

# Quantized Prompt for Efficient Generalization of Vision-Language Models

Tianxiang Hao<sup>1,2</sup>, Xiaohan Ding<sup>3</sup>(✉), Juexiao Feng<sup>1,2</sup>, Yuhong Yang<sup>1,2</sup>,  
Hui Chen<sup>2</sup>, and Guiguang Ding<sup>1,2</sup>(✉)

<sup>1</sup> School of Software, Tsinghua University, Beijing, China

<sup>2</sup> BNRist, Beijing, China

<sup>3</sup> Bytedance

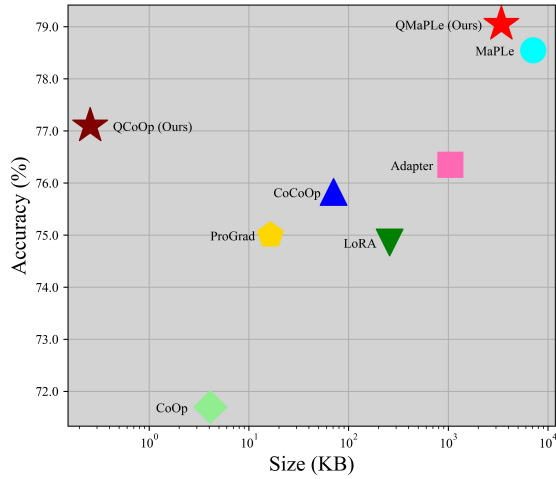
{beyondhtx,xiaohding,suise.con,jichenhui}@gmail.com  
fjx20@mails.tsinghua.edu.cn, dinggg@tsinghua.edu.cn

**Abstract.** In the past few years, large-scale pre-trained vision-language models like CLIP have achieved tremendous success in various fields. Naturally, how to transfer the rich knowledge in such huge pre-trained models to downstream tasks and datasets becomes a hot topic. During downstream adaptation, the most challenging problems are overfitting and catastrophic forgetting, which can cause the model to overly focus on the current data and lose more crucial domain-general knowledge. Existing works use classic regularization techniques to solve the problems. As solutions become increasingly complex, the ever-growing storage and inference costs are also a significant problem that urgently needs to be addressed. While in this paper, we start from an observation that proper random noise can suppress overfitting and catastrophic forgetting. Then we regard quantization error as a kind of noise, and explore quantization for regularizing vision-language model, which is quite efficiency and effective. Furthermore, to improve the model’s generalization capability while maintaining its specialization capacity at minimal cost, we deeply analyze the characteristics of the weight distribution in prompts, conclude several principles for quantization module design and follow such principles to create several competitive baselines. The proposed method is significantly efficient due to its inherent lightweight nature, making it possible to adapt on extremely resource-limited devices. Our method can be fruitfully integrated into many existing approaches like MaPLe, enhancing accuracy while reducing storage overhead, making it more powerful yet versatile. Extensive experiments on 11 datasets shows great superiority of our method sufficiently. Code is available at [github](#)

**Keywords:** Quantization · Generalization · Vision-language model

## 1 Introduction

Recently, deep learning models and related technologies have seen rapid development [9, 11–13, 22–24, 49, 50, 65, 66]. Vision-language model (VLM) like CLIP [58] is one of the hottest research topics and leads to huge success. The excellent generalization ability is a crucial cornerstone of such achievement [26, 53, 58, 59, 68].



**Fig. 1:** Overall performance comparison with existing vision-language tuning methods. Our method outperforms all of the state-of-the-art competitors with significantly fewer storage space. Based on the proposed quantization algorithm, our method could be integrated into many of the existing methods and bring consistent improvements with excellent efficiency.

When people have access to downstream data, it is better to tune the pre-trained VLM on the target dataset for higher accuracy. However, such full fine-tuning could easily cause the model to overfit the small downstream dataset and face catastrophic forgetting problem, leading to severe performance drop.

To solve the problem, in this paper we will start from rethinking the relationship between noise and generalization. We propose to think noise as a kind of regularization techniques, which may be helpful for alleviating overfitting and catastrophic forgetting problem. As in Sec. 3.2 and Fig. 2, we then find that directly adding some random Gaussian noise to the tunable prompts of vision-language models would result in a performance gain in several cases, which partly validates our conjecture about using noise as a form of regularization. In particular, excessive noise diminishes the model’s adaptation capability, while insufficient noise fails to provide effective regularization. Only noise of moderate intensity is beneficial for the model’s generalization.

However, Gaussian noise is absolutely random and hard to control, and thus it is difficult for us to take full advantage of such type of noise to benefit generalization. Instead, we point out that quantization error is also a form of noise, and therefore, it is also possible to leverage quantization error to enhance the model’s generalization performance. With this idea in mind, we thoroughly analyzed the distribution pattern of prompt weights in Sec. 3.3 and derived several design principles for quantization algorithms based on the observed phenomena. In Sec. 4, following the principles we summarized, we successfully designed an efficient quantization-aware training algorithm, which largely enhances the model’s generalization ability while quantizing.

Since our design is general, we could fruitfully integrate our quantization strategy into many existing methods and reach a higher accuracy with much smaller model size due to the lightweight nature of quantization. For example, we integrate our method onto an existing popular method MaPLe, and our QMaPLe earns 0.57% accuracy improvement with only  $0.25\times$  size.

In conclusion, we summarize our contribution as follows:

- We deeply analyze the effect of noise and rethink the relationship between noise and generalization in Sec. 3.2 for vision-language models. As a result, we confirm that moderate noise would promote the model generalization
- We are the first to propose to quantize prompts. By detailed observation and hard thinking, we conclude several principles about how to effectively design a quantization method for the prompts of vision-language models.
- Following the principles we concluded, we build our method and successfully quantize the prompts as well as some other weights, and our method significantly outperforms existing ones as in Fig. 1. Extensive experiments show great power of our method. In base-to-new generalization, domain generalization, cross-dataset transfer and few-shot learning settings, we consistently reach competitive results, winning many state-of-the-art tuning methods with a much smaller size of model.

## 2 Related Works

### 2.1 Vision-Language Models

In recent times, large-scale vision-language models have demonstrated remarkable performance across various tasks. Seminal works such as [26, 58, 68, 70, 73]. Classic works have focused on learning multimodal representations through self-supervised methods using extensive sets of image-text pairs. Among these, CLIP [58] stands out as a milestone achievement, employing contrastive learning to align vision and language representations and achieving exceptional performance. A well-trained vision-language model is invaluable, offering substantial support to various fields. Successful applications of these robust models include few-shot recognition [76, 77], detection tasks [15, 51, 60, 72], and segmentation tasks [7, 36, 47, 59]. Furthermore, for video data, research efforts have emerged in video classification [57] and video understanding [30].

### 2.2 Parameter-Efficient Fine-Tuning

Recently, a series of works [19, 21, 23, 24, 27, 29, 38, 41, 44, 48, 64, 76, 77] have been proposed to help transfer the learnt knowledge, where one of the most popular field is parameter-efficient fine-tuning (PEFT). PEFT aims to transfer a pre-trained model to downstream tasks by a minimum number of parameters. Originating from natural language processing tasks, classic methods like adapter [23], prompt tuning [28, 35, 38, 42, 43, 61] and LoRA [24] follows a similar

principle to add extra modules with a small number of parameters into the backbone model, freeze the original parameters and only tune and store the newly added parameters. Inspired by the success of PEFT in language field, researchers have extended such kind of approaches to adapt visual models in a similar fashion [6, 19, 27, 41, 74]. In the field of vision-language modeling, several explorations have been made as well. Bahng et al. [1] exclusively apply prompt tuning to the image encoder. CoOp [77] replaces the fixed template in CLIP [58] with tunable text prompts. CoCoOp [76] leverages image features to guide the optimization of tunable text prompts in CoOp. Other works [32, 34] optimize both image and text prompts simultaneously and establish additional connections between different modalities. To mitigate overfitting and catastrophic forgetting, various works [5, 33, 67, 75] integrate regularization modules or losses into prompt tuning.

### 2.3 Quantization

Quantization is one of the most effective compression methods [4, 8, 10, 20, 37, 46, 69] for deep learning models. Generally, parameters such as weights and activations are typically stored as 32-bit floating-point numbers, which consume a significant amount of memory and require intensive computation during inference. Quantization [17, 18, 55] involves representing these parameters with reduced precision, such as 8-bit integers or even lower bit-widths. By doing so, quantization can significantly reduce the memory footprint and computational complexity of the model without significantly sacrificing accuracy. Quantization methods can be divided into two groups, Post-Training Quantization (PTQ) [2, 16, 25, 40, 45, 52, 54, 71] that consumes few resources but suffers higher accuracy loss, and Quantization-Aware Training (QAT) [3, 14, 31, 39, 56] that relies on plenty of resources for training and shows better accuracy. Existing works aim to minimize quantization error to improve accuracy, while our work demonstrates that both excessive and insufficient errors are detrimental to model generalization. To achieve optimal generalization performance, a moderate error is required.

## 3 Exploring Quantization in Model Generalization

### 3.1 Preliminaries: Prompt Tuning of Vision-Language Models

CLIP comprises a text encoder  $\mathcal{L}$  and an image encoder  $\mathcal{V}$ . Typically,  $\mathcal{L}$  is implemented as a language transformer, whereas  $\mathcal{V}$  may be realized using either a convolutional neural network or a vision transformer. In this study, following the methodologies outlined by [76, 77], we employ a ViT-B/16 model [13] as the image encoder  $\mathcal{V}$ , except where otherwise specified. The subsequent sections will provide a brief overview of the methods used to prompt CLIP for prediction tasks.

**Text Encoder** Consider a text encoder composed of  $M$  layers. For the  $k$ -th layer, denoted as  $\mathcal{L}_k$ , the inputs consist of a sequence of prompt tokens  $P_{k-1}^l$

and a  $[CLS]$  token  $c_{k-1}^l$ , while the outputs are represented by  $P_k^l$  and  $c_k^l$ . The initial inputs,  $P_0^l$  and  $c_0^l$ , correspond to the word embeddings of the prompts combined with the label, such as “A photo of a  $[CLS]$ ” or alternatively, some randomly initialized vectors. Formally, we denote  $P_k^l \in \mathbb{R}^{n^l \times d^l}$  and  $c_k^l \in \mathbb{R}^{d^l}$ , where  $n^l$  signifies the length of the text prompts and  $d^l$  represents the dimension of the word embeddings. For each layer,  $1 \leq k \leq M$ , the relationship is given by  $[P_k^l, c_k^l] = \mathcal{L}_k([P_{k-1}^l, c_{k-1}^l])$ . The output feature of the text encoder  $f^l \in \mathbb{R}^{d^v}$ , where  $d^v$  is the dimension of the visual feature space, is calculated by projecting the  $[CLS]$  token of its last layer to the visual latent space through a linear transformation, *i.e.*  $f^l = \text{Proj}(c_M^l)$ .

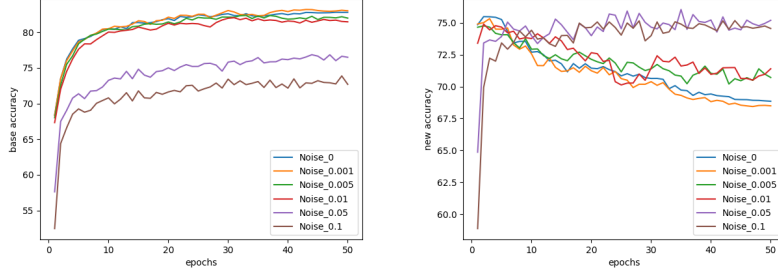
**Image Encoder** Suppose there are  $N$  layers in the image encoder. For  $k$ -th layer  $\mathcal{V}_k$ , the inputs are a series of image tokens  $I_{k-1}$ , a classification token  $c_{k-1}^v$  and prompt tokens  $P_{k-1}^v$ , and the outputs are  $I_k$ ,  $c_k^v$  and  $P_k^v$ . The inputs of the first layer  $I_0$  and  $c_0^v$  are the patch embeddings of the input image and the pre-trained class token.  $P_0^v$  is randomly initialized. Formally,  $I_k \in \mathbb{R}^{p \times d^v}$ ,  $c_k^v \in \mathbb{R}^{d^v}$  and  $P_k^v \in \mathbb{R}^{n^v \times d^v}$ , where  $p$  denotes the number of image patches and  $d^v$  denotes the dimension of visual embedding.  $\forall 1 \leq k \leq N$ ,  $[P_k^v, c_k^v, I_k] = \mathcal{V}_k([P_{k-1}^v, c_{k-1}^v, I_{k-1}])$ . The output feature of the image encoder is  $f^v = c_N^v$ .

**Prediction** For image classification, suppose there are  $C$  classes, and  $\{f_c^l\}_{c=1}^C$  are the text features. Label  $y$ 's probability is  $p(y|f^v) = \frac{\exp(\text{sim}(f^v, f_y^l)/\tau)}{\sum_{c=1}^C \exp(\text{sim}(f^v, f_c^l)/\tau)}$  where  $\text{sim}(\cdot, \cdot)$  denotes cosine similarity function and  $\tau$  is temperature. The final prediction is  $\hat{z} = \arg \max_{1 \leq y \leq C} (p(y|f^v))$ .

We have introduced shallow prompts. There are also different types of prompts. Several works [27, 32] use them to improve performance. They directly add and tune the prompt in each layer in the feature encoder, instead of inheriting the output prompt calculated by the last encoder. Now we have  $[_-, c_k^l] = \mathcal{L}_k([P_{k-1}^l, c_{k-1}^l])$  and  $[_-, c_k^v, I_k] = \mathcal{V}_k([P_{k-1}^v, c_{k-1}^v, I_{k-1}])$ . Note that  $P^l$  and  $P^v$  are independent tunable parameters. They are no longer determined by the previous layer.

### 3.2 Rethinking the Relationship between Noise and Generalization

Some existing work has explored how to use the addition of noise to suppress model overfitting, *e.g.*, techniques like Dropout [62] that randomly drops connections, and random jitter that directly introduce perturbations in the input data. However, the impact of directly adding noise to model weights has been less explored, especially in the context of prompt structures, which have become popular only in recent years, in Transformer architectures. The variations of the specialization capability represented by the test accuracy on base, seen classes and the generalization capability represented by the test accuracy on new, unseen classes. The curves of different colors represent the data under the influence of random Gaussian noise of different intensities, *e.g.* “Noise\_0.01” adds random noise with a distribution of  $\mathcal{N}(0, 0.01^2)$  to the prompt. “Noise\_0” denotes the baseline prompt tuning. As training progresses, the generalization capability



**Fig. 2:** The variations of the specialization capability represented by the test accuracy on base, seen classes and the generalization capability represented by the test accuracy on new, unseen classes on average of the ten datasets in base-to-new generalization setting in Sec. 5 except for time-consuming ImageNet. The curves of different colors represent the data under the influence of random Gaussian noise of different intensities, *e.g.* “Noise\_0.01” adds random noise with a distribution of  $\mathcal{N}(0, 0.01^2)$  to the prompt. “Noise\_0” denotes the baseline prompt tuning. As training progresses, the generalization capability of baseline prompt tuning continuously decreases while the specialization capability improves. Therefore, we expect that adding noise can achieve a better balance between generalization and specialization. However, excessive noise, *e.g.* 0.1, greatly diminishes the model’s specialization capability, while insufficient noise, *e.g.* 0.001, fails to provide effective regularization. Only noise of moderate intensity outperforms baseline in specialization-generalization trade-off, effectively enhancing the unseen class accuracy without significantly compromising seen class accuracy.

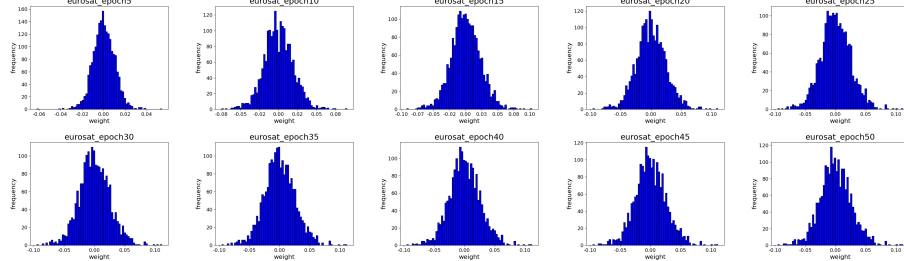
of baseline continuously decreases while the specialization capability improves. Therefore, we expect that adding noise can achieve a better balance between generalization and specialization. However, excessive noise, *e.g.* 0.1, greatly diminishes the model’s specialization capability, while insufficient noise, *e.g.* 0.001, fails to provide effective regularization. Only noise of moderate intensity outperforms baseline in specialization-generalization trade-off, effectively enhancing the unseen class accuracy without significantly compromising seen class accuracy.

Quantization, the technique of mapping parameter values from high precision to low precision, can also be viewed as introducing some form of noise into the parameters, and thus can possibly improve the generalizability as the noise did in Fig. 2. Compared to Gaussian noise, quantization error is more controllable. Better still, quantization also has another major advantage: it can significantly reduce the storage required for parameters. Therefore, using quantization algorithms to generalize vision-language models is a very promising direction.

### 3.3 Characteristics of Prompts in Vision-Language Models

In this subsection, we will demonstrate the characteristics of prompts in the vision-language model to facilitate targeted design of quantization schemes.

We start from analyzing the training procedure of CoOp [77], which freezes all the parameters in backbone and tunes the added prompts only. The histogram about the frequency of weights of prompts for different training epochs is shown

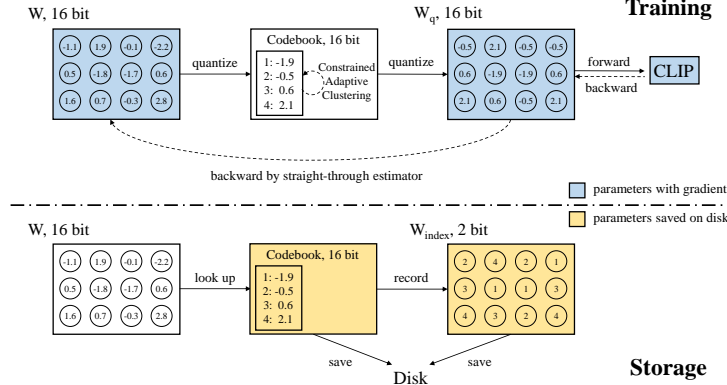


**Fig. 3:** The histogram about the frequency of weights of prompts during training of CoOp [77] on eurosat dataset. We find the shape of the prompt’s weight distribution remains largely unchanged throughout the entire training, but the variance of the weight distribution increases rapidly at the beginning of training. Additionally, we also notice that there are almost no outliers in the prompt’s weights throughout the entire training process, and apart from the unstable initial training phase, the changes in weights between adjacent phases are not significant, indicating a very gentle overall updating trend.

in Fig. 3. We observe that the shape of the prompt’s weight distribution remains mostly consistent throughout the entire training process. However, the variance of the weight distribution increases rapidly at the initial stages of training. Refer to Fig. 6 for more details. Moreover, we notice minimal presence of outliers in the prompt’s weights throughout the training process. Apart from the unstable initial training phase, the changes in weights between consecutive phases are not substantial, indicating a very gentle overall updating trend.

Taking into account the observations from Secs. 3.2 and 3.3, we can draw the following conclusions:

- Because the prompt weights are not sensitive to noise, and moderate noise in the current scenario not only does not degrade the model’s performance but actually enhances its generalization ability, we can possibly adopt some quantization strategies that are considered very aggressive and highly likely to significantly degrade performance, such as 1-bit quantization, without causing destructive effects on the model’s generalization ability.
- Given that the parameters targeted for quantization in the current scenario constitute only a tiny fraction of the total model parameters, we speculate that QAT may have more advantages over PTQ, as any additional training costs incurred by QAT which is considered a major drawback in traditional situations would naturally be kept at a low level due to the low proportion of parameters targeted for quantization. Due to the observation in Fig. 2 that both excessive and insufficient noise are detrimental to generalization, in designing the QAT algorithm, we no longer need to always minimize quantization error as the objective, as in classic algorithms. Instead, we aim to maintain the quantization error at a moderate level. Furthermore, as seen in Fig. 3, the changes in prompt weight distribution during most of the training time are gentle, indicating that we can appropriately continue to use past-time algorithm state without the need for real-time updates.



**Fig. 4:** Overview of Quantized Prompt. We set  $b = 2$  for a clear explanation. Normalization and denormalization processes are not shown here.

- Throughout the training process, the shape of the prompt weight distribution remains roughly unchanged, with the main variation coming from the distribution’s variance. Therefore, it may be beneficial to attempt to eliminate the translation and scaling transformations of the distribution by normalizing the distribution before quantization and denormalizing it after quantization, which might help improve quantization accuracy.
- Since there are almost no outliers throughout the entire training process, some classic algorithms are revitalized. Specifically, clustering algorithms like K-Means are good choices, because one of the major drawbacks of them in traditional usage is that they are greatly affected by outliers. In addition, as we have analyzed before, there is not a high demand for dynamically minimizing quantization error in vision-language generalization, opting for a more efficient non-parametric clustering approach seems a better choice compared with some complex parametric approaches.

## 4 Prompt Quantization

### 4.1 Preliminaries: Quantization Basis

Suppose there are  $m$  parameters in total to be quantized, which are denoted by  $W \in \mathbb{R}^m$ . Each  $w_i$  here is a high-precision float-type number. A  $b$ -bit model quantization algorithm divides the value range of parameters into  $2^b$  intervals  $\{\mathcal{U}_i, 1 \leq i \leq 2^b\}$ , aiming to find a mapping  $Q$  from intervals to points, mapping all values within an interval  $\mathcal{U}_i$  to the same quantized value  $q_i$ , *i.e.*  $Q(x) = q_i, \forall x \in \mathcal{U}_i$ . We define the quantization error  $E = \sum_{i=1}^N \|Q(W_i) - W_i\|^2$ . Such error serves as the objective of K-Means algorithm as well, so it can be ensured that each time clustering is redone using the K-Means algorithm, the quantization error will most likely decrease.



## 4.2 Overview of Quantized Prompt

Based on the observation and analysis in Sec. 3.3, we build a simple yet quite proper quantization algorithm for generalizing vision-language models. As stated before, we choose a widely-used clustering method K-Means, to construct the mapping  $Q$ . Initially, K-Means algorithm is run to fit the pre-trained prompt weights, and record the codebook, *i.e.* cluster centers. Normalization and de-normalization are conducted beyond quantization to alleviate the influence of varying variance. Formally, given the original prompt  $W$  with  $N$  parameters, we first calculate  $\mu = \frac{1}{N} \sum_{i=1}^N W_i$  and  $\sigma = \sqrt{\frac{\sum_{1 \leq i \leq N} (W_i - \mu)^2}{N}}$ , then normalize  $W$  and get  $\hat{W} = \frac{W - \mu}{\sigma}$ . K-Means quantization is applied on  $\hat{W}$ , and the quantized prompt  $W_q = \sigma Q(\hat{W}) + \mu$ . Since  $Q$  is not differentiable, we adopt a common practice to directly propagate the gradient across the quantization function by Straight-Through Estimator (STE), *i.e.*  $\frac{\partial Q(x)}{\partial x} = x$ .

The overall framework is shown in Fig. 4. In training, we keep tuning the weight by the fake gradient propagated from quantization operation. For storage, we first convert the fp16 parameters in prompt to  $b$ -bit indexes, which could be used to search for the corresponding cluster center in the codebook. In Fig. 4, we set  $b = 2$  for a clear explanation. Compared with baseline method, our quantized method could save a lot more storage space. Specifically, for storage, an ordinary method needs  $16N$  bits, while ours only needs  $bN + 2^b \times 16$ . In experiments, we usually set  $b$  to 1, 2 or 4, which is far more smaller than  $N$ . For example, with  $b = 1$ , the storage space of our method is roughly  $16\times$  smaller than baseline.

Note that the codebook here is updated by a rule called “constrained adaptive clustering”. We will give a detailed description of it in the following subsection.

## 4.3 Constrained Adaptive Clustering for Quantization

In Fig. 4, re-clustering the prompt every iteration like classic QAT algorithms did to keep the codebook updated at all times is not a good choice. There are three reasons: 1. K-Means algorithm is not quite efficient, and thus if we run it every iteration then the training efficiency would decrease; 2. Only moderate noise promotes generalization, so keep updating the clusters to minimize the quantization error may be not helpful but harmful; 3. From Fig. 3, we can observe that the weight changes are very gentle for most of the training time, and the weight distributions of adjacent stages are highly similar. Therefore, even if we update the internal clustering state of K-Means with these already very similar data at each iteration, it’s difficult to generate a better clustering solution, which is just futile effort.

We propose constrained adaptive clustering to instruct the update of parameters inside K-Means which would not updated by gradient at all. Intuitively, first, we do not want too often update, so we set a minimum cluster update interval  $t$ . Second, to avoid K-Means meaninglessly handling similar weights with current cached state, *i.e.* the weights of prompt that triggers the last re-clustering, we plan to only do re-clustering when current weight distribution is far different

from the cached weight distribution. To reach the goal, Kullback-Leibler divergence is a good metric. However, if we want to compute the Kullback-Leibler divergence between two sets of discrete random variables, they must be defined on the same event space. However, that’s not the case. So we first project each of them into the same event space spanning by the K-Means clustering algorithm. Specifically, given a current weight  $W^{cur}$ , cached weight  $W^{old}$ , quantization function  $Q$ , cluster centers  $\mathcal{C} = \{c_i, 1 \leq i \leq 2^b\}$ , we will compute the  $W_{index}^{cur}$  and  $W_{index}^{old}$  as in Fig. 4. Then, we can obtain the probability distributions of the indices  $p^{cur}$  and  $p^{old}$ , respectively, implied by  $W_{index}^{cur}$  and  $W_{index}^{old}$ , thus successfully transforming two originally unrelated discrete random variables into the same event space. The Kullback-Leibler divergence could be computed as follows:

$$KL(p^{cur}||p^{old}) = \sum_{i=1}^{2^b} p^{cur}(i) \log \frac{p^{cur}(i)}{p^{old}(i)} \quad (1)$$

The update would continue only if Kullback-Leibler divergence exceed a certain threshold  $T_{KL}$  which shows the distribution difference is significant.

## 5 Experiments

We evaluate the method and make comparisons with the latest state-of-the-art methods in terms of the following settings across a wide range:

1. **Base-to-new generalization**, which models are trained with base classes and evaluated on both base and new classes.
2. **Domain generalization**, which aims to show generalization to the domain shift, especially for out-of-distribution data.
3. **Cross-dataset transfer**, which aims to see if the method has the potential to transfer beyond a single dataset. It is a much more challenging problem because the fundamentals can be totally changed across different datasets.
4. **Few-shot learning**, which aims to evaluate the adaptation performance of the model to extract knowledge from a dataset whose samples are very few, *e.g.* 1, 2, 4, 8 or 16 samples.

All methods are initialized with the same CLIP weights provided by the open-source CLIP [58]. In Appendix, we provide more details of datasets, experimental setting and competitors introduction. Due to page limit, we also put some detailed experimental results into Appendix, *e.g.* the performance of each method on each dataset in base-to-new generalization setting.

### 5.1 Main Results

**Base-to-new Generalization** The average results over 11 datasets are shown in Tab. 1. For complete results, please refer to the Appendix. Our proposed QCoOp reaches **77.43%** average harmonic mean accuracy within merely **0.26KB** size. QCoOp outperforms all kinds of lightweight state-of-the-art methods with

**Table 1:** Comparisons with latest methods in base-to-new generalization. H: harmonic mean [63]. QCoOp and QMaPLe are significantly better than other competitors, which prove that our quantization design could be fruitfully integrated into many existing approaches to further improve the performance and efficiency.

	Size	Base	New	H
CLIP	0KB	69.34	74.22	71.70
CoOp	4.1KB	82.69	63.22	71.66
CoCoOp	70.8KB	80.47	71.69	75.83
Adapter	1051KB	82.62	70.97	76.35
LoRA	258KB	<b>84.30</b>	67.33	74.86
ProGrad	16.4KB	82.79	68.55	75.00
QCoOp	<b>0.26KB</b>	80.68	<b>74.44</b>	<b>77.43</b>
MaPLe	7096KB	82.28	75.14	78.55
QMaPLe	<b>1774KB</b>	<b>83.02</b>	<b>75.57</b>	<b>79.12</b>

**Table 2:** Comparisons with latest methods in domain generalization. QCoOp gets comparable or even better results with the latest state-of-the art methods with much fewer parameters, showing excellent robustness for domain shift.

	Size	Source	Target				Average
		ImageNet	-V2	-Sketch	-Adversarial	-Rendition	
CLIP	0KB	66.73	60.83	46.15	47.77	73.96	57.18
CoOp	4.1KB	71.51	64.20	47.99	49.71	75.21	59.28
CoCoOp	70.8KB	71.02	64.07	48.75	50.63	76.18	59.91
Adapter	1051KB	69.33	62.53	47.67	49.17	75.42	58.70
LoRA	258KB	70.30	62.37	42.43	38.40	68.97	53.04
ProGrad	16.4KB	<b>72.24</b>	<b>64.73</b>	47.61	49.39	74.58	59.07
QCoOp	0.26KB	70.67	63.87	<b>48.93</b>	<b>51.10</b>	<b>76.90</b>	<b>60.20</b>

much more efficiency and higher accuracy. For heavier methods like MaPLe, our method can be fruitfully integrated into existing solutions. Besides prompts, we also perform a similar quantization operation on the other weights of MaPLe that are in the linear layers. As a result, QMaPLe not only shows stronger generalization and adaptation capability and gives 0.57% higher accuracy, but also enjoys a much more smaller model size compared with the original MaPLe.

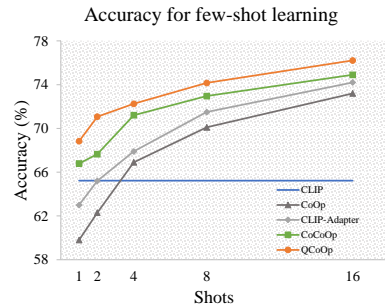
Notably, when comparing QCoOp and a lightweight method ProGrad, even if QCoOp is  $63\times$  smaller than ProGrad, QCoOp still outperforms it by a clear margin, demonstrating the outstanding efficiency and effectiveness.

**Domain Generalization** In this paragraph, ImageNet, ImageNet-A, ImageNet-R, ImageNet-v2, and ImageNet-S are used to construct domain generalization experiments. As shown in Tab. 2, on downstream target datasets, QCoOp gets better average accuracy compared with the other methods with significantly better efficiency. For CLIP, CoOp, CLIP-Adapter and ProGrad, there is a clear performance gap between our method and them.

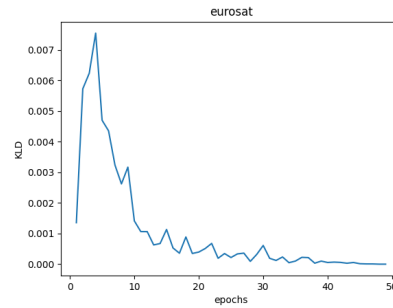
**Cross-dataset Transfer** In this paragraph, we do cross-dataset transfer evaluation to further verify our QPrompt. Results are shown in Tab. 3. CoOp is good on source domain but fails on target domains. Probably because it focus too much on the dataset shown to its eyes and face overfitting and catastrophic

**Table 3:** Results in the cross-dataset transfer setting. QCoOp gives the highest accuracy on 5 of 10 datasets, which well demonstrates that QCoOp could maximally extract general and data-agnostic knowledge from input images.

	Size	Source	Target										Average
		ImageNet	Caltech101	Pets	Cars	Flowers	Food101	Aircraft	Sun397	DTD	EuroSAT	UCF101	
CoOp	4.1KB	<b>71.51</b>	93.70	89.14	64.51	68.71	85.30	18.47	64.15	41.92	46.39	66.55	63.88
CoCoOp	70.8KB	71.02	<b>94.43</b>	90.14	65.32	<b>71.88</b>	86.06	22.94	<b>67.36</b>	<b>45.73</b>	45.37	68.21	65.74
Adapter	1051KB	69.33	93.43	88.87	64.40	70.27	85.63	<b>24.67</b>	65.80	44.90	47.70	66.00	65.17
QCoOp	0.26KB	70.63	94.07	<b>90.53</b>	<b>65.97</b>	71.33	<b>86.23</b>	22.73	66.80	44.20	<b>48.23</b>	<b>69.17</b>	<b>65.93</b>



**Fig. 5:** Average few-shot learning results on 11 datasets.



**Fig. 6:** KLD trend of prompt weights under the same experiment with Fig. 3.

forgetting problems, which leads to a severe performance drop on unseen objects. QCoOp wins on 5 of 10 datasets and its average accuracy is also slightly better than the best competitor CoCoOp, showing that QCoOp could maximally extract both general and data-agnostic knowledge from given images.

**Few-shot Learning** Here we will show the experiment results of QCoOp in the few-shot learning setting that is originated from CoOp. Seen from Fig. 5, QCoOp consistently outperforms zero-shot CLIP, CoOp, and CLIP-Adapter across all the shot numbers. Such results demonstrate the superiority of QCoOp in adaptation ability when there are few samples in downstream tasks.

Overall, in base-to-new generalization, domain generalization, cross-dataset transfer and few-shot learning, the proposed method can consistently accomplish state-of-the-art performance while enjoying extremely high parameter efficiency, fruitfully demonstrating the effectiveness and efficiency of the proposed method.

## 6 Analysis

### 6.1 Ablation Studies

**Component** In this subsection, we decompose the method into pieces and show the influence of each component. As in Tab. 4, we clearly show how

**Table 4:** Ablation study on separate component of QCoOp. Norm: Normalization/Denormalization. CAC: Constrained Adaptive Clustering.

	K-Means	Norm	CAC	base	new	H
QCoOp	✓	✗	✗	78.71	72.55	75.50
	✓	✓	✗	<b>78.84</b>	73.09	75.85
	✓	✓	✓	78.24	<b>74.02</b>	<b>76.07</b>

**Table 5:** Comparisons between QAT and PTQ based on CoOp. Both QAT and CoOp are trained with the same hyper-parameters. Here the quantization bit is 1.

	base	new	H
QAT	80.72	<b>72.35</b>	<b>76.31</b>
PTQ	<b>82.21</b>	68.50	74.73

much the K-Means algorithm, normalization/denormalization and constrained adaptive clustering influence the final performance. One interesting point is that K-Means+Norm+CAC only improves the new accuracy compared with K-Means+Norm, showing the superiority of our constrained adaptive clustering.

**Trend of the distributions of prompt between adjacent epochs.** To verify the opinions we proposed at the end of Sec. 6, we show the KLD of prompt distributions between adjacent epochs during CoOp’s training in Fig. 6. Clearly, this trend is consistent with what we summarized before.

**QAT v.s. PTQ** In Tab. 5, we study the choice of QAT or PTQ following the same strategy, K-Means clustering, in the paper. Results show that QAT consistently outperforms PTQ with the same hyper-parameters. The accuracy on unseen new classes of PTQ is significantly lower than QAT, again proving our opinion that quantization helps generalization by alleviating overfitting as well as catastrophic forgetting. Quantization error is not always undesirable.

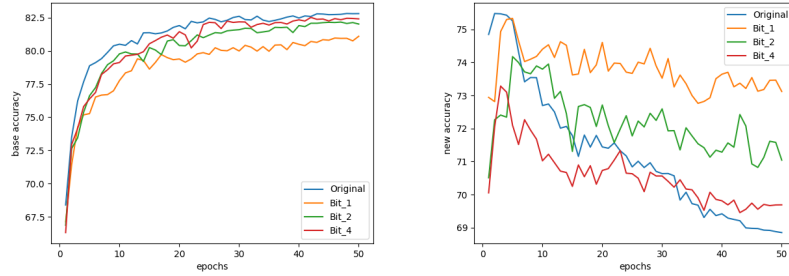
**Quantization bit** In Tab. 6 and Fig. 7, we show the results across multiple quantization bits. In conclusions, more bits did not always lead to good results, and new accuracy continues decreasing as the training goes on.

## 6.2 Performance on Self-Supervised Vision-Language Model

In this paragraph, we explore the usage of QCoOp among other different backbones besides CLIP. We choose a self-supervised vision-language model, SLIP [53] to further verify the universality and robustness of our proposed method. The experimental results are shown in Tab. 7. We could see that the base accuracies of QCoOp and CoOp are similar, but the new accuracies of QCoOp is much

**Table 6:** Results across different quantization bits.

	Size	base	new	H
QCoOp ( $b = 1$ )	0.26KB	79.49	72.65	75.92
QCoOp ( $b = 2$ )	0.52KB	80.30	71.98	75.91
QCoOp ( $b = 4$ )	1.05KB	80.92	71.22	75.76



**Fig. 7:** The same training curves with Fig. 2. Clearly, increasing quantization noise leads to the same phenomenon with increasing gaussian noise, *i.e.* more generalizability (new accuracy) and less adaptability (base accuracy).

**Table 7:** Results based on SLIP [53] model.

	Size	base	new	H
CoOp	4.1KB	<b>68.65</b>	46.60	55.51
QCoOp	0.26KB	68.33	<b>74.04</b>	<b>71.07</b>

higher than CoOp. Such observation verifies our assumption that quantization is quite helpful for generalization again.

## 7 Conclusion

With the development of huge vision-language models, how to effectively and efficiently adapt such huge models to downstream tasks becomes a challenging problem. Much effort has been made to leverage the potential of prompt tuning in adapting vision-language models. However, existing methods suffer from inefficiency. To reach extremely efficient generalization, we propose QPrompt based on the detailed analysis and deep understanding of the characteristics of the prompt weight distribution. Following the several principles concluded by us, we use K-Means clustering algorithm as the base of our quantization method. To adaptively control the quantization error and minimize the number of re-clustering operations, we propose to do a dynamic check every few iterations. If the Kullback-Leibler divergence between the current weights and the original weights used in the last clustering exceeds a predefined threshold and the quantization error also increases, we will let the model re-cluster the weights and update the new centers and weights. Similarly, continue this process until completion. The proposed method could be simply integrated into many of the existing vision-language tuning methods like CoOp and MaPLe and reach good performance. Importantly, our proposed method is significantly effective and efficient. Extensive experiments show that the designed quantization algorithm indeed improves the generalizability of vision-language model, and save a lot more storage space as well.

**Acknowledgements.** This work was supported by National Natural Science Foundation of China (Nos. 61925107, 62271281, 62021002)

## References

1. Bahng, H., Jahanian, A., Sankaranarayanan, S., Isola, P.: Visual prompting: Modifying pixel space to adapt pre-trained models. arXiv preprint arXiv:2203.17274 (2022) [4](#)
2. Banner, R., Nahshan, Y., Soudry, D.: Post training 4-bit quantization of convolutional networks for rapid-deployment. *Advances in Neural Information Processing Systems* **32** (2019) [4](#)
3. Bhalgat, Y., Lee, J., Nagel, M., Blankevoort, T., Kwak, N.: Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. pp. 696–697 (2020) [4](#)
4. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. In: *The Eleventh International Conference on Learning Representations* (2023) [4](#)
5. Bulat, A., Tzimiropoulos, G.: Lasp: Text-to-text optimization for language-aware soft prompting of vision & language models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 23232–23241 (2023) [4](#)
6. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems* **35**, 16664–16678 (2022) [4](#)
7. Ding, J., Xue, N., Xia, G.S., Dai, D.: Decoupling zero-shot semantic segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 11583–11592 (2022) [3](#)
8. Ding, X., Ding, G., Guo, Y., Han, J.: Centripetal sgd for pruning very deep convolutional networks with complicated structure. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 4943–4953 (2019) [4](#)
9. Ding, X., Guo, Y., Ding, G., Han, J.: Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 1911–1920 (2019) [1](#)
10. Ding, X., Hao, T., Tan, J., Liu, J., Han, J., Guo, Y., Ding, G.: Resrep: Lossless cnn pruning via decoupling remembering and forgetting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 4510–4520 (2021) [4](#)
11. Ding, X., Zhang, X., Han, J., Ding, G.: Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 11963–11975 (2022) [1](#)
12. Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., Sun, J.: Repvgg: Making vgg-style convnets great again. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 13733–13742 (2021) [1](#)
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Deghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations* (2020) [1](#), [4](#)
14. Esser, S.K., McKinstry, J.L., Bablani, D., Appuswamy, R., Modha, D.S.: Learned step size quantization. In: *International Conference on Learning Representations* (2019) [4](#)
15. Feng, C., Zhong, Y., Jie, Z., Chu, X., Ren, H., Wei, X., Xie, W., Ma, L.: Prompt-det: Towards open-vocabulary detection using uncurated images. In: *European Conference on Computer Vision*. pp. 701–717. Springer (2022) [3](#)

16. Finkelstein, A., Almog, U., Grobman, M.: Fighting quantization bias with bias. arXiv preprint arXiv:1906.03193 (2019) [4](#)
17. Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M.W., Keutzer, K.: A survey of quantization methods for efficient neural network inference. In: *Low-Power Computer Vision*, pp. 291–326. Chapman and Hall/CRC (2022) [4](#)
18. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149 (2015) [4](#)
19. Hao, T., Chen, H., Guo, Y., Ding, G.: Consolidator: Mergeable adapter with grouped connections for visual adaptation. arXiv preprint arXiv:2305.00603 (2023) [3](#), [4](#)
20. Hao, T., Ding, X., Han, J., Guo, Y., Ding, G.: Manipulating identical filter redundancy for efficient pruning on deep and complicated cnn. *IEEE Transactions on Neural Networks and Learning Systems* (2023) [4](#)
21. Hao, T., Lyu, M., Chen, H., Zhao, S., Han, J., Ding, G.: Re-parameterized low-rank prompt: Generalize a vision-language model within 0.5 k parameters. arXiv preprint arXiv:2312.10813 (2023) [3](#)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 770–778 (2016) [1](#)
23. Houlshby, N., Giurigu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: *International Conference on Machine Learning*. pp. 2790–2799. PMLR (2019) [1](#), [3](#)
24. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021) [1](#), [3](#)
25. Hubara, I., Nahshan, Y., Hanani, Y., Banner, R., Soudry, D.: Accurate post training quantization with small calibration sets. In: *International Conference on Machine Learning*. pp. 4466–4475. PMLR (2021) [4](#)
26. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: *International Conference on Machine Learning*. pp. 4904–4916. PMLR (2021) [1](#), [3](#)
27. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. arXiv preprint arXiv:2203.12119 (2022) [3](#), [4](#), [5](#)
28. Jiang, Z., Araki, J., Ding, H., Neubig, G.: How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics* **9**, 962–977 (2021) [3](#)
29. Jie, S., Wang, H., Deng, Z.H.: Revisiting the parameter efficiency of adapters from the perspective of precision redundancy. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 17217–17226 (2023) [3](#)
30. Ju, C., Han, T., Zheng, K., Zhang, Y., Xie, W.: Prompting visual-language models for efficient video understanding. In: *European Conference on Computer Vision*. pp. 105–124. Springer (2022) [3](#)
31. Jung, S., Son, C., Lee, S., Son, J., Han, J.J., Kwak, Y., Hwang, S.J., Choi, C.: Learning to quantize deep networks by optimizing quantization intervals with task loss. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4350–4359 (2019) [4](#)



32. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: Maple: Multi-modal prompt learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19113–19122 (2023) 4, 5
33. Khattak, M.U., Wasim, S.T., Naseer, M., Khan, S., Yang, M.H., Khan, F.S.: Self-regulating prompts: Foundational model adaptation without forgetting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15190–15200 (2023) 4
34. Lee, D., Song, S., Suh, J., Choi, J., Lee, S., Kim, H.J.: Read-only prompt optimization for vision-language few-shot learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1401–1411 (2023) 4
35. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691 (2021) 3
36. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. In: International Conference on Learning Representations (2022), <https://openreview.net/forum?id=RriDjddCLN> 3
37. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710 (2016) 4
38. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021) 3
39. Li, Y., Xu, S., Zhang, B., Cao, X., Gao, P., Guo, G.: Q-vit: Accurate and fully quantized low-bit vision transformer. *Advances in Neural Information Processing Systems* **35**, 34451–34463 (2022) 4
40. Li, Z., Xiao, J., Yang, L., Gu, Q.: Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17227–17236 (2023) 4
41. Lian, D., Zhou, D., Feng, J., Wang, X.: Scaling & shifting your features: A new baseline for efficient model tuning. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2022) 3, 4
42. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586 (2021) 3
43. Liu, X., Ji, K., Fu, Y., Du, Z., Yang, Z., Tang, J.: P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. arXiv preprint arXiv:2110.07602 (2021) 3
44. Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., Tang, J.: Gpt understands, too. arXiv preprint arXiv:2103.10385 (2021) 3
45. Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., Gao, W.: Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems* **34**, 28092–28103 (2021) 4
46. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE international conference on computer vision. pp. 2736–2744 (2017) 4
47. Lüddecke, T., Ecker, A.: Image segmentation using text and image prompts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7086–7096 (2022) 3
48. Lyu, M., Hao, T., Xu, X., Chen, H., Han, J., Ding, G.: Learn from the learnt: Source-free active domain adaptation via contrastive sampling and visual persistence. In: *European Conference on Computer Vision (ECCV)*. Springer (2024) 3
49. Lyu, M., Yang, Y., Hong, H., Chen, H., Jin, X., He, Y., Xue, H., Han, J., Ding, G.: One-dimensional adapter to rule them all: Concepts diffusion models and erasing

- applications. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7559–7568 (2024) [1](#)
50. Lyu, M., Zhou, J., Chen, H., Huang, Y., Yu, D., Li, Y., Guo, Y., Guo, Y., Xiang, L., Ding, G.: Box-level active detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23766–23775 (2023) [1](#)
  51. Maaz, M., Rasheed, H., Khan, S., Khan, F.S., Anwer, R.M., Yang, M.H.: Class-agnostic object detection with multi-modal transformer. In: The European Conference on Computer Vision. Springer (2022) [3](#)
  52. Meller, E., Finkelstein, A., Almog, U., Grobman, M.: Same, same but different: Recovering neural network quantization error through weight factorization. In: International Conference on Machine Learning. pp. 4486–4495. PMLR (2019) [4](#)
  53. Mu, N., Kirillov, A., Wagner, D., Xie, S.: Slip: Self-supervision meets language-image pre-training. In: European Conference on Computer Vision. pp. 529–544. Springer (2022) [1](#), [13](#), [14](#)
  54. Nagel, M., Baalen, M.v., Blankevoort, T., Welling, M.: Data-free quantization through weight equalization and bias correction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1325–1334 (2019) [4](#)
  55. Nagel, M., Fournarakis, M., Amjad, R.A., Bondarenko, Y., Van Baalen, M., Blankevoort, T.: A white paper on neural network quantization. arXiv preprint arXiv:2106.08295 (2021) [4](#)
  56. Nagel, M., Fournarakis, M., Bondarenko, Y., Blankevoort, T.: Overcoming oscillations in quantization-aware training. In: International Conference on Machine Learning. pp. 16318–16330. PMLR (2022) [4](#)
  57. Qian, R., Li, Y., Xu, Z., Yang, M.H., Belongie, S., Cui, Y.: Multimodal open-vocabulary video classification via pre-trained vision and language models. arXiv preprint arXiv:2207.07646 (2022) [3](#)
  58. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763. PMLR (2021) [1](#), [3](#), [4](#), [10](#)
  59. Rao, Y., Zhao, W., Chen, G., Tang, Y., Zhu, Z., Huang, G., Zhou, J., Lu, J.: Denseclip: Language-guided dense prediction with context-aware prompting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18082–18091 (2022) [1](#), [3](#)
  60. Rasheed, H.A., Maaz, M., Khattak, M.U., Khan, S., Khan, F.: Bridging the gap between object and image-level representations for open-vocabulary detection. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022), <https://openreview.net/forum?id=aKXBrjODHm> [3](#)
  61. Shin, T., Razeghi, Y., Logan IV, R.L., Wallace, E., Singh, S.: Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 4222–4235 (2020) [3](#)
  62. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014) [5](#)
  63. Xian, Y., Schiele, B., Akata, Z.: Zero-shot learning-the good, the bad and the ugly. In: CVPR (2017) [11](#)
  64. Xiong, Y., Chen, H., Hao, T., Lin, Z., Han, J., Zhang, Y., Wang, G., Bao, Y., Ding, G.: Pyra: Parallel yielding re-activation for training-inference efficient task adaptation. arXiv preprint arXiv:2403.09192 (2024) [3](#)

65. Xiong, Y., Chen, H., Lin, Z., Zhao, S., Ding, G.: Confidence-based visual dispersal for few-shot unsupervised domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11621–11631 (2023) [1](#)
66. Xiong, Y., Chen, X., Ye, X., Chen, H., Lin, Z., Lian, H., Niu, J., Ding, G.: Temporal scaling law for large language models. arXiv preprint arXiv:2404.17785 (2024) [1](#)
67. Yao, H., Zhang, R., Xu, C.: Visual-language prompt tuning with knowledge-guided context optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6757–6767 (2023) [4](#)
68. Yao, L., Huang, R., Hou, L., Lu, G., Niu, M., Xu, H., Liang, X., Li, Z., Jiang, X., Xu, C.: FILIP: Fine-grained interactive language-image pre-training. In: International Conference on Learning Representations (2022), <https://openreview.net/forum?id=cpDhcsEDC2> [1](#), [3](#)
69. Yu, S., Chen, T., Shen, J., Yuan, H., Tan, J., Yang, S., Liu, J., Wang, Z.: Unified visual transformer compression. In: International Conference on Learning Representations (2022) [4](#)
70. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al.: Florence: A new foundation model for computer vision. arXiv preprint arXiv:2111.11432 (2021) [3](#)
71. Yuan, Z., Xue, C., Chen, Y., Wu, Q., Sun, G.: P4v: Post-training quantization for vision transformers with twin uniform quantization. In: European Conference on Computer Vision. pp. 191–207. Springer (2022) [4](#)
72. Zang, Y., Li, W., Zhou, K., Huang, C., Loy, C.C.: Open-vocabulary detr with conditional matching. arXiv preprint arXiv:2203.11876 (2022) [3](#)
73. Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A., Beyer, L.: Lit: Zero-shot transfer with locked-image text tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18123–18133 (2022) [3](#)
74. Zhang, Y., Zhou, K., Liu, Z.: Neural prompt search. arXiv preprint arXiv:2206.04673 (2022) [4](#)
75. Zheng, K., Wu, W., Feng, R., Zhu, K., Liu, J., Zhao, D., Zha, Z.J., Chen, W., Shen, Y.: Regularized mask tuning: Uncovering hidden knowledge in pre-trained vision-language models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11663–11673 (2023) [4](#)
76. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16816–16825 (2022) [3](#), [4](#)
77. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision **130**(9), 2337–2348 (2022) [3](#), [4](#), [6](#), [7](#)