Advancing Vision-Language Models with Adapter Ensemble Strategies

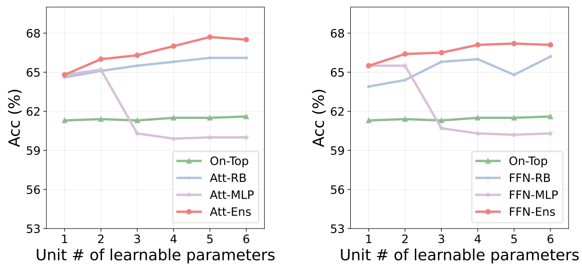
Anonymous ACL submission

Abstract

CLIP (Radford et al., 2021) revolutes vision-language pretraining by using contrastive learning on paired web data. However, the sheer size of these pretrained models makes full-model finetuning exceedingly costly. One common solution is the "adapter", which finetunes a few additional parameters while freezing the backbone. It harnesses the heavy-duty backbone while offering a light finetuning for small downstream tasks. This synergy prompts us to explore the potential of augmenting large-scale backbones with traditional machine learning techniques. Often employed in traditional fields and overlooked in the large-scale era, these techniques could provide valuable enhancements. Herein, we delve into the "adapter ensembles" in the realm of large-scale pretrained vision-language models. We begin with a proof-of-concept study to establish the efficacy of combining multiple adapters. We then present extensive evidence showing these ensembles excel in a variety of settings, particularly when employing a Multi-Scale Attention (MSA) approach thoughtfully integrated into the ensemble framework. We further incorporate the LoRA to mitigate the additional parameter burden. We focus on vision-language retrieval, using different backbones under constraints of minimal data, parameters, and finetuning budgets. This research paves the way for a synergistic blend of traditional, yet effective, strategies with modern large-scale networks.

1 Introduction

Large-scale pretraining leverages massive data, robust architectures with strategic training to push performance boundaries (Devlin et al., 2018; Radford et al., 2018; Li et al., 2022; Radford et al., 2021). It notably advances vision-language capabilities, exemplified by CLIP (Radford et al., 2021), which through contrastive learning on a vast image-text corpus, seamlessly integrates visual and linguistic modalities.



(a) Attn ensemble ablation. (b) FFN ensemble ablation.

Figure 1: CLIP ViT-B/16 ensemble ablation on self-attention and feedforward (Sec. 2). Y-axis/x-axis are the retrieval accuracy and the unit number of learnable parameters in each layer. Baselines (**On-Top**, **RB**, **MLP**) and our **Ens** are finetuned/evaluated on YFCC. Sharing the same amount of learnable parameters, ensemble outperforms baselines and derives improvement when the number of ensemble parameters increases.

Various studies further advance vision-language pretraining by integrating auxiliary supervision (e.g., self-supervision/captioning loss) or extra information (e.g., tags/bounding boxes) (Ramesh et al., 2022; Saharia et al., 2022; Tewel et al., 2022; Chen et al., 2022a; Mokady et al., 2021; Jia et al., 2021; Mu et al., 2022). However, the necessity for extensive datasets and complex training pipelines for pretraining remain a challenge, particularly affecting finetuning efficiency. Adapter (Houlsby et al., 2019) is a favored technique for efficient finetuning, initially for language models like BERT (Devlin et al., 2018) and recently adapted for the visual domain (Chen et al., 2022b; Gao et al., 2021). Along with its variants such as LoRA (Hu et al., 2021) and Compactor (Karimi Mahabadi et al., 2021), adapter offers the solution by updating a few additional parameters with limited data while fixing the pretrained backbone. These approaches combine large-scale pretraining with small-sized efficient adapters, proposing a unified modeling pipeline. This fusion compels us to consider if we can borrow certain traditional machine learning techniques, which work well on previous small-sized scenarios but are easily ignored in the current large-scale era, to benefit the popular pretrained models. Informed by this, our study delves the classic *ensemble* on adapter for large-scale vision-language pretrained models and assesses its impact on cross-modal retrieval.

Ensemble has long been a cornerstone in traditional machine learning, combining diverse base learners to harness collective intelligence, thereby enhancing model performance and robustness (Dietterich, 2000; Sagi and Rokach, 2018; Rokach, 2010). In past decades, early methods provided weak yet cheap base learners using limited data, the ensemble compensated by pooling their strengths. Recently neural networks, with more data and complex models, present base learners of greater individual capability. Yet, the ensemble continues to offer performance boosts (Li et al., 2019; Lee et al., 2018), albeit at a cost, given the non-negligible resources to entirely train each deep network as a base learner. Nowadays, the focus shifts towards leveraging single, robustly pretrained models, leaving ensembles less tapped for these larger models due to their prohibitive computational demands. However, our curiosity lies in applying ensemble to efficiently finetune large-scale pretrained models using adapters, which act as a nexus for integrating large-scale backbone and small-sized techniques.

This study marks the initial exploration into the use of the adapter-based ensembles in large-scale pretrained models. We infuse the pretrained model with parallel learnable parameters in an ensemble fashion while fixing original weights. Our proof-of-concept study (Sec. 2) shows substantial performance gain of ensemble over baselines (Fig. 1). We further extensively validate its effectiveness with a well-designed Multi-Scale Attention (MSA) in an ensemble framework (Sec. 3). Finally, we enhance our strategy by incorporating LoRA (Hu et al., 2021) technique, managing the extra parameter overhead to maintain efficiency with competitive performance even when scaling to ensemble applications. We summarize contributions of our study as below:

- Driven by the adapter efficiency, we are intrigued by the potential of leveraging classical small-sized machine learning techniques to enhance the large-scale model performance.

- We recall the ensemble, which is a classical practice but mostly overlooked in current large-scale era. Herein, we use *adapter ensemble* as an intermediary between large-scale pretrained model and small-sized technique to improve pretrained model under efficient finetuning budget.

- We conduct 1) a proof-of-concept study, promising our exploration as a valuable perspective; 2) an extensive ensemble test, showing consistent performance gain over different settings; 3) a simple ensemble-style Multi-Scale Attention (MSA), reaching the largest performance gain of cross-modal retrieval (e.g., **6%** YFCC zero-shot improvement with only **0.1M** Laion finetuning data); 4) an incorporation with LoRA into our ensemble to maintain the adapter parameter efficiency (e.g., **2.2%** additional parameters with competitive performance).

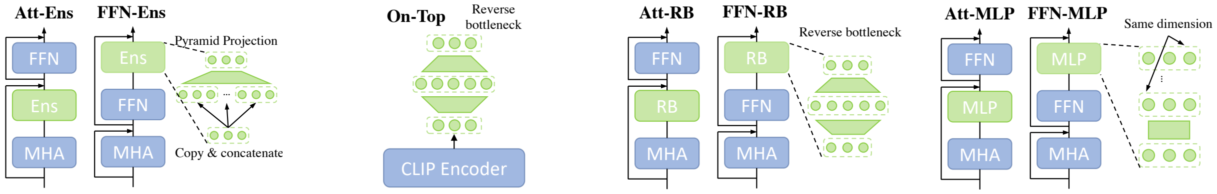
2 Ensemble Proof-of-Concept Study

Ensemble is often interpreted as a weighting strategy (Rokach, 2010; Dietterich, 2000), where data or feature fusion can be regarded as an ensemble process to some extent. For example, residual connection (He et al., 2016) is an ensemble process fusing identity mapping and learned residual information. In this section, we conduct an instructive empirical analysis as a proof-of-concept study to show the effectiveness of using an ensemble strategy on adapter. We finetune (using limited 0.1M data) and test on YFCC (Thomee et al., 2016) to compare our ensemble (**Att-Ens/FFN-Ens**) with three baselines (**On-Top, Att-RB/FFN-RB, Att-MLP/FFN-MLP**) on CLIP backbone.

Att-Ens/FFN-Ens.
We make a simple implementation to include a few sets of learnable parameters for ensemble, which is different from typical bottleneck adapter (Houlsby et al., 2019). Given a feature $f \in \mathbb{R}^d$ after multi-head attention (Att) or feedforward (FFN) in each transformer block, we project the copied and concatenated feature using a pyramid layer:

$$f^{\text{ens}} = f + ([\overbrace{f, ..., f}^{N}])W^{\text{ens}}, \tag{1}$$

where $W^{\text{ens}} \in \mathbb{R}^{Nd \times d}$ and we omit bias term for convenience (Fig. 2a). N is the number of copied feature to be concatenated. In this way, each d-dim sub-matrix in W^{ens} can be treated as a base learner. The pyramid projection is an ensemble module.



(a) **Att-Ens/FFN-Ens** add a pyramid projection to ensemble concatenated copied features for MHA or FFN.

(b) **On-Top** adds additional parameter (reverse bottleneck) on the top of both CLIP vision/language towers.

(c) **Att-RB/FFN-RB** add a reverse bottleneck as additional parameters after MHA or FFN.

(d) **Att-MLP/FFN-MLP** add projections (same dimension) as learnable parameters after MHA or FFN.

Figure 2: Instructive analysis to show our ensemble strategy (Fig. 2a) works better than baselines (Fig. 2b 2c 2d) while sharing the same number of additional learnable parameters overall. We adjust 1) the number of copied feature for Att-Ens/FFN-Ens (Fig. 2a); 2) the hidden dimension in reverse bottleneck for On-Top/Att-RB/FFN-RB (Fig. 2b 2c); 3) the number of hidden layers for Att-MLP/FFN-MLP (Fig. 2d) to keep the same amount of additional parameter for all methods. All four methods are deployed in both vision and language towers. In figures, green and blue blocks represent learnable and frozen modules, respectively.

Accordingly, we can conveniently calculate the total number of additional learnable parameters. Assuming we have total L blocks in pretrained CLIP, the totally amount of additional parameters is $L \times Nd \times d$. We regard $d \times d$ as an adapter unit and $L \times N$ is the number of the total units. To show the benefits of ensemble strategy, we make a comparative analysis with the following three designed baselines, w.r.t. different numbers of units of additional parameters, shown as the number of x-axis in Fig. 1.

On-Top

To eliminate any potential ensemble effect, we use CLIP to extract feature f and place all the additional learnable parameters as a reverse bottleneck on the top (Fig. 2b) without any residual skip, which is given by

$$f^{\text{top}} = (f \cdot W^1)W^2, \tag{2}$$

where $W^1 \in \mathbb{R}^{d \times (LNd/2)}$ and $W^2 \in \mathbb{R}^{(LNd/2) \times d}$. This is the most basic baseline, with no ensemble influence.

Att-RB/FFN-RB

We insert a reverse bottleneck after Att or FFN in each block (Fig. 2c). Residual skip is used here to relatively involve ensemble factor and alleviate the non-ensemble constraint compared with On-Top, given by:

$$f^{rb} = f + (f \cdot W^1)W^2, \tag{3}$$

where $W^1 \in \mathbb{R}^{d \times (Nd/2)}$ and $W^2 \in \mathbb{R}^{(Nd/2) \times d}$. Skip connection involves ensemble concept but the reverse bottleneck is not for ensemble compared with Att-Ens/FFN-Ens.

Att-MLP/FFN-MLP

We insert an MLP after Att or FFN in each block (Fig. 2d). This is another version to allow ensemble by using skip connection, given by

$$f^{rb} = f + (f \cdot W^1)W^2 \cdots W^N, \tag{4}$$

where $W^i \in \mathbb{R}^{d \times d}, i = \{1, 2, ..., N\}$. We keep the same dimension for all hidden layers across i.

For a fair comparison, we keep the same total number of additional parameters ($L \times Nd \times d$) for all four methods through adjusting the number of layers for Att-MLP/FFN-MLP and hidden dimension for others. All of four methods (Fig. 2) are deployed on both vision and language towers simultaneously. Fig. 1 shows the performance comparison between ensemble and baselines. ***-Ens** consistently outperforms others. With more additional parameters, we also observe the increasing ensemble performance. ***-RB** and ***-MLP** using ensemble to some extent obtain competitive results, even if adding more units of parameters damages the learning process for ***-MLP** due to no skip connection inside. **On-Top** with no ensemble has lowest performance and adding more parameters fails to improve more. Based on these observations, we conclude relaxing a few learnable parameters to execute a light-weight ensemble is effective in efficiently improving a pretrained large-scale model.

3 Adapter Ensemble

We show the effectiveness of involving an adapter ensemble into a pretrained model in Sec. 2. Next, we introduce a bottleneck adapter baseline, a pyramid ensemble, and a well-designed multi-scale attention (MSA) ensemble for our comprehensive validation on multiple settings. Furthermore, we easily adopt LoRA (Hu et al., 2021) into our ensemble design to ease the parameter burden caused by ensemble operation.

Bottleneck Adapter/Pyramid Ensemble

We follow the typical adapter (Houlsby et al., 2019) and insert two bottlenecks after self-attention and feedforward modules, and ensemble them together with the skip connections, given by

$$f^{bo} = f + F((f \cdot W^1)W^2, (f \cdot W^3)W^4), \quad (5)$$

where $W^1, W^3 \in \mathbb{R}^{d \times d_a}$ and $W^2, W^4 \in \mathbb{R}^{d_a \times d}$. d_a is the hidden dimension. $F(\cdot,\cdot)$ serves as an ensemble operation implemented as averaging in our case. The pyramid ensemble is based on our introduction in Fig. 2a. The same feature is encoded several times by different sub-matrices in the pyramid projection and integrated in an ensemble fashion. Specifically, we set $N = 2$ to ensemble two base learners for our extensive validation.

Multi-Scale Attention

Recall that the success of ensemble leveraging on diverse base learners to achieve the *crowd intelligence* (Rokach, 2010; Ganaie et al., 2021). The learners' diversity can be reflected from different aspect by different fashions (Dietterich, 2000; Rokach, 2010). For example, base learners can be trained from different datasets for ensemble. They can also come from different models such as neural network, decision tree, etc. Similarly, since neural networks are commonly trained by SGD introducing randomness into the trained model, repeatedly training model is also an effective way for ensemble (Li et al., 2019; Lee et al., 2018). Here, we are motivated by the Longformer (Beltagy et al., 2020) to tailor a multi-scale attention (MSA) to diversify our attention features. We propose a simple ensemble-based approach to implement this strategy. Formally, self-attention in transformer is originally given by

$$Att(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \quad (6)$$

where Q, K, V are query, key, and value vectors after projections. d_k is the feature dimension of K. We separate the original attention into three different scales (large, middle, and small) by applying different masks. For language tower, we define the mask as

$$M_C^*[i,j] = \begin{cases} 1, & |i-j| < D_C^*, \\ 0, & |i-j| \geq D_C^*, \end{cases} \quad (7)$$

where $M_C^* \in \mathbb{R}^{T_C \times T_C}$ and T_C is the number of caption tokens. D_C^* is the length of scale $*$ and $* \in \{L, M, S\}$ for large, middle and small scales, respectively. Since the language token is a 1D sequence, the mask for language is just as a banded matrix (Fig. 3). Similarly, we define the mask for the image tower as

$$M_I^*[i,j] = \begin{cases} 1, & \max(|x_i - x_j|, |y_i - y_j|) < D_I^*, \\ 0, & \max(|x_i - x_j|, |y_i - y_j|) \geq D_I^*, \end{cases} \quad (8)$$

where x_*, y_* are the 2D visual patch positions converted from the 1D token sequence given by $x_k = \lfloor k/P_I \rfloor, y_k = k - x_k \cdot P_I$. P_I is the number of patches in each row (or column) in a given image. The converting step makes the mask not as a banded matrix but representing different scales in the original 2D visual scenario (Fig. 3). After defining M_C and M_I, we describe the MSA by revising Eq. 6 as

$$Att^*(Q, K, V) = \text{softmax}(\frac{QK^T \odot M^*}{\sqrt{d_k}})V, \quad (9)$$

for different scales in vision/language towers. $\odot$ applies mask on corresponding attention score matrix. We ensemble the MSA features from Eq. 9 as

$$f^{\text{ens}} = f + [f^L, f^M, f^S]W^{\text{ens}}, \quad (10)$$

where W^{ens} is the pyramid projection to ensemble f^L, f^M, and f^S for large, middle, and small scales, respectively. In addition, we also add a bottleneck adapter after feedforward layer with our MSA to further enhance network capacity.

LoRA Adoption

Our MSA integrates multiple branches as basic learners for ensemble and may also cause additional parameter burden for finetuning, even if we only focus on the adapter module. We simply adopt a low-rank (Hu et al., 2021) design here to solve this concern. We replace the ensemble operation (Eq. 10) by adding a learnable low-rank matrix on each scale branch as

$$f^* = Att^*(f^*) + BA^*f^*, \quad (11)$$

where $* \in \{L, M, S\}$ are different branches. B and A^* are learnable low-rank matrices, where B is shared for all branches. We add features of all branches for ensemble as $f^{ens} = f^L + f^M + f^S$ instead of using a pyramid layer. We also replace the bottleneck adapter after FFN in MSA with this

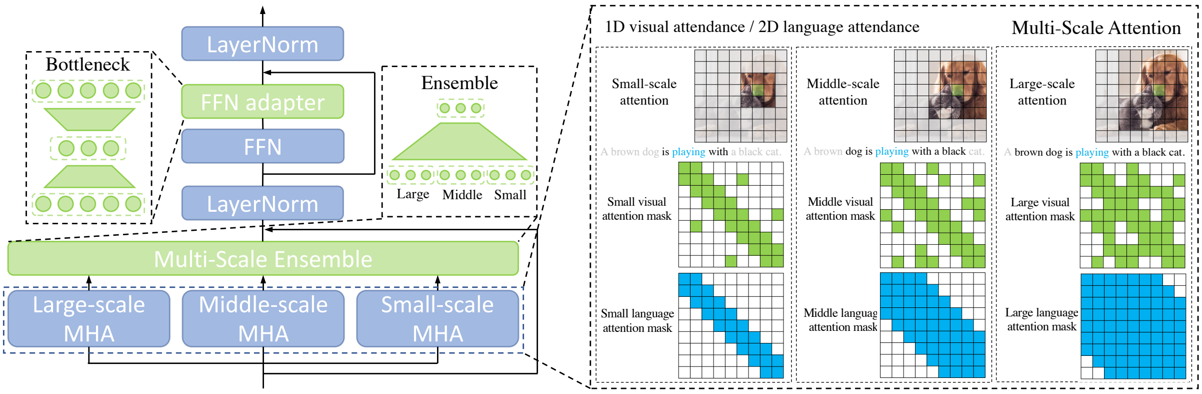


Figure 3: Illustration of multi-scale attention (MSA). It is specifically designed to benefit ensemble strategy by extracting diverse representations from multiple different scales. It consists of two parts: 1) MSA and 2) FFN adapter shown on the left. Different masks of large, middle, and small scales are applied on self-attention score matrix to yield different features representing corresponding scales. Given a scale, corresponding masks are constructed for vision and language shown on the right. Visual and language tokens are originally placed in 2D and 1D, respectively. A pyramid projection is used to make multi-scale ensemble and map back to original dimension. The FFN adapter is realized by typical bottleneck adapter. Blue and green parts on the left represent frozen and learnable modules.

low-rank structure. Detailed implementations and discussions of the LoRA structure are provided in the supplementary material.

4 Empirical Validation

4.1 Vision-Language Retrieval on CLIP

Datasets

We use Laion (Schuhmann et al., 2021), YFCC (Thomee et al., 2016), and MS-COCO (Lin et al., 2014) for CLIP backbones. We randomly choose 0.1 million subset from Laion and YFCC to make light finetuning. We use 10K, 60K, and 5K evaluation sets for Laion, YFCC, and MS-COCO, respectively.

Settings

We use CLIP (pretrained on Laion) with ViT-B/16 and ViT-L/14 as backbone[1] and set three finetuning settings: 1) **Regular** uses Laion for both finetuning and evaluation; 2) **Zero-shot** finetunes and validates the pretrained model on different datasets (e.g., finetuning on Laion and validating on YFCC or MS-COCO); 3) **Adaptation** finetunes and validates the model on the same data but different from pretraining dataset (e.g., finetuning and testing on YFCC). In addition, we also include the model evaluated on Laion but finetuned on YFCC, which is not a common scenario but for a comprehensive validation. As image retrieval is more commonly used for practice (e.g., searching engine) compared with text retrieval, we only report image retrieval results for real-world large-scale datasets (Laion, YFCC). We still report both image and text retrieval for COCO, which is a typical evaluation for this small dataset.

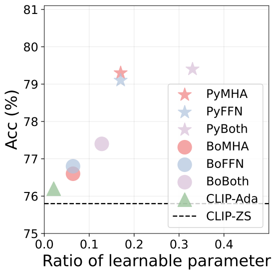
Comparison Methods

We include zero shot performance on CLIP (**CLIP-ZS**) and CLIP-Adapter (Gao et al., 2021) (**CLIP-Ada**) as two baselines. We refer the bottleneck adapter/pyramid ensemble as **Bo/Py**, respectively. Bo and Py can be used after multi-head attention (**MHA**), feedforward (**FFN**), or **Both**. Thus, there are several combinations, such as pyramid emsemble with multi-head attention (**PyMHA**), bottleneck adapter with feedforward (**BoFFN**), etc. Detailed combinations are show in Fig. 4 and Fig. 5. We refer the multi-scale attention/multi-scale attention with LoRA adoptation as **MSA/MSA-Lo**, respectively. All comparisons are separeted into two groups for a clear analysis as below.

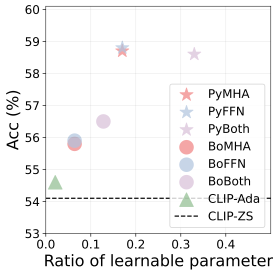
Bottleneck Adapter/Pyramid Ensemble

Fig. 4 shows the comparisons using ViT-B/16 CLIP. Y-axis means the Top1 accuracy and X-axis represents the ratio of additional learnable parameter compared with original CLIP. We conclude 1) both Bo/Py achieve sizable performance gains compared with CLIP-ZS and CLIP-Ada. 2) improving Laion performance is harder compared with that of YFCC (e.g., (b)/(d) have larger improvements than (a)/(c)). 3) Py-family ensemble is generally better than Bo-family. 4) FFN and MHA ensembles have comparable results. 5) adding ensemble after both MHA and FFN always outperforms each

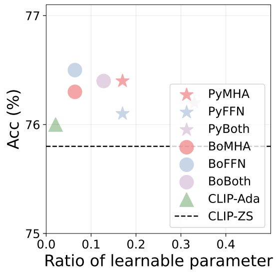
[1] https://github.com/openai/CLIP



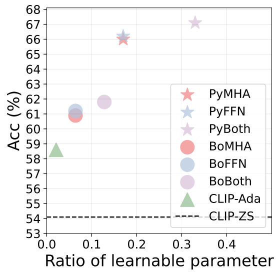
(a) Image retrieval on ViT-B/16 CLIP: model is fine-tuned and tested both on Laion (**regular** setting) with several ensemble strategies and baselines.



(b) Image retrieval on ViT-B/16 CLIP: model is fine-tuned on Laion and tested on YFCC (**zero-shot** setting) with several ensemble strategies and baselines.



(c) Image retrieval on ViT-B/16 CLIP: model is fine-tuned on YFCC and tested on Laion (see Sec. 4.1) with several ensemble strategies and baselines.



(d) Image retrieval on ViT-B/16 CLIP: model is fine-tuned and tested both on YFCC (**adaptation** setting) with several ensemble strategies and baselines.

Figure 4: Evaluation of image retrieval using ViT-B/16 CLIP. Four evaluation settings are tested based on Laion and YFCC datasets for finetuning or testing. Two ensemble strategies, bottleneck adapter and pyramid ensemble, are tested by being deployed after multi-head attention (MHA), feedforward (FFN), or both. The zero-shot evaluation using pretrained CLIP without finetuning (CLIP-ZS) and CLIP adapter (CLIP-Ada) are used as baselines. Y-axis means the Top1 retrieval accuracy and X-axis denotes the ratio of additional learnable parameter size to the original CLIP. Several ensemble designs generally outperform two baselines.

individual one except for the setting (c). It may be caused by using YFCC to finetune but testing on Laion which is also used for pretraining. 6) Compared with CLIP, the number of additional parameter for all settings is relatively small. The most expensive setting PyBoth requires around 30% additional learnable parameters but others still derive promising improvement.

Fig. 5 shows the ViT-L/14 CLIP results. Ensemble on larger model performs differently compared with a smaller one: 1) improving Laion performance is even harder as it originally pretrained on Laion and less improvement potential left in larger CLIP. Performance gain in (a) and (c) is smaller than ViT-B/16 and performance may drop sometimes after finetuning. 2) Ensemble on FFN is better than MHA here while they are almost comparable in ViT-B/16. Please note even if our adapter ensemble requires more additional parameters compared with the typical adapter (shown in x-axis in Fig. 4 and Fig. 5), our exploration uses an very limited 0.1M data, which is 1/4000 of the original 400M pretraining Laion data and a few epochs (5 in our cases). We use the 256/128 batch size for ViT-B/16 and ViT-L/14 CLIP. They are more memory efficient, unlike recently methods using a much larger batch size (Radford et al., 2021). Overall, we observe significant improvements on various settings, validating our adapter ensemble is effective for vision-language retrieval based on the pretrained CLIP. The parameter efficiency solution and corresponding discussion are provided next.

ViT-B/16 CLIP: Image Retrieval						
Setting	CLIP	w/o MSA	V-MSA	L-MSA	MSA	MSA-Lo
Regular	75.8	77.8	79.0	78.9	**79.6**	78.7
Zero-shot	54.1	57.3	**59.7**	56.5	58.6	58.8
Adaptation	54.1	62.3	67.7	61.0	**67.9**	65.3
ratio (%)	-	5.3	37.3	37.3	74.7	**2.2**

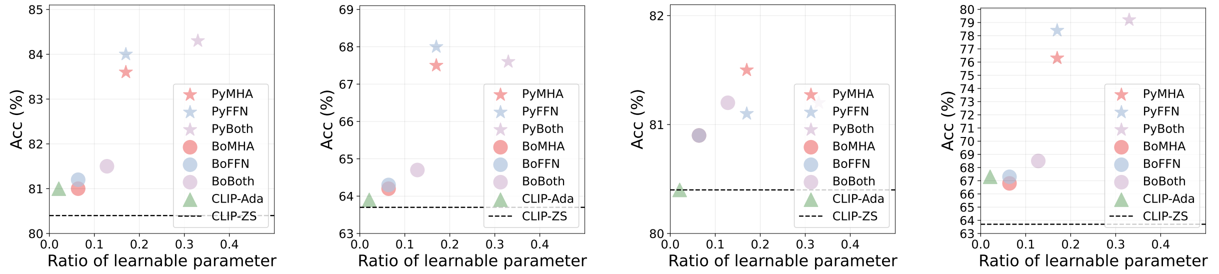
Table 1: MSA evaluation on Regular, Zero-shot, and Adaptation settings using ViT-B/16 CLIP. The ratio of learnale parameter compared with backbone is in the last row. Three ablations, w/o MSA, V-MSA, and L-MSA, are provided. MSA-Lo obtains competitive performance with much less additional parameters.

ViT-L/14 CLIP: Image Retrieval						
Setting	CLIP	w/o MSA	V-MSA	L-MSA	MSA	MSA-Lo
Regular	80.1	81.6	83.6	83.3	**84.3**	83.8
Zero-shot	63.7	64.7	**69.6**	65.3	68.0	67.8
Adaptation	63.7	67.2	**79.2**	69.2	78.6	78.4
ratio (%)	-	5.3	37.3	37.3	74.7	**2.2**

Table 2: MSA evaluation on Regular, Zero-shot, and Adaptation settings using ViT-L/14 CLIP. The ratio of leranable parameter compared with backbone is in the last row. Three ablations, w/o MSA, V-MSA, and L-MSA, are provided. MSA-Lo obtains competitive performance with much less additional parameters.

MSA Performance

Tab. 1 2 shows the MSA results with different settings on ViT-B/16 and ViT-L/14 backbones. **CLIP-ZS** is the pretrained CLIP zero-shot evaluation. **w/o MSA** is the model without MSA. **V-MSA**, **L-MSA**, and **MSA** represent using MSA on vision only, language only, both towers, respectively. **MSA-Lo** means MSA with LoRA adoption. We test on Reg-



(a) Image retrieval on ViT-L/14 CLIP: model is finetuned and tested both on Laion (**regular** setting) with several ensemble strategies and baselines.

(b) Image retrieval on ViT-L/14 CLIP: model is finetuned on Laion and tested on YFCC (**zero-shot** setting with several ensemble strategies and baselines.

(c) Image retrieval on ViT-L/14 CLIP: model is finetuned on YFCC and tested on Laion (see Sec. 4.1) with several ensemble strategies and baselines.

(d) Image retrieval on ViT-L/14 CLIP: model is finetuned and tested both on YFCC (**adaptation** setting) with several ensemble strategies and baselines.

Figure 5: Evaluation of image retrieval using ViT-L/14 CLIP. Four evaluation settings are tested based on Laion and YFCC datasets for finetuning or testing. Two ensemble strategies, bottleneck adapter and pyramid ensemble, are tested by being deployed after multi-head attention (MHA), feedforward (FFN), or both. The zero-shot evaluation using pretrained CLIP without finetuning (CLIP-ZS) and CLIP adapter (CLIP-Ada) are used as baselines. Y-axis means the Top1 retrieval accuracy and X-axis denotes the ratio of additional learnable parameter size to the original CLIP. Several ensemble designs generally outperform two baselines.

ular, Zero-shot, and Adaptation settings and the ratio of additional parameter to the original backbone is shown in the last row. Our MSA outperforms the zero-shot baseline and the ablated model for all settings. Further, employing MSA on vision tower is more effective than language tower and sometimes even better than using MSA on both. The MSA involves more additional parameter, yet, the MSA with LoRA (MSA-Lo) significantly reduces the number of additional parameters and still obtains competitive performance. It ensures the parameter efficiency for our adapter ensemble strategy. Please note, herein, we mainly consider the parameter aspect for the model efficiency. It is directly related to disk space instead of latency and flops which are mainly for model compression and out of the scope of this study. In addition, we also evaluate our MSA strategy with its ablated models using MS-COCO dataset on a zero-shot retrieval setting (Tab. 3).

Ablation

We make ablation analysis using MS-COCO dataset on zero-shot evaluation. Specifically, we remove different branches in our MSA to validate the effectiveness of the multi-scale strategy (Tab. 4). As the large-scale branch represents the full attention score matrix, we remove middle and small branches to observe the performance changes. We find that adding each of them benefits the model to achieve better performance and three scales working together in an ensemble fashion obtains the best performance gain.

MS-COCO Zero-Shot Image Retrieval					
Backbone	CLIP	w/o MSA	V-MSA	L-MSA	MSA
ViT-B/16	32.7	34.5	35.2	34.3	**35.2**
ViT-L/14	35.3	35.9	38.7	37.2	**38.8**

Table 3: MSA zero-shot evaluation of MS-COCO on ViT-B/16 and ViT-L/14 CLIP. The CLIP zero-shot baseline and three ablated models, without MSA (w/o MSA), vision-only MSA (V-MSA), and language-only MSA (L-MSA), are also provided.

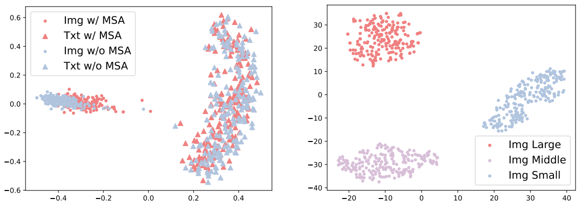
MSA Ablation for MS-COCO Zero-Shot Retrieval				
Backbone	MSA-L	MSA-L+M	MSA-L+S	MSA-L+M+S
ViT-B/16	34.1	34.9	34.9	**35.2**
ViT-L/14	37.1	38.3	38.4	**38.8**

Table 4: MSA ablation study by removing branches for different scales on zero-shot MS-COCO evaluation. Large, middle, and small scales are referred by "L", "M", and "S", respectively. Our complete MSA obtains the best performance.

4.2 Further Analysis

Feature Visualization

The MSA provides diverse visual and language representations from different scales, which benefits the ensemble strategy. To provide a better intuition of the ensemble operation, we use PCA to show the feature distribution variations between MSA and w/o MSA on YFCC (Fig. 6a). Compared with model without MSA, the vision and language representations are further pulled closer by MSA



(a) PCA visualization of model features with and without MSA.

(b) t-SNE visualization for feature distributions of different scale models.

Figure 6: Visualization analysis of feature distributions of MSA (Fig. 6a) and different branches (Fig. 6). Features are extracted from ViT-L/14 CLIP finetuned on YFCC dataset.

operation which improves the cross-modal retrieval performance. In addition, we use t-SNE to show the features from large, middle, and small scales (Fig. 6b). They are clearly separated and provide diverse features, benefiting the ensemble strategy.

Due to the limited space, we leave retrieval visualizations (see Sec. A.5) and backbone generalization results (see Sec. A.3) in the appendix.

5 Related Works

Vision-language retrieval is pioneered by VSE++ (Faghri et al., 2017), using hard-negative mining. SCAN (Lee et al., 2018) designs cross-modal encoding for fine-grained features. VSRN (Li et al., 2019) uses graph and recurrent networks to reason visual semantics. Large-scale pretraining boosts the performance using massive web data (Radford et al., 2021). Recent works (Jia et al., 2021; Kim et al., 2021; Ramesh et al., 2022; Saharia et al., 2022) explore different strategies for pretraining such as CoCa (Yu et al., 2022) jointly using retrieval and captioning loss and BLIP (Li et al., 2022) utilizing cross-modal encoding. They significantly improves retrieval performance yet requires much more resource. Herein, we explore an efficient ensemble, combined with adapter, to further enhance the pretrained vision-language backbones for retrieval tasks.

Ensemble leverages on diverse base learners to achieve crowd intelligence. It is seen as a weighting/voting strategy. Ensemble is simple yet effective for traditional machine learning (Dietterich, 2000; Sagi and Rokach, 2018). It is also applied to neural networks (Ganaie et al., 2021). Dropout (Srivastava et al., 2014) as a common way to avoid overfitting can be interpreted from an ensemble aspect. Different applications using ensemble derive promising performance compared with individual model (Li et al., 2019; Lee et al., 2018). Recent model soups (Wortsman et al., 2022) manages to integrate several checkpoints of a large pretrained models in an ensemble fashion to boost final performance. Different from them, our study focuses on introducing ensemble into current large-scale backbones, combined with adapter, to improve the pretrained model in an efficient manner.

Adapter is originally proposed for efficient finetuning of language model (Houlsby et al., 2019). It leverages on the large-scale pretrained models and relaxes a few learnable parameters which is friendly to limited downstream data. Several parameter-efficient strategies are designed to relieve the finetuning difficulties of pretrained language models (Hu et al., 2021; Karimi Mahabadi et al., 2021; Eichenberg et al., 2021; He et al., 2021). This insight is also adopted into vision and vision-language fields to benefit various pretrained models for several downstream applications (Chen et al., 2022b; Zhang et al., 2021; Sung et al., 2022; Gao et al., 2021; Chen et al., 2023; Zheng et al., 2023; Upadhyay et al., 2023; Zhang et al., 2023a,b). In our study, we are inspired by the adapter insight. However, instead of injecting one set of learnable parameters, we propose to supplement a few sets of learnable parameters with diverse focuses (e.g. multi-scale attention) for efficient ensemble on pretrained large-scale models.

6 Conclusion

Our curiosity lies in exploring how traditional machine learning techniques, typically used for small-sized models, can be leveraged to benefit recent large-scale pretrained vision-language models. We identify *adapter ensemble* as an ideal fusion point, effectively finetuning large-scale models while seamlessly integrating small-sized methodologies. Through a proof-of-concept study, we validate the ensemble adapter efficacy. We then demonstrate its effectiveness for vision-language retrieval on different settings. Specifically, a multi-scale attention (MSA) is designed to benefit ensemble operation. Furthermore, to address the potential increase in parameter requirements brought by the ensemble, we integrate the LoRA for MSA, significantly reducing the parameter overhead. Our empirical results showcase the ensemble capacity to enhance the performance of large-scale pretrained models, achieving efficiency in data, parameter, and finetuning budgets.

7 Limitations

This work proposes to explore ensemble, a typical machine learning technique, in current large-scale model era. We mainly take CLIP backbone as a study case and make evaluation on cross-modal retrieval task. Due to the limited computational resource, we do not include other model backbones and tasks like language models or multi-modal models. However, the proposed adapter ensemble can be easily extended to other scenarios and we leave it into our future work.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*.

Guangyi Chen, Xiao Liu, Guangrun Wang, Kun Zhang, Philip HS Torr, Xiao-Ping Zhang, and Yansong Tang. 2023. Tem-adapter: Adapting image-text pretraining for video question answer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13945–13955.

Jun Chen, Han Guo, Kai Yi, Boyang Li, and Mohamed Elhoseiny. 2022a. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18030–18040.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. 2022b. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*.

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*, pages 248–255. Ieee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Constantin Eichenberg, Sidney Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. 2021. Magma–multimodal augmentation of generative models through adapter-based finetuning. *arXiv preprint arXiv:2112.05253*.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*.

Mudasir A Ganaie, Minghui Hu, et al. 2021. Ensemble deep learning: A review. *arXiv preprint arXiv:2104.02395*.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. 2021. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

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*.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. *CoRR*, abs/2102.05918.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5583–5594. PMLR.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *Proceedings of the European conference on computer vision (ECCV)*, pages 201–216.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*.

Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. 2019. Visual semantic reasoning for image-text matching. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4654–4662.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Ron Mokady, Amir Hertz, and Amit H Bermano. 2021. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*.

Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. 2022. Slip: Self-supervision meets language-image pre-training. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 529–544. Springer.

Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. 2022. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Lior Rokach. 2010. Ensemble-based classifiers. *Artificial intelligence review*, 33(1):1–39.

Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5227–5237.

Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. 2022. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17918–17928.

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73.

Uddeshya Upadhyay, Shyamgopal Karthik, Massimiliano Mancini, and Zeynep Akata. 2023. Probvlm: Probabilistic adapter for frozen vison-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1899–1910.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2021. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*.

Yi Zhang, Ce Zhang, Xueting Hu, and Zhihai He. 2023a. Unsupervised prototype adapter for vision-language models. *arXiv preprint arXiv:2308.11507*.

Yi Zhang, Ce Zhang, Zihan Liao, Yushun Tang, and Zhihai He. 2023b. Bdc-adapter: Brownian distance covariance for better vision-language reasoning. *arXiv preprint arXiv:2309.01256*.

Kecheng Zheng, Wei Wu, Ruili Feng, Kai Zhu, Jiawei Liu, Deli Zhao, Zheng-Jun Zha, Wei Chen, and Yujun Shen. 2023. Regularized mask tuning: Uncovering hidden knowledge in pre-trained vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11663–11673.

A Supplementary Material

A.1 Supplementary MS-COCO Performance

We supplement the MSA-Lo and zero-shot text retrieval performance on MS-COCO dataset using both ViT-B/16 and ViT-L/14 CLIP. Specifically, we augment the image retrieval table (Tab.2 in the main draft) with MSA-Lo in Tab. 8, and we newly provide text retrieval results in Tab. 5. We also provide text retrieval ablation study in Tab. 6. We observe the consistent improvement compared with baselines and different ablated models and draw the similar conclusions as our main draft.

MS-COCO Zero-Shot Text Retrieval						
Backbone	CLIP	w/o MSA	V-MSA	T-MSA	MSA	MSA-Lo
ViT-B/16	51.7	53.5	53.5	54.5	54.9	54.7
ViT-L/14	56.1	56.7	57.8	59.2	59.5	59.4

Table 5: MSA zero-shot text retrieval evaluation of MS-COCO on ViT-B/16 and ViT-L/14 CLIP.

MSA Ablation for MS-COCO Zero-Shot Text Retrieval				
Backbone	MSA-L	MSA-L+M	MSA-L+S	MSA-L+M+S
ViT-B/16	53.7	54.5	54.5	**54.9**
ViT-L/14	57.8	59.1	59.0	**59.5**

Table 6: MSA ablation study by removing branches for different scales on zero-shot MS-COCO text retrieval.

A.2 Implementation Details

We provide more implementation details for our adapter ensemble exploration. We run our experiments on 8 V100 GPUs. For bottleneck adapter used in our experiments, we consistently set 128 as hidden dimension. To maintain the near-identity initialization for finetuning the pretrained model, we initialize the values of weights using 0/1e-3 for means/variances values without bias for the bottleneck adapter. For the pyramid structure of our MSA, we initialize the sub-matrix, corresponding to the large-scale branch, as identity matrix and the other values using 0/1e-3 for means/variances. For LoRA structure in MSA-Lo, we add it parallel to the attention module for large-scale branch, and after the attention module for middle-scale and small-scale branches, setting 16 as low-rank hidden dimension. The outputs of three branches are added as an ensemble operation. In addition, we also use the ensemble strategy for the LoRA structure after FFN. Specifically, we use a shared matrix A and three different matrices B, and three outputs are added together as an ensemble operation. For all backbones used in our experiments, we follow their original finetuning configurations to conduct our adapter ensemble finetuning, except for the available finetuning data and epochs (always 0.1M available data and 5 epochs in our study).

Herein, we also discuss the MSA-Lo implementation for the potential latency issue caused by ensemble operations. We simply use the LoRA structure after FFN as an example. Since several different B matrices need multiple forward computations, we concatenate them along with the feature dimension the achieve the parallel computation. In this way, multiple branches of the ensemble can be processed efficiently. The time consumption comparison of the FFN ensemble operation in one MSA-Lo block is shown in Tab. 7. "One-branch" means a typical LoRA baseline. "Three-branch" means the ensemble in three-time forward fashion. "Three-branch (parallel)" is the parallel implementation of the ensemble. Results are based on 10 runs average. We find leveraging on parallel implementation, the ensemble strategy can be achieved in an efficient fashion without too much additional latency cost.

One-branch	Three-branch	Three-branch (parallel)
1.34e-4	3.52e-4	1.58e-4

Table 7: Time consumption comparison of LoRA in one FFN block of MSA-Lo.

MS-COCO Zero-Shot Image Retrieval						
Backbone	CLIP	w/o MSA	V-MSA	L-MSA	MSA	MSA-Lo
ViT-B/16	32.7	34.5	35.2	34.3	35.2	35.2
ViT-L/14	35.3	35.9	38.7	37.2	38.8	38.6

Table 8: MSA zero-shot image retrieval evaluation of MS-COCO on ViT-B/16 and ViT-L/14CLIP.

A.3 Backbone Generalization

Besides of the CLIP architecture, we further consider other backbones to validate the generalizability of the proposed adapter ensemble strategy. Specifically, SLIP (Mu et al., 2022) uses self-supervised learning to help vision-language pretraining. It further improve the cross-modal modeling capacity compared with CLIP. We follow its original paper to use a linear probing to evaluate image classification on Imagenet (Deng et al., 2009). We also use a 0.1M Imagenet subset

Image classification on SLIP (ViT/B16)				
Pretraining Data	Zero-shot	Linear	w/o MSA	w/ MSA
CC3M	23.0	47.5	51.0	**51.4**
CC12M	40.7	55.8	63.3	**64.3**

Table 9: Image classification results of SLIP based on CC3M and CC12M pretraining dataset. We compare our MSA with zero-shot, linear baselines and the ablated w/o MSA model. Our MSA shows the generalizability on SLIP backbone.

Image classification results on Beit V2				
Pretraining Data	Model	Linear	w/o MSA	w/ MSA
Imagenet1K	ViT-B	55.3	66.3	**68.6**
	ViT-L	63.8	69.4	**72.0**

Table 10: Image classification results of Beit V2 based on ViT-B and ViT-L backbones. We compare our MSA with zero-shot, linear baselines and the ablated w/o MSA model. Our MSA shows the generalizability on Beit V2 backbone.

to finetune the pretrained backbone 5 epochs for our ensemble strategy. Tab. 9 shows the comparisons of MSA ensemble with baselines on SLIP with different pretraining datasets (e.g., CC3M and CC12M (Changpinyo et al., 2021)). The zero-shot is evaluated by using prompt template while others using typical label prediction.

Beit V2 (Peng et al., 2022) is a backbone only for vision domain. Herein, we also include it to test the generalizability of our ensemble strategy on visual only task. We use a 0.1M Imagenet subset to finetune the pretrained backbone 5 epochs. Since the Beit V2 is pretrained in self-supervised fashion, it cannot perform zero-shot evaluation without finetuning. Similar to SLIP, we make a linear probing classifier as a baseline. Tab. 10 shows the comparisons of MSA ensemble with baselines on different backbones. We observe the proposed adapter ensemble is a general finetuning strategy for different backbones.

A.4 More Feature Distribution Visualizations

We provide more feature distribution visualizations for our multi-scale attention (MSA) on different settings. The **Regular** setting finetunes and evaluates the pretrained model on Laion dataset using CLIP backbone. Since it is a more challenging setting and its performance gain is not as much as other settings, we do not observe significant feature variations on this setting. Therefore, we mainly show **Adaptation** and **Zero-shot** settings for feature distribution visualization. Like our main draft, we show image and text feature distributions from models w/ and w/o MSA (each subfigure (a)), and image feature distributions of different scales (each subfigure (b)). Fig. 9 shows the zero-shot setting visualization on ViT-L/14 CLIP. Fig. 10 shows the adaptation setting visualization on ViT-B/16 CLIP. Fig. 11 shows the zero-shot setting visualization on ViT-B/16 CLIP. We find the adaptation setting shows significant feature variations, which indicates the features from different modalities become closer with each other and improve the retrieval performance.

A.5 Retrieval Visualizations

Retrieval Visualization

We show retrieval results to compare the models w/ and w/o MSA. In Fig. 7, MSA obtains the correct Recall@1 image retrieval in the first five samples but fails in the last. We observe compared with w/o MSA, MSA retrieval better matches with the query at different scales. For example, in the first example, MSA retrieves the image with correct cat object and street corner background while w/o MSA retrieves house and chair as background which are incorrect. In Fig. 8, MSA successes in the first five samples and fails in the last. Similarly, MSA matches the query with more details for text retrieval. For example, another standing woman on the edge of the image is captured by our method in the first example. The small zebra instead of giraffe is accurately attended in the second. The water background in both the second and third examples are captured by MSA but missed by w/o MSA.

We show more cross-modal retrieval visualizations on MS-COCO dataset using our model (w/ MSA) and w/o MSA. We show text retrieval visualizations in Fig. 12, where the image query is shown on the left and text retrieval with green color means the groundtruth retrieval. Our model obtains the correct results on Recall@1 in subfigure (a), (b), and (c), where our MSA captures more fine-grained patterns from different scales. For example, MSA finds the “brick” element in (b) and the “bathroom” element in (c) for cross-modal matching in a small scale but w/o MSA ignores them. w/o MSA derives the correct results on Recall@1 in subfigure (d), (e), and (f). However, MSA also finds reasonable retrievals. For example, in (d), our MSA captures the “BMW” information which is shown in the middle of figure at a very small scale and provides the retrieval accordingly. Similarly, in (f), the fine-grained visual element “jet way” is considered by MSA for retrieval but w/o MSA ignores it. Fig. 13 and Fig. 14 show the image retrieval visualizations, where text query is shown on the top and image with green box means the groundtruth retrieval. In Fig. 13, our model (w/ MSA) obtains the correct retrieval on Recall@1 with more details. For example, in subfigure (a), our model captures the detailed color information of the clock tower and finds the most accurate retrieval while w/o MSA only finds it at Top3. In Fig. 14, w/o MSA derives the correct retrieval on Recall@1. However, our model also retrieve promising results at Top1 compared with the groundtruth. In addition, for all top five retrievals, our model generally obtains more reasonable results. For example, in subfigure (a), w/ MSA finds motor cycles in all five retrievals but w/o MSA misses this component at Top4.

Figure 7: MS-COCO zero-shot image retrieval examples for ViT-B/16 CLIP backbone. MSA and w/o MSA represent if the model uses our multi-scale strategy. Caption queries are shown on the top and we show the Top1 image retrieval of both MSA and w/o MSA models. Our MSA obtains correct retrieval for the first five examples (in green) but fails at the last one (in red).

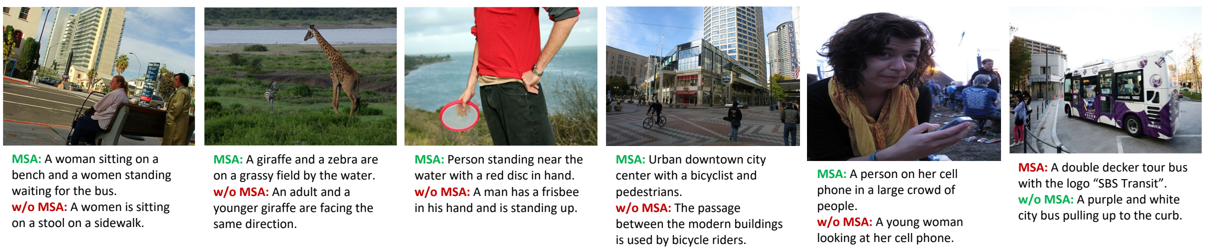
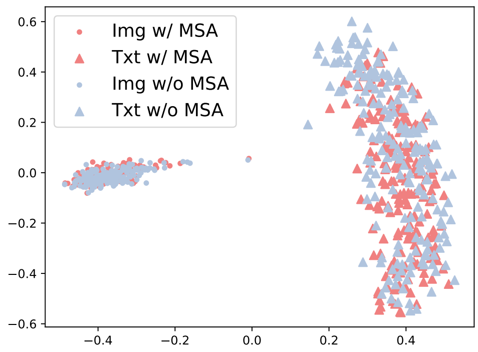
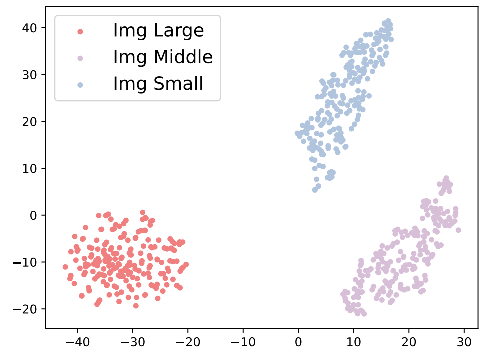


Figure 8: MS-COCO zero-shot text retrieval examples for ViT-B/16 CLIP backbone. MSA and w/o MSA represent if the model uses our multi-scale attention strategy. Image queries are shown on the top and we show the Top1 text retrieval of both MSA and w/o MSA models. Our MSA obtains correct retrieval for the first five examples (in green) but fails at the last one (in red).

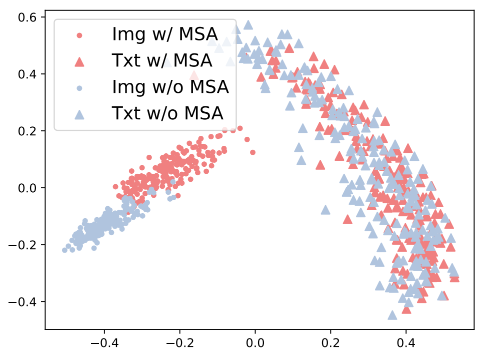


(a) Distribution visualization of model w/ and w/o MSA.

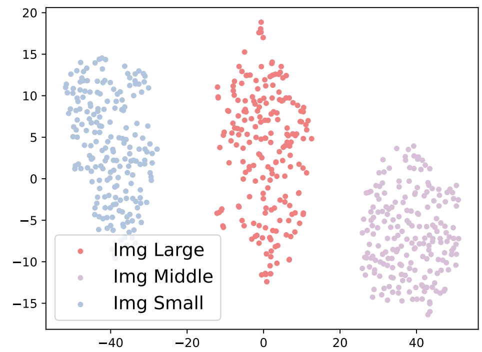


(b) Distribution visualization of different scales feature.

Figure 9: YFCC feature visualization on **Zero-shot** setting using ViT-L/14 CLIP.

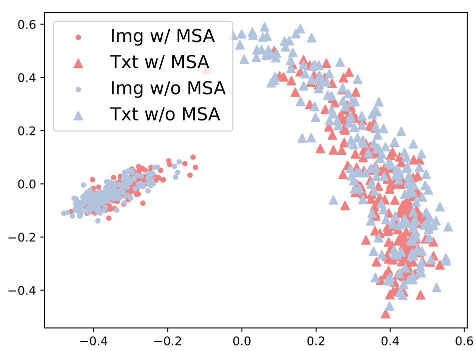


(a) Distribution visualization of model w/ and w/o MSA.

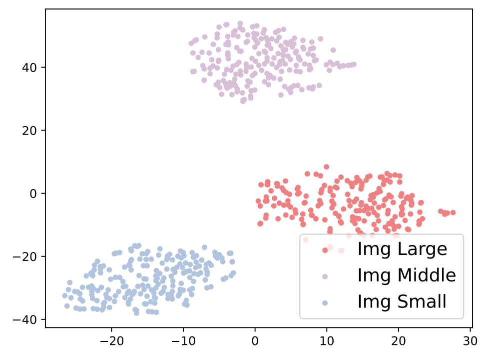


(b) Distribution visualization of different scales feature.

Figure 10: YFCC feature visualization on **Adaptation** setting using ViT-B/16 CLIP.



(a) Distribution visualization of model w/ and w/o MSA.



(b) Distribution visualization of different scales feature.

Figure 11: YFCC feature visualization on **Zero-shot** setting using ViT-B/16 CLIP.

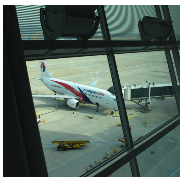
(a)

(b)

(c)

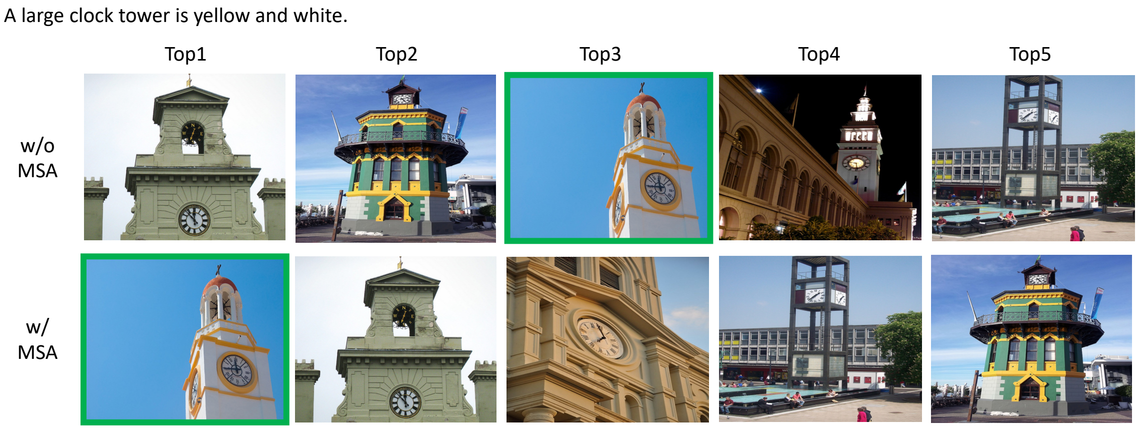
(d)

(e)



(f)

Figure 12: Text retrieval visualization on MS-COCO using w/ and w/o MSA models. Our model (w/ MSA) obtains the correct retrieval on Recall@1 in (a), (b), and (c). w/o MSA derives the correct retrieval on Recall@1 in (d), (e), and (f). Image query is shown on the left and text with green color means the groundtruth retrieval result.



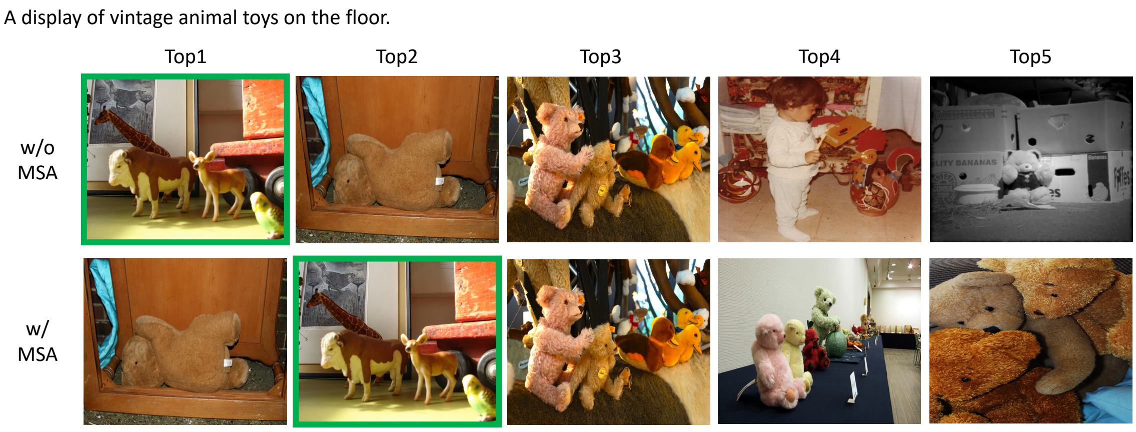
(a)

(b)

(c)

Figure 13: Image retrieval visualization on MS-COCO. We compare the models w/ and w/o MSA strategy. For these three samples, our model (w/ MSA) obtains the correct retrieval on Recall@1. Text query is shown on the top and image retrieval with green box means the groundtruth retrieval result.

(a)



(b)

(c)

Figure 14: Image retrieval visualization on MS-COCO. We compare the models w/ and w/o MSA strategy. For these three samples, w/o MSA derives the correct retrieval on Recall@1. Text query is shown on the top and image retrieval with green box means the groundtruth retrieval result.