

images as the calibration data to solve Eq. 5 and obtain scales and zero-points via Eqs. 2–4. We include some conventional PTQ methods as low-bit comparisons, including MinMax, EMA [22] and Percentile [28]. Building on these PTQ baselines, we further validate the effectiveness of our P4Q framework, ensuring that performance improvements stem specifically from our proposed Q-Adapter distribution alignment and prompt-based calibration. We also introduce a supervised low-bit comparison termed Quantized Linear Probe. Here, the linear classifier parameters are quantized using MinMax, while all other parameters follow the same OMSE-based PTQ baseline described above.

**Experimental settings.** In the experiments, we choose ViT-B/32 [13] and a text transformer as image and text encoder, both initialized with pre-trained full-precision CLIP weights with  $L_i = L_t = 12$ . Learnable prompt tokens use  $M=16$ , are drawn from  $\mathcal{N}(0, 0.02^2)$  at initialization, and are trained with SGD at learning rate  $5 \times 10^{-4}$  and weight decay 0. QAdapter uses 8-bit QFC layers, adaptation ratio  $\alpha=0.2$ , Kaiming-uniform weight initialization and zero bias initialization, and AdamW with initial learning rate  $10^{-3}$ . The distillation weight is  $\lambda=1$ .

We train P4Q for 50 epochs in the main experiments. For the ImageNet-based prompt-tuning setting, this follows the 50-epoch ImageNet setting used in CoOp [56]. The train, validation, and calibration batch size is set to 128. We explore two adaptation methods on full-precision models: Linear Probe, which adopts a linear classifier on CLIP image features, and Prompt, which employs learnable prompt on full-precision models as detailed in Sec. 3.2, trained for 50 epochs. We implement these methods using PyTorch [39] on 4 NVIDIA 3090Ti GPUs with 24 GB memory.

## 4.2 Results on CIFAR-100

We evaluate P4Q against low-bit PTQ baselines and a quantized linear probe on CIFAR-100. Table 1 summarizes the results for CLIP-ViT-B/32 and CLIP-ViT-L/14. For ViT-B/32, standard PTQ baselines—including MinMax, EMA [22], Percentile [28], and the OMSE baseline—perform poorly in low-bit settings; for example, top-1 accuracy reaches only 5.89% in the 2-bit setting. In contrast, P4Q recovers substantial accuracy, lifting the 2/3/4/8-bit baseline by +25.20%, +31.04%, +23.15%, and +14.94%, respectively. Compared to the quantized linear probe, P4Q gains +21.89%, +17.3%, +3.09%, and +0.39% at 2-bit, 3-bit, 4-bit, and 8-bit, respectively. 8-bit P4Q reaches 79.42% top-1 accuracy, exceeding the full-precision prompt-tuned model by 2.91% and matching full-precision linear probe performance. For the larger ViT-L/14, P4Q outperforms the PTQ baseline by large margins, with gains of +21.10, +17.02, +10.60, and +7.25 percentage points at 2-bit, 3-bit, 4-bit, and 8-bit, respectively. The 2-bit variant attains 42.81% top-1 accuracy, while 3-bit, 4-bit, and 8-bit versions achieve 66.74%, 80.09%, and 85.41%, respectively. Relative to the quantized linear probe, P4Q is higher at 2/3/4-bit by +23.12, +8.97, and +1.95 percentage points, respectively, and lower at 8-bit by 0.03 percentage points, i.e., 85.41% versus 85.44%. For instance, 4-bit P4Q compresses the model size by 6.1 $\times$ , reduces FLOPs by 4.0 $\times$ , and still delivers 69.20% top-1 accuracy on ViT-B/32, forming a highly efficient yet accurate variant suitable for edge deployment.

## 4.3 Results on ImageNet-1k

We further evaluate P4Q on the large-scale ImageNet-1k benchmark. Results are summarized in Table 2. For ViT-B/32, standard PTQ baselines exhibit a sharp accuracy drop in low-bit settings; for example, top-1 reaches only 15.32% in the 2-bit setting. P4Q recovers substantial performance, improving the 2/3/4/8-bit baseline by +9.87, +8.44, +5.01, and +2.46 percentage points, respectively. Compared to the quantized linear probe, P4Q improves Top-1 by +5.79, +4.22, +3.17, and +0.04 percentage points at 2-bit, 3-bit, 4-bit, and 8-bit, respectively. 8-bit P4Q reaches 66.94% top-1 accuracy, outperforming the full-precision prompt-tuned model by 2.24%. With the larger ViT-L/14 model, P4Q also surpasses the quantized linear probe in Top-1 accuracy across all bit-widths,

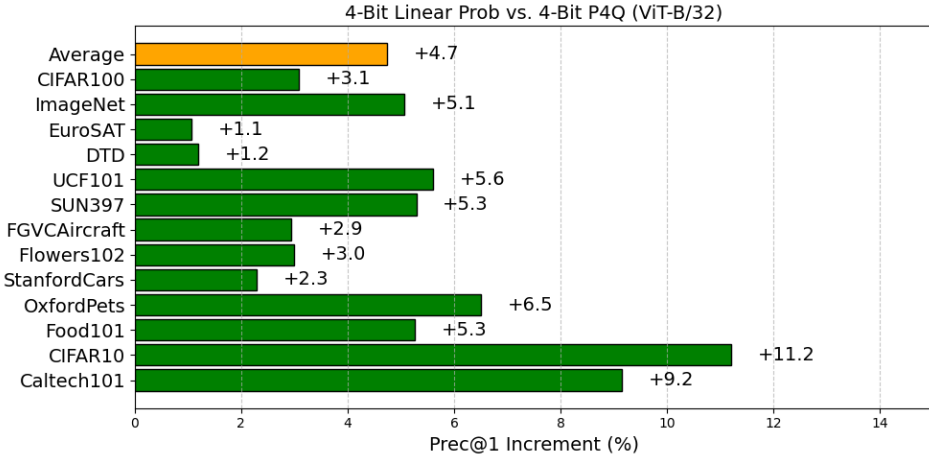


Fig. 5. Comprehensive comparisons of 4-bit Linear Probe and P4Q across 13 datasets using the ViT-B/32 backbone demonstrate that P4Q consistently outperforms Linear Probe in all datasets, with an average improvement of 4.7%.

with margins of +8.78, +5.26, +5.33, and +3.02 percentage points at 2-bit, 3-bit, 4-bit, and 8-bit, respectively. The 2-bit variant attains 42.05% Top-1 accuracy, while 3-bit, 4-bit, and 8-bit versions achieve 74.23%, 75.56%, and 77.69%, respectively. These results confirm that P4Q maintains robust accuracy on large-scale vision tasks while enabling significant model compression and inference acceleration, demonstrating its scalability and practicality for real-world deployment.

#### 4.4 Results on Diverse Datasets

To further evaluate the robustness and generalizability of P4Q, we benchmark it against 4-bit linear probe across thirteen diverse image classification datasets. As illustrated in Fig. 5, P4Q consistently surpasses the performance of standard 4-bit linear probing on all datasets, achieving an average improvement of +4.7% in Top-1 accuracy. Notably, P4Q delivers substantial gains in several datasets with challenging characteristics: +11.2% on CIFAR10, +9.2% on Caltech101, and +6.5% on OxfordPets. Mid-level improvements are observed on datasets such as ImageNet at +5.1%, UCF101 at +5.6%, and SUN397 at +5.3%, highlighting the ability of P4Q to generalize well to both object- and scene-centric tasks. Smaller yet consistent improvements are also present in fine-grained datasets such as FGVCAircraft at +2.9%, Flowers102 at +3.0%, and StanfordCars at +2.3%, as well as domain-specific datasets such as EuroSAT at +1.1% and DTD at +1.2%. These consistent improvements across datasets of varying complexity and domain confirm the effectiveness of P4Q in enhancing the performance of low-bit CLIP. By significantly improving performance while reducing deployment cost, P4Q proves to be a robust solution for deploying multimodal models in resource-constrained environments.

#### 4.5 Domain Generalization

In domain generalization experiments, we train a 4-bit P4Q using all data samples from the ImageNet-1k train and test the zero-shot performance on four out-of-distribution datasets. As shown in Table 3, P4Q demonstrates consistent improvements over the quantized baselines. For instance, the 10-epoch and 20-epoch 4-bit P4Q models improve ImageNet-V2 accuracy by 1.47 and 1.62 percentage points over the PTQ baseline of 50.45%, respectively, indicating that P4Q captures

Table 3. Comparison of generalization performance in the domain generalization setting. P4Q achieves higher accuracy on downstream target datasets than the PTQ baseline. In contrast, the prompt learning comparison lowers the baseline by a consistent margin, demonstrating a risk in overfitting base dataset classes. For the Quantized Linear Probe baseline, only the source-domain ImageNet Top-1 is reported, since the linear classifier is trained on the source domain and does not zero-shot transfer to disjoint label spaces.

| Method                   | #Bits    | Source<br>ImageNet | Target Datasets |              |              |              |
|--------------------------|----------|--------------------|-----------------|--------------|--------------|--------------|
|                          |          |                    | -A              | -R           | -V2          | -Sketch      |
| Zero Shot CLIP           | 32-32-32 | 63.36              | 31.54           | 69.36        | 56.72        | 41.30        |
| Baseline [9]             | 4-4-8    | 56.96              | 22.64           | 66.48        | 50.45        | 33.04        |
| + Quantized Linear Probe | 4-4-8    | 58.80              | -               | -            | -            | -            |
| + Prompt                 | 4-4-8    | 58.50              | 22.12           | 65.78        | 50.12        | 32.65        |
| + P4Q (10 epochs)        | 4-4-8    | 59.63              | 22.62           | <b>66.56</b> | 51.92        | <b>34.25</b> |
| + P4Q (20 epochs)        | 4-4-8    | 61.21              | <b>22.65</b>    | 66.52        | <b>52.07</b> | 33.40        |
| + P4Q (50 epochs)        | 4-4-8    | 61.97              | 22.62           | <b>66.56</b> | 50.65        | 32.92        |

generalizable features that remain effective in unsupervised domains. In contrast, prompt learning shows a consistent decline in performance, such as a drop to 22.12% on ImageNet-A and 50.12% on ImageNet-V2, indicating a risk of overfitting to the source dataset compared to manual prompts. The generalization capabilities of low-bit P4Q models can be attributed to two factors. For one, the PTQ approach for quantizing the CLIP encoders minimizes alterations to the pre-trained parameters. In contrast, QAT methods risk overfitting the quantized model to downstream datasets. Additionally, the distillation process of the P4Q model leverages logits from its full-precision counterpart, derived from the pre-trained model’s performance on the source dataset. This method ensures the retention of critical prior knowledge from the pre-trained model, contributing to the preservation of generalization performance on unsupervised target datasets.

Meanwhile, we observe that the training duration can affect the transfer performance of P4Q. The 50-epoch schedule follows the ImageNet setting used in CoOp and is used as the default for the main in-domain comparison. However, shorter schedules can be preferable for some transfer settings. For example, the 20-epoch model achieves the best ImageNet-V2 result, while the 10-epoch model achieves the best ImageNet-Sketch result in Table 3. This phenomenon aligns with prior research where supervised adaptation methods on CLIP, such as linear probe [41] and prompt tuning [56], often lead to improved training set performance at the cost of reduced generalizability.

#### 4.6 Convergence Analysis

In Fig. 6, we present the Top-1 accuracy trends of P4Q across varying quantization configurations at 2-bit, 3-bit, 4-bit, and 8-bit over training epochs, evaluated on five datasets: CIFAR-100, CIFAR-10, Caltech101, Food101, ImageNet, and a combined average. Across all bit widths and datasets, P4Q consistently outperforms the PTQ baseline, demonstrating its robust effectiveness in low-bit regimes. Notably, P4Q achieves substantial gains even within the first five epochs. The phenomenon is especially significant in ultra-low-bit settings such as 2-bit, where baseline models suffer from severe performance degradation. Additionally, we observe stable convergence behavior across all configurations. As training progresses, the performance gap between different bit settings becomes more pronounced, reflecting P4Q’s ability to fully exploit the representational capacity of each bit configuration. The 4-4-8 and 8-8-8 variants nearly match full-precision performance across datasets, exhibiting strong performance. These results suggest that the P4Q not only enhances the low-bit model’s performance efficiently but also ensures stable training, making it a compelling choice for deployment in resource-constrained environments.

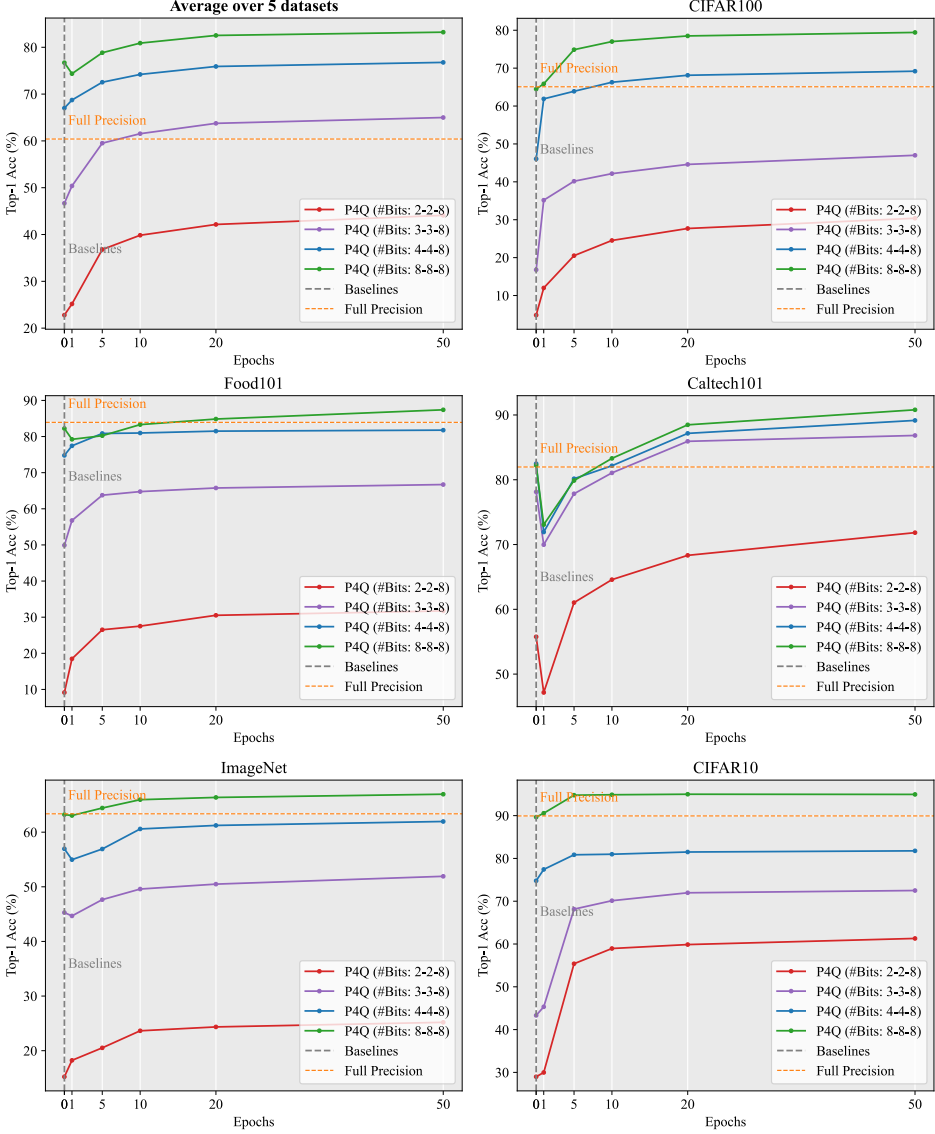


Fig. 6. Top-1 Accuracy over epochs on the 5 datasets. P4Q on zero epoch represent the baseline model of the corresponding bit type. The yellow dashed lines indicate the Zero Shot CLIP performance on each dataset.

## 4.7 Ablation Study

In the following experiments, we explore the best-performed structure of P4Q by ablating and tuning its parts. All tests are performed on CIFAR100 dataset.

**4.7.1 Component Analysis.** Table 4 ablates prompt tuning and QAdapter on CIFAR-100 under the 4-4-8 setting. Prompt tuning alone improves Top-1 accuracy from 46.05% to 63.89%, while the visual QAdapter alone reaches 66.28%. Their combination, namely P4Q, further improves the result to 69.20%, outperforming the visual QAdapter alone by 2.92 percentage points and matching

Table 4. Component Analysis of P4Q on CIFAR-100. Bold indicates best quantized performance. FLOPs measured in billions (G), model size in megabytes (MB).

| Method                    | #Bits    | Top-1 (%)    | FLOPs (G) | Size (MB) |
|---------------------------|----------|--------------|-----------|-----------|
| Full Precision            | 32-32-32 | 65.34        | -         | -         |
| Baseline [9]              | 4-4-8    | 46.05        | 13.271    | 94.76     |
| + Prompt                  | 4-4-8    | 63.89        | 13.271    | 94.76     |
| + QAdapter                | 4-4-8    | 66.28        | 13.271    | 94.89     |
| + QAdapter-text           | 4-4-8    | 60.50        | 13.271    | 94.76     |
| + QAdapter-text, QAdapter | 4-4-8    | 65.07        | 13.271    | 94.89     |
| + <b>P4Q</b>              | 4-4-8    | <b>69.20</b> | 13.271    | 94.89     |

Table 5. Evaluating the losses of P4Q. Bold indicates best performance.

| Method                        | $\lambda$ | Epochs | #Bits    | Top-1 (%)    |
|-------------------------------|-----------|--------|----------|--------------|
| Full Precision                | -         | -      | 32-32-32 | 65.34        |
| Baseline [9]                  | -         | -      | 4-4-8    | 46.05        |
| + $\mathcal{L}_{\text{dist}}$ | -         | 20     | 4-4-8    | 64.50        |
| + $\mathcal{L}_c$             | -         | 20     | 4-4-8    | 66.34        |
| + P4Q                         | 0.5       | 20     | 4-4-8    | 67.80        |
|                               | 1.0       | 20     | 4-4-8    | <b>68.12</b> |
|                               | 1.5       | 20     | 4-4-8    | 67.76        |

the ViT-B/32 4-bit P4Q result in Table 1. This improvement introduces only 0.13 MB additional parameters and keeps the FLOPs unchanged.

We further implement QAdapter-text by applying the same bottleneck adapter to the final text feature  $\hat{\mathbf{w}}_k$ , producing an adapted text prototype  $\hat{\mathbf{u}}_k$ . As shown in Table 4, QAdapter-text performs post-encoder prototype correction on class text embeddings, which are shared across inputs. In contrast, the visual QAdapter performs input-conditioned calibration on quantized image features. Combining QAdapter-text with the visual QAdapter achieves 65.07%, which is higher than QAdapter-text alone at 60.50%, but still lower than the visual QAdapter alone at 66.28%. This suggests that the text-side residual does not provide complementary gains once visual calibration is applied. Therefore, P4Q uses prompt tuning for text-side adaptation and the visual QAdapter for image-branch calibration.

**4.7.2 Hyper-parameters Selection.** We select the hyper-parameters prompt length  $M$  and adaptation ratio  $\alpha$  using a 4-bit P4Q trained for 20 epochs. Top-1 accuracy under different hyper-parameter pairs  $\{M, \alpha\}$  is presented in Fig. 7. Results indicate that performance initially improves and then declines as  $\alpha$  varies from 0 to 1. This demonstrates the necessity of image feature adaptation, with full adaptation at  $\alpha = 1$  outperforming the vanilla base at  $\alpha = 0$ . Additionally, P4Q with prompt learning exhibits stronger performance than with  $M = 0$ , but full prompt learning at  $M = 32$  performs worse than all alternatives. Exploring the setups, we identify  $\{M, \alpha\} = \{16, 0.2\}$  as the combination that boosts P4Q's performance the most, achieving

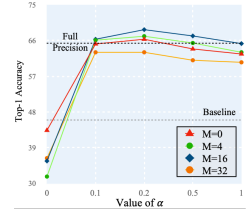
Fig. 7. Effect of prompt length and adaptation ratio  $\{M, \alpha\}$ .

Table 6. Effect of attention bit-width on ImageNet (CLIP ViT-B/32). 32-32-32 (FP) is full precision under the linear-probe protocol (cf. Table 2). Rows 4-4-8 / 4-4-4 are PTQ-only sweeps over attention precision and are not comparable to the adapted 61.97% Top-1 P4Q entry in Table 2. W-A-Attn: weight, activation, and attention bit-widths.

| Bits (W-A-Attn) | Top-1 | Top-5 | $\Delta$ |
|-----------------|-------|-------|----------|
| 32-32-32 (FP)   | 76.1  | 93.2  | –        |
| 4-4-8           | 66.9  | 92.5  | -9.2     |
| 4-4-4           | 57.4  | 85.1  | -18.7    |

68.12% Top-1 accuracy. Based on the ablative study above, we set hyper-parameters  $M$  and  $\alpha$  as 16 and 0.2 for the experiments in this paper.

**4.7.3 Loss Analysis.** We analyze the effectiveness of the proposed  $L_c$  and  $L_{dist}$  in the 4-bit setting, trained for 20 epochs as listed in Table 5. Relative to the 4-bit PTQ baseline of 46.05%,  $L_{dist}$  alone yields +18.45 percentage points to 64.50%, and  $L_c$  alone yields +20.29 percentage points to 66.34%. With both losses and  $\lambda=1$ , Top-1 reaches 68.12%, i.e., +1.78 percentage points over  $L_c$  alone; this setting gives the best  $\lambda$  among {0.5, 1.0, 1.5}.

**4.7.4 Effect of Attention Bit-Width.** We further analyze the sensitivity of attention precision under low-bit quantization. Specifically, we vary the attention activation bit-width while keeping weight and non-attention activation precision fixed. Table 6 reports results on ImageNet using CLIP ViT-B/32. We observe that reducing attention precision from 8-bit to 4-bit leads to a substantial accuracy drop of about 9.5 percentage points in absolute Top-1 when comparing the 4-4-8 and 4-4-4 settings, indicating that attention maps are significantly more sensitive to quantization noise than linear layers. This phenomenon is consistent with prior multimodal PTQ findings, where attention activations are typically preserved at higher precision to stabilize softmax dynamics and cross-modal alignment. Based on these observations, we adopt the W4A4Attn8 configuration as a balanced and commonly accepted setting.

## 5 Discussions

While our experimental evaluation focuses on image classification, the P4Q framework is designed with generality in mind. It addresses a fundamental challenge in quantized multimodal models: the embedding space misalignment introduced when vision and language encoders are quantized independently. This issue is not unique to classification and is likely to affect any downstream task relying on cross-modal feature matching.

The proposed quantization-aware prompt tuning and Q-Adapter modules could therefore be explored beyond classification. Potential extensions include open-vocabulary detection and segmentation, where methods [56, 55] have demonstrated the transferability of prompt-based adaptation. In such settings, P4Q might maintain alignment between quantized visual features and task-adapted textual prompts. Similarly, for vision-language reasoning models that feed visual embeddings into a frozen language model, P4Q could serve as a lightweight calibration interface to mitigate quantization-induced error propagation, drawing inspiration from recent prompt-based VLM adaptation works [14].

P4Q can be extended to recent MLLMs by placing QAdapter at the modality-bridging interface between visual representations and the language model. In projector- or connector-based architectures, the projected visual-token sequence can be adapted as  $\tilde{\mathbf{V}} = \mathbf{V} + \text{QAdapter}(\text{LN}(\mathbf{V}))$ , where  $\mathbf{V}$

denotes the visual tokens after the vision–language connector. This design keeps the quantized backbone frozen and calibrates the visual tokens before they are processed by the language model. For models such as Qwen2.5-VL [2] and LLaVA-OneVision-1.5 [1], the same interface-level calibration can be applied after the visual encoder or vision–language connector, because the visual representations are ultimately mapped into the language-model embedding space. For Qwen3-VL [3], which introduces DeepStack-style multi-level visual feature injection, one may retain one QAdapter at the final vision–language bridge or place lightweight adapters at the injected visual-feature levels. These extensions do not modify the quantized LLM backbone. A systematic empirical study on these newer architectures is left to future work.

## 6 Conclusion

This paper proposes a method that integrates parameter-efficient fine-tuning with quantization named “Prompt for Quantization” (P4Q) to enable efficient and effective deployment of CLIP in downstream applications. We design a low-bit and lightweight adapter with learnable prompts to significantly improve the recognition performance of a PTQ model. Besides, we propose a joint loss function including a contrastive loss and a distillation loss, which can effectively align the cross-modal distributions of the quantized model to their full-precision counterparts. Overall, our P4Q quantized low-bit CLIP models obtain superior performance than PTQ baselines, even achieving comparable results to its full-precision counterparts with negligible added parameters on 13 diverse image classification datasets. In our future work, we will explore the potential of our method in more diverse VLMs and other applications.

## References

- [1] Xiang An et al. 2025. Llava-onevision-1.5: fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*. arXiv: 2509.23661 [cs.CV]. doi:10.48550/arXiv.2509.23661.
- [2] Shuai Bai et al. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*. arXiv: 2502.13923 [cs.CV]. doi:10.48550/arXiv.2502.13923.
- [3] Shuai Bai et al. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*. arXiv: 2511.21631 [cs.CV]. doi:10.48550/arXiv.2511.21631.
- [4] Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. 2023. Composed image retrieval using contrastive learning and task-oriented clip-based features. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20, 3, 1–24.
- [5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part VI 13*. Springer, 446–461.
- [6] Tom Brown et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877–1901.
- [7] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Zeroq: a novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13169–13178.
- [8] Xi Chen et al. 2022. Pali: a jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*.
- [9] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. 2019. Low-bit quantization of neural networks for efficient inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 3009–3018.
- [10] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3606–3613.
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: a large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.
- [12] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.
- [13] Alexey Dosovitskiy et al. 2021. An image is worth 16x16 words: transformers for image recognition at scale. In *ICLR*.
- [14] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. 2022. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*.

- [15] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2004. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*. IEEE, 178–178.
- [16] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.
- [17] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. 2021. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*.
- [18] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. Eurosat: a novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12, 7, 2217–2226.
- [19] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. 2021. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15262–15271.
- [20] Dan Hendrycks et al. 2021. The many faces of robustness: a critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8340–8349.
- [21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- [22] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- [23] Yifan Jiao, Chenglong Cai, and Bing-Kun Bao. 2025. Unified text-image space alignment with cross-modal prompting in clip for uda. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21, 3, 1–20.
- [24] Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W Mahoney, and Kurt Keutzer. 2023. Squeezellm: dense-and-sparse quantization. *arXiv preprint arXiv:2306.07629*.
- [25] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 554–561.
- [26] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. In *Citeseer*.
- [27] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*. PMLR, 12888–12900.
- [28] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. 2019. Fully quantized network for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2810–2819.
- [29] Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: optimizing continuous prompts for generation. In *IJCNLP*, 4582–4597.
- [30] Yanjing Li, Sheng Xu, Baochang Zhang, Xianbin Cao, Peng Gao, and Guodong Guo. 2022. Q-vit: accurate and fully quantized low-bit vision transformer. *arXiv preprint arXiv:2210.06707*.
- [31] Ji Lin et al. 2024. Awq: activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6, 87–100.
- [32] Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. 2022. Fq-vit: post-training quantization for fully quantized vision transformer. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 1173–1179.
- [33] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36, 34892–34916.
- [34] Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2023. Llm-qat: data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*.
- [35] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. 2013. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.
- [36] Brais Martinez, Jing Yang, Adrian Bulat, and Georgios Tzimiropoulos. 2020. Training binary neural networks with real-to-binary convolutions. *ICLR*.
- [37] Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*. IEEE, 722–729.
- [38] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 3498–3505.
- [39] Adam Paszke et al. 2017. Automatic differentiation in pytorch. In *NIPS-W*.
- [40] Genevieve Patterson, Chen Xu, Hang Su, and James Hays. 2014. The sun attribute database: beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108, 59–81.

- [41] Alec Radford et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.
- [42] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21, 140, 1–67.
- [43] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2019. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*. PMLR, 5389–5400.
- [44] Weimin Shi, Dehong Gao, Yuan Xiong, and Zhong Zhou. 2024. Qr-clip: introducing explicit knowledge for location and time reasoning. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20, 11, 1–22.
- [45] K Soomro. 2012. Ucf101: a dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.
- [46] Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. 2024. Q-vlm: post-training quantization for large vision-language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- [47] Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. 2024. Q-vlm: post-training quantization for large vision-language models. *Advances in Neural Information Processing Systems*, 37, 114553–114573.
- [48] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. 2019. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32.
- [49] Runqi Wang, Hao Zheng, Xiaoyue Duan, Jianzhuang Liu, Yuning Lu, Tian Wang, Songcen Xu, and Baochang Zhang. 2023. Few-shot learning with visual distribution calibration and cross-modal distribution alignment. *CVPR*.
- [50] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. 2021. A simple baseline for zero-shot semantic segmentation with pre-trained vision-language model. In *arXiv preprint arXiv:2112.14757*.
- [51] Baochang Zhang, Runqi Wang, Xiaodi Wang, Jungong Han, and Rongrong Ji. 2021. Modulated convolutional networks. *IEEE Transactions on Neural Networks and Learning Systems*.
- [52] Haifeng Zhao, Chi Zhang, Deyin Liu, and Lin Wu. 2025. Deformation field fusion for medical image registration. *ACM Transactions on Multimedia Computing, Communications and Applications*.
- [53] Huazhong Zhao, Lei Qi, and Xin Geng. 2024. Clip-dfgs: a hard sample mining method for clip in generalizable person re-identification. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21, 1, 1–20.
- [54] Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [mask]: learning vs. learning to recall. *NAACL*.
- [55] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16816–16825.
- [56] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130, 9, 2337–2348.
- [57] Sheng Zhou, Dan Guo, Xun Yang, Jianfeng Dong, and Meng Wang. 2024. Graph pooling inference network for text-based vqa. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20, 4, 1–21.
- [58] Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian Reid. 2018. Towards effective low-bitwidth convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7920–7928.