MMA++: Effective Multi-Modal Adaptation for Vision-Language Models

Lingxiao Yang, *Member, IEEE*, Ru-Yuan Zhang, Yanchen Wang, and Xiaohua Xie, *Member, IEEE*

***Abstract*—Large scale pre-trained Vision-Language Models (VLMs) have shown good generalization capabilities across diverse downstream tasks. However, adapting such large-scale models to few-shot generalization scenarios remains challenging due to the trade-off between preserving general knowledge and incorporating task-specific information. In this paper, we propose MMA++, an advanced and effective Multi-Modal Adapter framework for parameter-efficient VLM adaptation. Unlike prior works that independently inject adapters into each modality or uniformly across layers, MMA++ performs a dataset-level analysis to identify discriminative and generalizable features, and selectively applies adapters to the higher layers of both vision and text encoders. To bridge the modality gap, we further propose a shared feature projection space that enhances alignment between modalities. Beyond architecture design, we identify the fusion scale α—which controls the strength of adapter integration—as a key factor in few-shot generalization. We empirically and theoretically demonstrate that α should not be static, but adapted based on training data size. To reduce the effort of tuning this value across different datasets, we propose the α-*consistency* framework, consisting of: (1) a consistency training strategy under varying fusion scales; and (2) an α-decoupling strategy that uses a larger fusion scale during training and a smaller one at inference to account for sample size mismatch. We evaluate MMA++ on a wide range of few-shot generalization tasks, including base-to-novel generalization, cross-dataset transfer, and domain generalization. Our method consistently achieves leading performance.**

***Index Terms*—Multi-modal adapter, vision-language models, few-shot adaptation, scale-consistency training, train-test decoupling.**

I. Introduction

MULTI-MODAL learning has become a cornerstone of modern artificial intelligence, especially with the emergence of large-scale Vision-Language Models (VLMs) [1], [2], [3], [4]. These models learn to align visual and textual representations by training on massive web-scale image-text datasets [2], [5], demonstrating strong generalization across a wide range of downstream tasks. However, their large parameter sizes and dual-encoder architectures present significant challenges for adaptation—particularly in few-shot or distribution-shifted settings—where fully fine-tuning the entire model is both computationally expensive and prone to overfitting.

To adapt large-scale pre-trained VLMs such as CLIP to downstream tasks, prompt engineering [2] has emerged as a widely adopted strategy. It involves crafting input patterns or textual templates that guide the model to produce task-relevant outputs. For instance, CLIP uses a series of handcrafted prompts (e.g., "a photo of a <category >", "a bad photo of a < category>") to generate category-level features via the text encoder. These features are then compared with image representations from the vision encoder to perform classification. However, manually designing effective prompts is time-consuming and requires domain expertise. To overcome these limitations, recent works propose learnable prompt tuning methods that inject trainable tokens into either the text encoder [6], [7], [8], [9], [10], [11], the image encoder [12], [13], or both [14], [15], [16], [17], [18], [19], [20]. In these approaches, only the added prompts are updated during training, while the backbone VLM remains frozen. This design significantly reduces computational costs and mitigates overfitting risks, making prompt learning a practical and efficient paradigm for domain adaptation.

In addition to prompt learning, another prominent line of research explores adapter-based tuning, where lightweight, trainable modules—referred to as *adapters*—are inserted into pre-trained models to facilitate downstream adaptation [21], [22], [23], [24], [25]. Unlike prompt tuning, which operates at the input level, adapters act on intermediate features, enabling flexible fusion and modulation of representations within the network. These modules are typically shallow neural blocks (e.g., bottleneck projections or low-rank layers) that are optimized while

Received 11 July 2025; revised 2 February 2026; accepted 21 April 2026. Date of publication 25 May 2026; date of current version 6 August 2026. This work was supported in part by NSFC under Grant 62206316, Grant 62072482, Grant U22A2095, and Grant 32100901, in part by the Key-Area Research and Development Program of Guangzhou under Grant 202206030003, in part by the NSF of Shanghai under Grant 21ZR1434700, and in part by the Project of Guangdong Provincial Key Laboratory of Information Security Technology under Grant 2023B1212060026. Recommended for acceptance by Y. Naoto. *(Corresponding author: Xiaohua Xie.)*

Lingxiao Yang is with the School of Systems Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China (e-mail: yanglx9@mail.sysu.edu.cn).

Ru-Yuan Zhang is with the Beijing Key Laboratory of Behavior and Mental Health, School of Psychological and Cognitive Sciences, Peking University, Beijing 100871, China, also with the IDG/McGovern Institute for Brain Research, Peking University, Beijing 100871, China, and also with the Key Laboratory of Machine Perception (Ministry of Education), Peking University, Beijing 100871, China (e-mail: ruyuanzhang@pku.edu.cn).

Yanchen Wang is with Columbia University, New York, NY 10027 USA (e-mail: yw4503@columbia.edu).

Xiaohua Xie is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China, also with the Guangdong Province Key Laboratory of Information Security Technology, Guangzhou 510006, China, and also with the Key Laboratory of Machine Intelligence and Advanced Computing, MOE, Guangzhou 510275, China (e-mail: xiexiaoh6@mail.sysu.edu.cn).

Code is available at https://github.com/ZjjConan/VLM-MultiModalAdapterPP.

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2026.3691448, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2026.3691448

keeping the backbone frozen, thereby preserving general knowledge from large-scale pretraining and reducing overfitting risks. Recent studies have validated the effectiveness of adapters in VLMs. For instance, CLIP-Adapter [25] and AdaptFormer [23] leverage additive fusion, combining outputs from the pre-trained model and the adapter. Like prompt tuning, these methods are parameter-efficient, as only the adapters are updated during training. A notable advantage of adapter-based approaches is their architectural generality—they can be seamlessly integrated into diverse backbones, including ResNets [26], Vision Transformers [27], and Swin Transformers [28], without requiring substantial modifications to the original model structure.

Despite their effectiveness, most existing adapter designs treat each modality independently, overlooking the deep cross-modal interactions that are fundamental to vision-language models (VLMs). In dual-encoder architectures such as CLIP [2], visual and textual representations are processed separately and aligned only through the final projection layers. Applying independent adapters to each modality—without explicitly modeling shared semantics or alignment dynamics—can limit the model's ability to capture task-specific correspondences. Furthermore, prior works often insert adapters uniformly across all transformer layers [21], [23], ignoring the distinct functional roles associated with different depths. As analyzed in this study, lower layers typically encode generic, transferable features, while higher layers capture task-specific and domain-sensitive information. Uniform adaptation may therefore introduce unnecessary parameter updates and lead to performance degradation, particularly in low-data regimes.

To this end, we propose a novel **M**ulti-**M**odal **A**dapter (**MMA**) architecture for VLMs such that the text and image representations can be better aligned. Our MMA contains independent projection layers to learn task-specific knowledge in both text and vision branches. To promote different modal alignment, we design a unified feature projection layer shared by both modalities. During the tuning, this unified feature space communicates gradients from both modalities to improve the alignment. In addition, we evaluate the discriminability and generalizability of features in both image and text encoders across datasets. This dataset-level classification identifies that higher layer features as task-specific features should be fine-tuned, while lower layer features as pre-trained generalizable features should be frozen. Therefore, we only incorporate our adapters to higher layers. This design circumvents the issue of insufficient statistical information to understand feature characteristics due to limited number of training samples in each dataset.

Besides, recent studies highlight that the feature-level fusion scale α, which controls how adapter features are linearly combined with the pre-trained backbone representations, plays a critical role in downstream tasks [23], [25], [29], [30]. However, existing approaches often rely on empirically chosen such scale, lacking theoretical guidance on how to set or adapt it—especially in few-shot generalization scenarios, where the risks of overfitting and underfitting are exacerbated. This gap underscores the need for a principled understanding of how fusion scale influences generalization behavior.

To bridge this gap, we further propose **MMA++**, a theoretically grounded framework that investigates the relationship between the fusion scale α and the amount of training data, and how this interaction affects the generalization bound. Through both empirical studies and theoretical analysis, we show that α is not a fixed hyperparameter, but should be adaptively adjusted based on the sample size. Building on this insight, we introduce the α-*consistency* framework, which consists of: (1) a training method that promotes consistency between training and testing behaviors under varying fusion scales, and (2) a novel α-decoupling strategy that mitigates the mismatch between training and testing conditions by using a relatively larger α during training and a smaller one at test time. This strategy explicitly accounts for discrepancies in sample size between training and inference. Together, these components offer a principled solution for adapting fusion strength to improve few-shot generalization, supported by both theoretical justification and empirical validation.

In summary, the contributions of this paper are:

- We introduce a dataset-level analysis method to systematically examine feature representations in transformer-based CLIP models, providing insights for designing more effective and efficient adapters for VLMs.
- We propose a novel adapter architecture with modality-specific projection layers that enhance feature representations in both image and text encoders, along with a shared projection layer to improve cross-modal alignment.
- We reveal, through empirical studies and theoretical analysis, the critical role of the fusion scale α in few-shot generalization, which motivates the development of our α-*consistency* framework.
- We design a training strategy within this framework that enforces consistent model behavior across varying fusion scales by leveraging stochastic scale perturbations and α-aware consistency losses.
- We further propose an α-decoupling strategy that reduces the fusion scale at test time, addressing the discrepancy between training and testing conditions and enhancing generalization reliability.
- We integrate MMA and MMA++ into the CLIP model and evaluate them on multiple few-shot generalization tasks, where our methods achieve leading performance.

The preliminary version of this work appeared in [29]. In this extended version, we make several significant improvements and additions. (1) We deepen both the theoretical and empirical understanding of the fusion scale α in few-shot generalization, which motivates the formal introduction of the α-*consistency* framework. (2) We propose a novel training strategy within this framework that promotes consistent model behavior across varying fusion scales. (3) We propose an α-decoupling strategy to explicitly address the discrepancy between training and testing conditions. (4) We perform extensive experiments on a range of few-shot generalization benchmarks, consistently demonstrating state-of-the-art performance and validating the effectiveness of our theoretical and methodological contributions. Compared to our conference version, this extended work offers a more principled approach, deeper theoretical insights, and stronger empirical evidence to guide adapter fusion scale design in VLMs.

II. Related Work

A. Vision-Language Models

Recent advances in VLMs have significantly impacted both fields of natural language processing and image understanding. Prominent models in this field, such as CLIP [2], ALIGN [1], FILIP [3], and LiT [31], utilize self-supervised training strategies on vast web-scale datasets. For example, CLIP [2] is trained on approximately 400 million image-text pairs, while ALIGN [1] is trained on around 1 billion multi-modality data. By training on more large-scale multi-modal data such as LAION-5B [5], these models exhibit good performance across various tasks [32]. Despite their success, effectively adapting these pre-trained VLMs to specific downstream tasks, particularly in few-shot settings, remains a formidable challenge. To address this, numerous studies [6], [7], [14], [15], [16], [17], [18], [19], [25], [29], [30], [33], [34], [35], [36] have been proposed. In this study, we propose a novel multi-modal adapter and its improved version, which are specifically designed to enhance the adaptation of VLMs to few-shot generalization tasks.

B. Efficient Transfer Learning for VLMs

Applying pre-trained models to downstream tasks traditionally involves fine-tuning all parameters of the pre-trained networks [26], [37]. However, as models increase in size, this method becomes computationally prohibitive. Moreover, fine-tuning a large number of parameters raises the risk of overfitting, particularly in few-shot scenarios. A few training-free methods have been proposed to enhance zero-shot performance of VLMs [34], [38], [39], [40]. These methods either leverage detailed texture descriptions as input or utilize dataset-level features as a gallery to enhance final classification. However, these methods often suffer from limited generalization due to the lack of task-specific tuning. To address this, multiple customized training methods have been proposed. For example, DETA++ [41] proposes a unified denoising framework that can mitigate noises in both support and query sets via contrastive relevance aggregation, prototype refinement, and intra-class region swapping. SkipT [42] introduces an efficient task adaptation paradigm that skips unimportant network layers and class tokens during fine-tuning, achieving good generalization performance on downstream tasks. Another line of research on VLMs focuses on parameter-efficient adaptation techniques. These techniques are emerged initially in the NLP community [21], [22], [43], and have subsequently been introduced for VLMs [6], [7], [8], [9], [14], [15], [16], [17], [18], [19], [20], [25], [29], [30], [35], [44], [45], [46], [47]. These approaches primarily fall into two categories: token-based prompt learning and adapter-based methods.

Token-based prompt learning enhances a model's task understanding and adaptability by providing specific instructions to VLMs. For instance, CoOp [6] optimizes a continuous set of prompt vectors in CLIP's language branch to improve few-shot learning. CoCoOp [7] extends this by conditioning prompts on specific image instances. Other significant advancements in this area include methods to capture diverse prompt distributions [44], [48], mitigate overfitting by leveraging pre-trained CLIP as general knowledge [8], [9], [16], [18], construct multi-modal prompts for both image and text branches [14], and enhance text diversity using large language models such as [17], [35]. These developments have significantly improved the adaptation of CLIP, outperforming CoOp and CoCoOp across various metrics. In the domain of adapters, existing methods often employ uni-modal adapters for fine-tuning. For example, Clip-Adapter [25] and Tip-Adapter [34] add an adapter layer after the image encoder. More recently, multi-modal adapters [45] have been introduced for tasks such as text-video retrieval, incorporating adapters after every self-attention and feed-forward MLP module.

Recent advances in VLMs emphasize parameter-efficient alternatives to full fine-tuning, while also revealing the heterogeneous roles across layers in few-shot generalization. Moreover, most adapter-based methods overlook the importance of the feature-level fusion scale α, which controls the relative contribution of adapter features when fused with pre-trained backbone representations. In practice, this scale is often treated as a heuristically tuned hyperparameter without theoretical justification, limiting both interpretability and generalization.

III. Methods

Following most existing studies [6], [7], [9], [14], [15], [16], [17], [18], [35], we base our approach on the pre-trained transformer-based CLIP model [2]. The model employs transformers in both the text and vision encoders. In the following sections, we first introduce some preliminary background on CLIP, and then present our methods: MMA and MMA++.

A. Preliminary

CLIP [2] is a prominent VLM that has received substantial attention in both the natural language processing and computer vision communities. It consists of two encoders: a text encoder $\boldsymbol{T}$ and a vision encoder $\boldsymbol{V}$, which are jointly trained using a contrastive loss [49] on large-scale image-text pairs [2]. The training objective encourages alignment between semantically related image-text pairs while separating unrelated ones. Consequently, CLIP maps both images and textual descriptions into a shared embedding space, enabling it to support a wide range of downstream tasks. More details, to obtain a single image feature $\boldsymbol{x}$, an image $\boldsymbol{I}$ is passed through the image encoder $\boldsymbol{V}$ as follows:

$$\boldsymbol{x}_0 = PatchEmbed(\boldsymbol{I}) \tag{1}$$

$$[\boldsymbol{c}_i, \boldsymbol{x}_i] = \boldsymbol{\mathcal{V}}_i([\boldsymbol{c}_{i-1}, \boldsymbol{x}_{i-1}]) \quad i = 1, 2, \ldots, L \tag{2}$$

$$\boldsymbol{x} = PatchProj(\boldsymbol{c}_L). \tag{3}$$

In this process, the image encoder begins by applying a patch embedding module ($PatchEmbed$), which partitions the input image $\boldsymbol{I}$ into fixed-size patches (e.g., 16×16) and maps them to feature embeddings. A learnable class token $\boldsymbol{c}_0$ is then prepended to these embeddings, resulting in $[\boldsymbol{c}_0, \boldsymbol{x}_0]$, which is subsequently processed by a series of L transformer blocks $\boldsymbol{\mathcal{V}}_{i=1}^{L}$. The output class token $\boldsymbol{c}_L$ from the final block $\boldsymbol{\mathcal{V}}_L$ is projected via a linear

projection layer ($PatchProj$) to produce the final image representation $\boldsymbol{x}$, situated in the shared vision-language embedding space. Likewise, a text description $\boldsymbol{TD}$ is processed by the text encoder $\boldsymbol{T}$ to generate the corresponding textual feature $\boldsymbol{w}$, as follows:

$$[\boldsymbol{w}_0^j]_{j=1}^D = TextEmbed(\boldsymbol{TD}) \tag{4}$$

$$[\boldsymbol{w}_i^j]_{j=1}^D = \mathcal{T}_i([\boldsymbol{w}_{i-1}^j]_{j=1}^D) \quad i = 1, 2, \ldots, L \tag{5}$$

$$\boldsymbol{w} = TextProj(\boldsymbol{w}_L^D). \tag{6}$$

As illustrated, the text encoding also comprises three steps. First, the $TextEmbed$ module tokenizes the input text description $\boldsymbol{TD}$ and projects it into D word embeddings. Second, a sequence of transformer blocks $\boldsymbol{\mathcal{T}}_{i=1}^L$ is employed to extract high-level representations from the word embeddings. Third, the final token $\boldsymbol{w}_L^D$ from the last transformer block $\boldsymbol{\mathcal{T}}_L$ is projected by a linear layer ($TextProj$) into the shared vision-language embedding space to obtain the text feature $\boldsymbol{w}$. Given the image and text features, the cosine similarity score $sim(\boldsymbol{w}, \boldsymbol{x})$ is computed to perform task-specific predictions.

B. MMA: Multi-Modal Adapter

Our work primarily focuses on few-shot generalization task [7], where pre-trained CLIP models are first tuned on a set of classes with limited training samples, and then evaluated on unseen instances, e.g., novel classes or across different datasets without further adaptation. In such settings, it is widely acknowledged that effective instance representations must be both *discriminable*—to distinguish among classes—and *generalizable*—to transfer across diverse domains. These two properties are critical for successful transfer learning.

However, systematically quantifying these two characteristics remains challenging, particularly due to the limited number of available samples. Inspired by the concept of dataset bias introduced in [50], we introduce a task termed *dataset-level recognition* as a diagnostic analysis task, in which a model is trained to predict the dataset origin of a given sample. In this analysis, the same 11 public datasets from the base-to-novel and cross-dataset settings are used. Moreover, we employ only their training splits to ensure robustness and prevent overfitting to test data. Specifically, for each dataset, 16 training images and their corresponding class names are used, yielding $16 \times C$ image samples and C text samples (i.e., C is the number of classes). For the visual encoder analysis, the $16 \times C$ images are randomly and equally split into training and test subsets. This fully random splitting is performed without preserving sample balance per class. As a result, classes may or may not overlap between the training and test subsets. For the text encoder analysis, the C class name texts from each dataset are randomly and equally divided into training and test subsets, resulting in non-overlapping class names between these two subsets. Under this formulation, features with high discriminability facilitate the distinction between different datasets, whereas highly generalizable features exhibit invariance across datasets. To investigate these properties, we conduct an analysis using three pre-trained transformer-based CLIP models [2]: ViT-B/16, ViT-B/32, and ViT-L/14. These models are selected because of their competitive performance in vision-language tasks [6], [7], [25], [51]. All models share similar processes in their vision and text encoders, as described in (1) to (6). Furthermore, to gain a deeper understanding of the representations learned at various depths, we extract features from all layers of both the vision and text encoders. Linear classifiers are then trained on these features to perform dataset-level recognition. To improve the reliability of our conclusions, the entire process – including random data splitting, model training, and evaluation – is repeated independently for 3 runs, with results reported as the average over these runs. The analysis results are presented in Fig. 1, revealing two key observations and showing in the following.

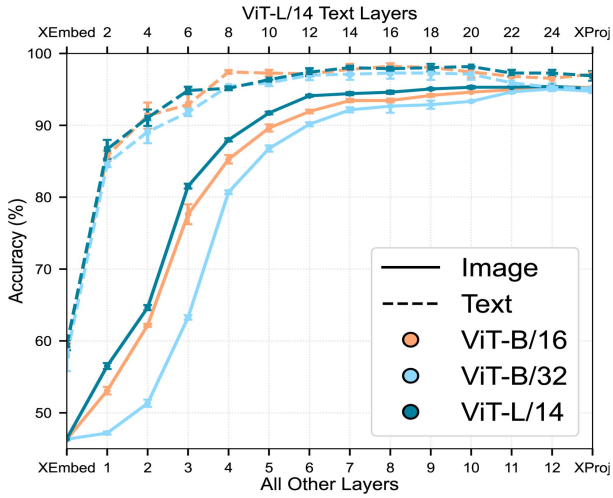

Fig. 1. Dataset-level recognition accuracy across transformer layers. The model identifies the source dataset of each sample, evaluated on training data from all datasets. Results show mean ± standard deviation over three random seeds. $XEmbed$ denotes pre-transformer embedding layers, while $XProj$ indicates final projection layers in text or image encoders.

Observation-1. In both the text and image encoders, the higher layers tend to capture more discriminable, dataset-specific representations, whereas the lower layers encode more generalizable features that are consistent across different datasets. These findings suggest that tuning higher layers is more effective for adapting to downstream tasks, while freezing the lower layers helps retain transferable, domain-invariant knowledge.

Macro Design: Motivated by **Observation-1**, we propose a novel Multi-Modal Adapter (MMA) framework, as illustrated in Fig. 2. Unlike most existing approaches that insert adapters or prompts throughout the entire network [21], [22], [23], [45] or into lower layers only [6], [7], [14], [16], the proposed adapter $\boldsymbol{\mathcal{A}}$ (described in the next) is selectively inserted into a few higher layers within both the visual and text encoders.

Formally, for the image encoder $\boldsymbol{\mathcal{V}}$, we incorporate the adapters $\boldsymbol{\mathcal{A}}^v$ starting from the k-th transformer block onward, and accordingly modify (2) as follows:

$$[\boldsymbol{c}_i, \boldsymbol{x}_i] = \boldsymbol{\mathcal{V}}_i([\boldsymbol{c}_{i-1}, \boldsymbol{x}_{i-1}]) \quad i = 1, 2, \ldots, k-1 \tag{7}$$

$$[\boldsymbol{c}_j, \boldsymbol{x}_j] = \boldsymbol{\mathcal{V}}_j([\boldsymbol{c}_{j-1}, \boldsymbol{x}_{j-1}]) + \alpha \underline{\boldsymbol{\mathcal{A}}_j^v([\boldsymbol{c}_{j-1}, \boldsymbol{x}_{j-1}])}$$

$$j = k, k+1, \ldots, L. \tag{8}$$

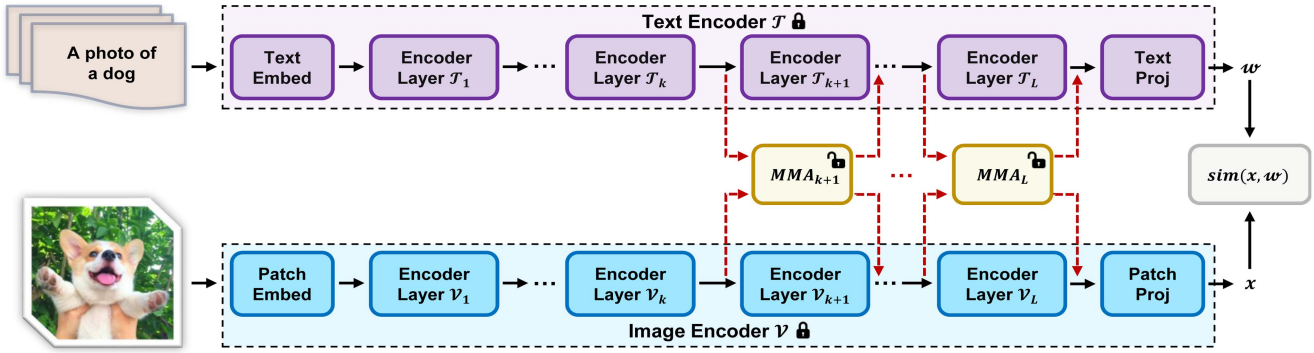

Fig. 2. The proposed Multi-Modal Adapter (MMA) framework for transformer-based CLIP models. MMA integrates lightweight, trainable adapters into both the image and text encoders while keeping the core CLIP model frozen. To address the trade-off between discriminative capacity and generalization, adapters are inserted only into the higher layers ($\geq k$) of each encoder, as guided by empirical analysis. Moreover, our MMA shares weights between image and text representations to learn shared cues from different branches. By this design, our MMA eliminates feature-wise interactions between each image-text pair [7], greatly reducing the computational cost.

Here, the underlined components indicate trainable blocks. The coefficient α serves to balance task-specific knowledge and the pre-trained general knowledge. Notably, when $\alpha = 0$, the model reduces to the original transformer block, without incorporating any task-specific adaptation. Similarly, we insert adapters $\mathcal{A}^t$ into the text encoder $\mathcal{T}$, and modify (5) accordingly as follows:

$$[\boldsymbol{w}_i^j]_{j=1}^D = \mathcal{T}_i([\boldsymbol{w}_{i-1}^j]_{j=1}^D) \quad i = 1, 2, \ldots, k-1 \tag{9}$$

$$[\boldsymbol{w}_i^j]_{j=1}^D = \mathcal{T}_j([\boldsymbol{w}_{i-1}^j]_{j=1}^D) + \alpha \underline{\mathcal{A}_j^t([\boldsymbol{w}_{i-1}^j]_{j=1}^D)}$$
$$j = k, k+1, \ldots, L \tag{10}$$

Observation-2. In most cases, text features, as they are encoded with semantic category names, are more discriminable across datasets than visual features. In addition, there are larger gaps between text and image features in lower layers than in higher layers. Therefore, we argue that it is more difficult to align lower layers between text and image features than between higher layers, especially tuning with limited training samples.

Micro Design: To encode task-specific knowledge, our approach inserts adapters independently into the image and text branches. However, as highlighted in **Observation-2**, the substantial semantic gap between visual and linguistic modalities poses a challenge for effective alignment, particularly under limited supervision. To mitigate this discrepancy, we propose a multi-modal unit featuring a shared projection layer (Fig. 3). Each modality is first mapped to a feature space via branch-specific projection layers. These features are then processed through a shared projection layer, followed by modality-specific transformations to restore the original branch dimensions. Formally, this process is defined as:

$$\mathcal{A}_k^v(\boldsymbol{z}_k^v) = \boldsymbol{W}_{ku}^v \cdot \delta(\boldsymbol{W}_{ks} \cdot \delta(\boldsymbol{W}_{kd}^v \cdot \boldsymbol{z}_k^v))$$
$$\boldsymbol{z}_k^v = [\boldsymbol{c}_k, \boldsymbol{x}_k] \tag{11}$$

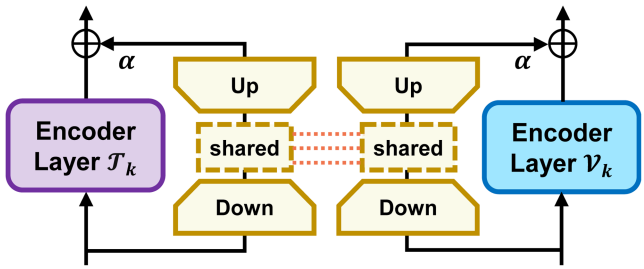

Fig. 3. The proposed multi-modal unit comprises modality-specific projection layers ("Down" and "Up") that adapt the respective encoders. It also has a shared projection layer ("Shared") that facilitates tight coupling between the visual and linguistic representations.

A similar process is added to text encoder as follows:

$$\mathcal{A}_k^t(\boldsymbol{z}_k^t) = \boldsymbol{W}_{ku}^t \cdot \delta(\boldsymbol{W}_{ks} \cdot \delta(\boldsymbol{W}_{kd}^t \cdot \boldsymbol{z}_k^t))$$
$$\boldsymbol{z}_k^t = [\boldsymbol{w}_k^j]_{j=1}^D \tag{12}$$

Here, $\boldsymbol{W}_{ku}$ and $\boldsymbol{W}_{kd}$ denote the k-th "Up" and "Down" projection layers, respectively. δ is the non-linear activation function. As shown in Fig. 3, with modality branches indicated by superscripts. $\boldsymbol{W}_{ks}$ represents the k-th shared projection layer used in both (11) and (12). Critically, this shared component serves as a bridge between modalities, enabling gradient flow across branches and thereby promoting more effective cross-modal alignment.

C. Mma++

While our MMA introduces specific modulation, its effectiveness hinges not only on the adapter architecture but also on how its output is integrated with the pre-trained backbone. In particular, we adopt a residual fusion strategy as shown in (8) and (10). As presented, α is a handcrafted scaling factor that directly controls the degree to which the adapter influences the final representation. Understanding the role of this scalar is crucial for balancing stability and adaptability.

Although several methods involve fusion scales—such as setting fixed weights between pre-trained and adapted features [23],

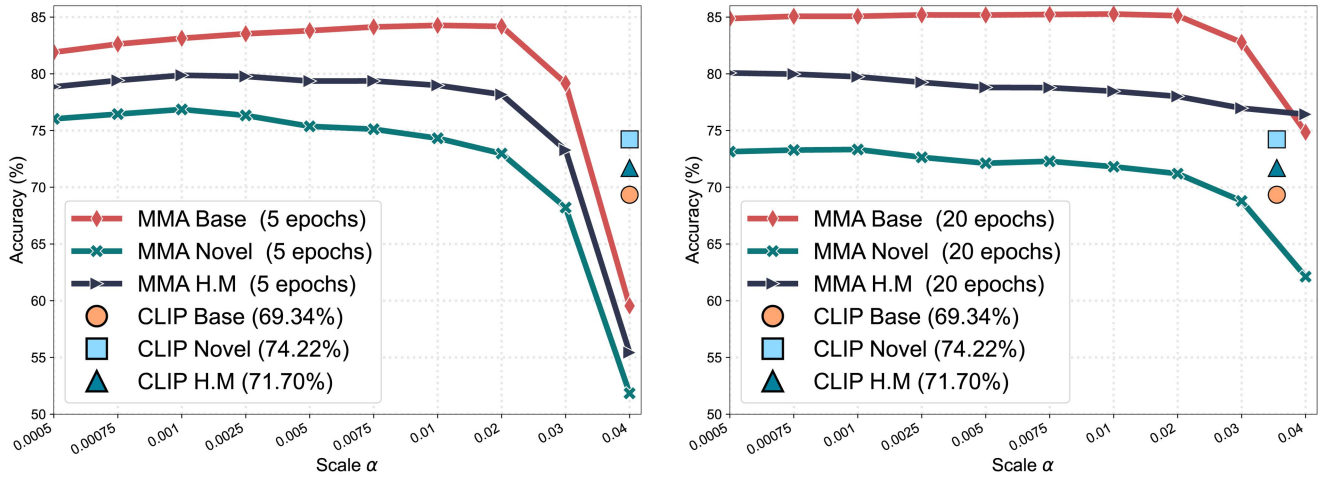

Fig. 4. Accuracy comparison (%) under different implementations. MMA results are reported for 5- (**left**) and 20-epoch (**right**) training regimes, while CLIP serves as the baseline (69.34% Base, 74.22% Novel, 71.70% H.M). The scale parameter α controls adaptation strength in MMA. These results are obtained on the widely used 11 image classification tasks.

[25], [29], [30]—these methods do not offer a systematic investigation of its effects. In particular, we argue that the fusion scale α plays a central role in adapter-based tuning, yet its impact remains largely unexplored.

A Preliminary Experiment to Understand α: To investigate the impact of the fusion scale α, we conduct a systematic study by varying α and evaluating model performance on 11 widely used image classification datasets under the Base-to-Novel Generalization setting. Because the 5-epoch setting used in our earlier implementation (MMA [29]) may not fully reflect the training dynamics, we extend the training schedule to 20 epochs. This longer schedule allows us to better examine the influence of α under more stable training conditions.

As shown in Fig. 4, increasing α from 0.0005 to 0.01 leads to consistent improvements in base class accuracy, particularly under the shorter training regime. This indicates that stronger adapter signals can enhance specialization on the trained categories. However, this gain comes at the expense of generalization: performance on novel classes begins to deteriorate as α increases beyond 0.001, suggesting that excessive reliance on adapter outputs weakens the model's ability to generalize to underrepresented or unseen categories. Furthermore, when α becomes excessively large (e.g., $\alpha \geq 0.02$), we observe a pronounced decline in both base and novel performance. This indicates that the model begins to overfit to the limited training data, overemphasizing the adapter outputs while disregarding the robust representations encoded in the pre-trained backbone. Consequently, the learned features become less transferable, ultimately impairing both memorization and generalization.

Theoretical Analysis of the Scale α: To rigorously understand the relationship among the fusion scale α, the amount of training data N, and the generalization ability, we consider the generalization behavior of the proposed blockwise fusion scheme, where each Transformer block integrates a frozen pre-trained feature A_l with a trainable adapter output B_l via a fusion coefficient α_l. Assuming the block functions are differentiable with bounded Jacobians $\|J_k\| \leq 1$, we can approximate the final model output as:

$$f_\alpha(x) \approx A(x) + \sum_{l=1}^{L} \alpha_l B_l(x), \tag{13}$$

where $A(x)$ is the fixed pre-trained backbone and $\{B_l\}$ are adapter modules. To assess generalization, we analyze the empirical Rademacher complexity of this model class [52], [53], which can yield the bound:

$$\mathcal{R}_N(\mathcal{H}_{\boldsymbol{\alpha}}) \leq \frac{CM}{\sqrt{N}} \sum_{l=1}^{L} \alpha_l, \tag{14}$$

where C bounds the adapter parameter norm ($\|\theta_l\|_F \leq C$) and M bounds the adapter outputs ($\|B_l(x)\| \leq M$). Futher, we assume the loss function $\ell(f(x), y)$ be 1-Lipschitz and bounded in $[0, 1]$. This can be easily achieved with normalization to bound output logits. After that, using standard generalization bounds for Lipschitz loss functions, we have that with probability at least $1 - \delta$:

$$\underbrace{\mathbb{E}[\ell(f(x), y)]}_{\text{Expected risk}} \leq \underbrace{\frac{1}{N} \sum_{i=1}^{N} \ell(f(x_i), y_i)}_{\text{Empirical risk}} \tag{15}$$

$$+ \underbrace{\frac{2CM}{\sqrt{N}} \sum_{l=1}^{L} \alpha_l + \sqrt{\frac{\log(1/\delta)}{2N}}}_{\text{Error gap}}. \tag{16}$$

To ensure the generalization gap does not exceed ε, we must require the following:

$$\sum_{l=1}^{L} \alpha_l \leq \frac{\sqrt{N}}{2CM} \left(\varepsilon - \sqrt{\frac{\log(1/\delta)}{2N}} \right). \tag{17}$$

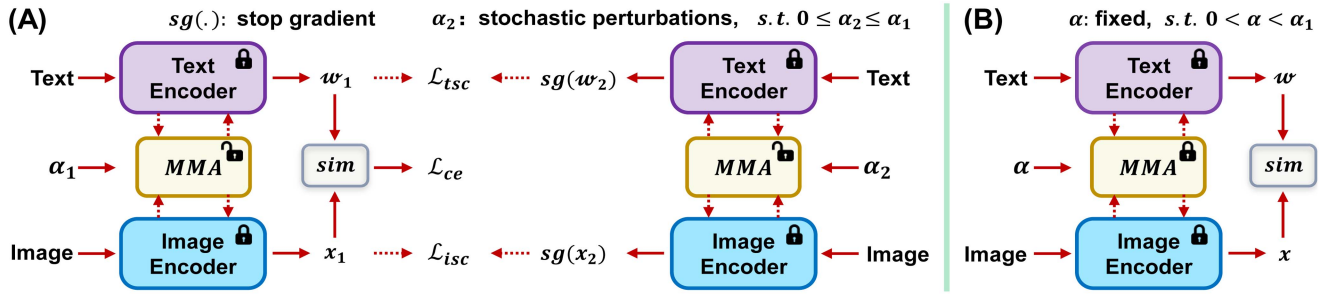

Fig. 5. The proposed α**-consistency** framework for our MMA. (a) shows the α-consistency training pipeline, which employs stop-gradient ($sg(\cdot)$) and stochastic perturbations ($0 \leq \alpha_2 \leq \alpha_1$) to enforce feature alignment. Here, $\mathcal{L}_{isc}$ and $\mathcal{L}_{tsc}$ ensure consistency between text ($\boldsymbol{w}$) and image ($\boldsymbol{x}$) features across varying α scales, while $\mathcal{L}_{ce}$ denotes the standard cross-entropy loss. (b) simplifies inference by using a fixed α decoupled from training. This enables the test-time scale to differ from those used during training, thus enhancing overall generalization (e.g.H.M).

Assuming uniform scaling $\alpha_l = \alpha$, this implies:

$$\alpha \leq \frac{1}{2CML}\left(\sqrt{N}\varepsilon - \sqrt{\frac{\log(1/\delta)}{2}}\right) \Rightarrow \alpha \propto \frac{\sqrt{N}}{L} \tag{18}$$

Theoretical analysis indicates that the fusion scale α increases with the training set size N but decreases inversely with the model depth L. This holds even when the Jacobian norm exceeds one ($\|J_k\| > 1$). In practice, where L is typically fixed due to the use of a frozen pre-trained Transformer, choosing a smaller α is essential for maintaining good generalization under few-shot or low-resource settings. *We kindly refer readers to Appendix for the full derivation*.

α-Consistency Framework for Our MMA: Our comprehensive analysis, combining empirical evidence from Fig. 4 with theoretical insights, reveals a critical trade-off in the fusion mechanism governed by scale parameter α. Increasing α enhances memorization capacity through amplified adapter contributions, but may at the expense of generalization performance, particularly on unseen instances. This trade-off is exacerbated under few-shot conditions, where over-amplified adapter outputs may overshadow the general representations pre-learned by the frozen backbone, leading to suboptimal hypothesis transfer.

To address this challenge, we conceptualize both the training and testing stages of our MMA framework as risk-driven processes operating under different data regimes but sharing a unified objective: minimizing task-specific risk through the appropriate choice of model behavior. During training, empirical risk minimization is carried out over a labeled dataset of size N, and our theoretical derivation shows that training may benefit from a relatively larger α. At test time, even though model parameters are frozen and no gradient-based optimization is performed, the inference process can still be framed as an *implicit form of adaptation*. Specifically, we interpret model inference as a test-time training procedure. In this procedure, the adapter parameters are fixed, and the fusion scale α should be tuned. However, the effective number of samples for this test-time training is one. Under this unified view, both training and testing can be interpreted as solving a risk minimization problem with fusion scale α as a controllable inductive bias, adjusted in proportion to the effective number of samples available in each stage. This reinterpretation naturally explains the need to reduce α during evaluation and offers a principled basis for different scales.

However, setting an optimal α at test time is inherently challenging, as no labeled samples or optimization feedback are available. To enable robust behavior under such constraints, we propose the **α-consistency framework**, which explicitly aligns the behaviors of the model under different fusion scales during training, thus preparing it for test-time inference. The overall procedure is illustrated in Fig. 5.

As illustrated in Fig. 5(a), the training phase incorporates two forward passes that share the same encoders and MMA units, but operate under different fusion scales. The first pass adopts a relatively large scale α_1 in (8) and (10), which facilitates better fitting to the training data distribution. The second pass employs a stochastically perturbed scale α_2 in (8) and (10), constrained such that $0 \leq \alpha_2 \leq \alpha_1$. Let $\boldsymbol{w}_1$ ($\boldsymbol{x}_1$) and $\boldsymbol{w}_2$ ($\boldsymbol{x}_2$) denote the text (image) features obtained under fusion scales α_1 and α_2, respectively. After that, a standard cross-entropy loss $\mathcal{L}_{ce}$ is applied to the similarity score $sim(\boldsymbol{w}_1, \boldsymbol{x}_1)$ for image classification. Besides, to enhance feature robustness across fusion conditions, we introduce two auxiliary consistency losses: the text consistency loss $\mathcal{L}_{tsc} = -sim(\boldsymbol{w}_1, \boldsymbol{w}_2)$ and the image consistency loss $\mathcal{L}_{isc} = -sim(\boldsymbol{x}_1, \boldsymbol{x}_2)$, where sim is the cosine similarity. Both losses operate at the feature level, encouraging alignment between representations extracted under α_1 and α_2. The total objective is defined as:

$$\mathcal{L}_{overall} = \mathcal{L}_{ce} + \lambda\left(\mathcal{L}_{tsc} + \mathcal{L}_{isc}\right), \tag{19}$$

where λ controls the strength of the consistency regularization.

As demonstrated in [54], consistency learning is prone to representational collapse, which can be mitigated by incorporating a stop-gradient operation. Therefore, in our framework, we apply the stop-gradient to the α_2-generated features, i.e., $\boldsymbol{w_2}$ and $\boldsymbol{x}_2$, to prevent the collapse of representations during optimization. Moreover, as presented in Fig. 5(a), we introduce stochasticity in α_2 for the following reasons. First, it simulates the inherent uncertainty of test-time conditions, where the optimal fusion scale is unknown and label supervision is unavailable. Second, it encourages the model to generalize over a distribution of plausible fusion behaviors, rather than overfitting to a fixed configuration. By sampling α_2 during training, the model is exposed to diverse fusion conditions.

Although the above proposed α-consistency training enhances representation robustness to varying fusion scales, a discrepancy remains between training and testing conditions–specifically in the number of available samples as our previous insights. To mitigate this gap, we introduce a train-test scale decoupling strategy, where a smaller fusion scale $0 < \alpha < \alpha_1$ is employed during testing. As illustrated in Fig. 5(b), testing is performed deterministically using the lower scale α, without relying on stochastic perturbations or additional adaptation. By decoupling the training and testing fusion scales, we enforce conservative usage of the adapter branch, which reduces variance and improves generalization to unseen inputs. This setting ensures that the inductive bias introduced by the adapter remains calibrated to the test-time data regime, and is motivated by the empirical observation that lower α improves generalization, which aligns with the trend suggested by our theoretical analysis.

IV. Experiments

We evaluate the performance of MMA and MMA++ based on previous works [7], [14], including *Generalization from Base-to-Novel Classes*, *Cross-Dataset Evaluation*, and *Domain Generalization*. All these experiments are based on 16-shot settings, i.e., only 16 training examples per category.

Generalization from Base-to-Novel Classes: Following standard practice in prior works [6], [7], we evaluate our method on 11 image classification datasets. These include two general object recognition datasets: ImageNet [55] and Caltech101 [56]; five fine-grained recognition datasets: OxfordPets [57], StanfordCars [58], Flowers102 [59], Food101 [60], and FGVCAircraft [61]; one scene understanding dataset: SUN397 [62]; one texture dataset: DTD [63]; one satellite image classification dataset: EuroSAT [64]; and one action recognition dataset: UCF101 [65]. These benchmarks span a diverse set of visual recognition tasks, enabling a comprehensive assessment of a model's generalization ability. We follow the common evaluation protocol used in [7], [8], [14], [44], where the model is trained using only base classes under a 16-shot setting without using information from novel categories, and evaluated on both base and novel classes.

Cross-dataset Evaluation: Consistent with the Base-to-Novel experiments, we conduct cross-dataset evaluation using the same 11 datasets. Following the protocol in CoCoOp [7], all models are trained on ImageNet using 16 training samples per class across its 1,000 categories. The trained models are then directly evaluated on the remaining datasets without any further dataset-specific tuning.

Domain Generalization: To assess model robustness under domain shift, Zhou et al. [7] propose evaluating ImageNet tuned models on four out-of-distribution variants of ImageNet: ImageNetV2 [66], ImageNet-Sketch [67], ImageNet-A [68], and ImageNet-R [69]. Following this protocol, we also conduct evaluations on these datasets to measure the out-of-distribution generalization performance.

Implementation Details: Following prior works [6], [7], [14], [29], all experiments are conducted under the few-shot setting, specifically with 16 shots per category. We adopt the ViT-B/16 variant of CLIP as the backbone model throughout. Recently, several approaches [17], [18], [30], [35] have explored LLMs to enrich class-wise textual descriptions, aiming to improve robustness. In line with these studies, we incorporate LLM-generated descriptions [38] into our framework. The design of our multi-modal units and their insertion points in MMA++ follow the same configuration as in the Base-to-Novel setting from our earlier version,[1] and remain fixed across all settings. In this study, we use a batch size of 16 for all datasets. In the Base-to-Novel setting, models are trained for 50 epochs. The learning rate is set to 0.3 for handcrafted prompts and 0.2 for LLM-augmented ones. The loss weight λ in (19) and the test-time fusion scale α are fixed at 2.5 and 0.0065, respectively, for both prompt types. For the other two experimental settings, following prior studies [14], [29], [30], we train for a shorter epoch (i.e., 15) and enable train-test decoupling by using a lower test-time α (i.e., 0.005). All other configurations remain consistent with the Base-to-Novel setting. Model optimization is performed using SGD with a momentum of 0.9 and weight decay of 0.0005. All models are trained with mixed precision on a single GPU, using a cosine learning rate schedule, and the reported results are averaged over multiple runs with different random seeds. We report classification accuracies for both **Base** and **Novel** classes, along with their harmonic mean (**H.M**). For the remaining two settings, we report average per-class accuracy on each dataset. We also report average rank (Avg.Rank), which denotes the average ranking of each method across all datasets. It is computed by ranking methods according to the target metric (e.g., H.M. or accuracy) on each dataset and then averaging the ranks, where lower values indicate better overall performance.

A. Main Results

1) Base-to-Novel Generalization Without LLM: We first evaluate the generalization performance of our MMA and MMA++ under the standard Base-to-Novel setting without utilizing LLM-based text augmentation, as summarized in Table I. We compare our approach against a broad spectrum of state-of-the-art methods, including the zero-shot baseline CLIP [2]; prompt-based learners such as CoOp [6], CoOpOp [7], ProDA [44], KgCoOp [8], LASP-V [9], and PromptSRC [16]; multi-modal prompt learning methods such as MaPLe [14] and RPO [15]; and more recent strategies including DePT [19], DPC [47] and SkipT [42]. For a fair comparison, we exclude PromptKD [20], which adopts transductive learning by leveraging the entire unlabeled test set during training, and MMRL [46], which relies on dataset-specific models or hyperparameter tuning, making it less suitable for evaluating generalizable approaches. From the results in Table I(b), we draw three key observations:

First, MMA++ achieves the second best average H.M of 80.81 across all 11 datasets, outperforming both our previous method MMA (79.87) and some other strong competitors such as DPC (80.04) and DePT (80.43). This result highlights

[1]Implementation details of MMA are provided in our previous conference paper, we kindly refer readers there for more information.

TABLE I

BASE-TO-NOVEL GENERALIZATION WITHOUT LLM FOR TEXT AUGMENTATION. "BASE" AND "NOVEL" INDICATE ACCURACIES ON BASE AND NOVEL CLASSES, RESPECTIVELY; "H.M" IS THEIR HARMONIC MEAN. ALL RESULTS ARE CITED FROM ORIGINAL PAPERS. MMA++ ACHIEVES STATE-OF-THE-ART H.M BY EFFECTIVELY BALANCING ADAPTATION AND GENERALIZATION. GREY ENTRIES USE DATASET-SPECIFIC ARCHITECTURES OR HYPERPARAMETERS (E.G., VARYING FEATURE DIMENSIONS FOR EUROSAT, CUSTOM LOSS WEIGHTS ON EACH DATASET), AND THUS DEVIATE FROM MOST OF EXISTING METHODS AND ARE NOT DIRECTLY COMPARABLE. **RED** AND **BLUE** ARE THE BEST AND SECOND BEST ACCORDING TO THE H.M.

(a) Average rank (Avg.Rank) of the H.M of all compared methods on each dataset.

Method	CLIP	CoOp	CoOpOp	ProDA	KgCoOp	MaPLe	LASP-V	RPO	PromptSRC	DePT	DPC	SkipT	MMA	MMA++
Avg.Rank	13.09	13.55	11.00	10.64	9.82	7.64	5.55	8.45	5.82	**4.00**	4.18	**2.82**	5.27	**2.82**

(b) Detail comparisons on all 11 datasets of state-of-the-art methods.

Method	Avg			ImageNet			Caltech101			Pets		
	Base	Novel	H.M	Base	Novel	H.M	Base	Novel	H.M	Base	Novel	H.M
CLIP [ICML2021] [2]	69.34	74.22	71.70	72.43	68.14	70.22	96.84	94.00	95.40	91.17	97.26	94.12
CoOp [IJCV2022] [6]	82.69	63.22	71.66	76.47	67.88	71.92	98.00	89.81	93.73	93.67	95.29	94.47
CoOpOp [CVPR2022] [7]	80.47	71.69	75.83	75.98	70.43	73.10	97.96	93.81	95.84	95.20	97.69	96.43
ProDA [CVPR2022] [44]	81.56	72.30	76.65	75.40	70.23	72.72	98.27	93.23	95.68	95.43	97.83	96.62
KgCoOp [CVPR2023] [8]	80.73	73.60	77.00	75.83	69.96	72.78	97.72	94.39	96.03	94.65	97.76	96.18
MaPLe [CVPR2023] [14]	82.28	75.14	78.55	76.66	70.54	73.47	97.74	94.36	96.02	95.43	97.76	96.58
LASP-V [CVPR2023] [9]	83.18	76.11	79.48	76.25	71.17	73.62	98.17	94.33	96.21	95.73	97.87	**96.79**
RPO [ICCV2023] [15]	81.13	75.00	77.78	76.60	71.57	74.00	97.97	94.37	96.03	94.63	97.50	96.05
PromptSRC [ICCV2023] [16]	84.26	76.10	79.97	77.60	70.73	74.01	98.10	94.03	96.02	95.33	97.30	96.30
DePT [CVPR2024] [19]	85.19	76.17	80.43	78.20	70.27	74.02	98.57	94.10	96.28	95.43	97.33	96.37
DPC [CVPR2025] [47]	86.10	74.78	80.04	78.48	70.72	**74.40**	98.90	94.21	96.50	96.13	97.30	96.71
SkipT [CVPR2025] [42]	85.04	77.53	**81.11**	77.73	70.40	73.89	98.50	95.33	**96.89**	95.70	97.87	**96.77**
MMRL [CVPR2025] [46]	85.68	77.16	81.20	77.90	71.30	74.45	98.97	94.50	96.68	95.90	97.60	96.74
MMA [CVPR2024] [29]	83.20	76.80	79.87	77.31	71.00	74.02	98.40	94.00	96.15	95.40	98.07	96.72
MMA++ [This Work]	85.24	76.80	**80.81**	78.33	71.20	**74.56**	98.70	94.53	**96.57**	95.83	97.40	96.61

Methods	Cars			Flowers			Food101			Aircraft		
	Base	Novel	H.M	Base	Novel	H.M	Base	Novel	H.M	Base	Novel	H.M
CLIP [ICML2021] [2]	63.37	74.89	68.65	72.08	77.80	74.83	90.10	91.22	90.66	27.19	36.29	31.09
CoOp [IJCV2022] [6]	78.12	60.40	68.13	97.60	59.67	74.06	88.33	82.26	85.19	40.44	22.30	28.75
CoOpOp [CVPR2022] [7]	70.49	73.59	72.01	94.87	71.75	81.71	90.70	91.29	90.99	33.41	23.71	27.74
ProDA [CVPR2022] [44]	74.70	71.20	72.91	97.70	68.68	80.66	90.30	88.57	89.43	36.90	34.13	35.46
KgCoOp [CVPR2023] [8]	71.76	75.04	73.36	95.00	74.73	83.65	90.50	91.70	91.09	36.21	33.55	34.83
MaPLe [CVPR2023] [14]	72.94	74.00	73.47	95.92	72.46	82.56	90.71	92.05	91.38	37.44	35.61	36.50
LASP-V [CVPR2023] [9]	75.23	71.77	73.46	97.17	73.53	83.71	91.20	91.90	**91.54**	38.05	33.20	35.46
RPO [ICCV2023] [15]	73.87	75.53	74.69	94.13	76.67	84.50	90.33	90.83	90.58	37.33	34.20	35.70
PromptSRC [ICCV2023] [16]	78.27	74.97	76.58	98.07	76.50	**85.95**	90.67	91.53	91.10	42.73	37.87	40.15
DePT [CVPR2024] [19]	80.80	75.00	77.79	98.40	77.10	**86.46**	90.87	91.57	91.22	45.70	36.73	40.73
DPC [CVPR2025] [47]	82.28	74.98	**78.46**	97.44	73.19	83.59	91.40	91.58	**91.49**	46.74	35.33	40.24
SkipT [CVPR2025] [42]	82.93	72.50	77.37	98.57	75.80	85.70	90.67	92.03	91.34	45.37	37.13	**40.84**
MMRL [CVPR2025] [46]	81.30	75.07	78.06	98.97	77.27	86.78	90.57	91.50	91.03	46.30	37.03	41.15
MMA [CVPR2024] [29]	78.50	73.10	75.70	97.77	75.93	85.48	90.13	91.30	90.71	40.57	36.33	38.33
MMA++ [This Work]	82.63	74.83	**78.54**	95.07	77.17	85.19	90.83	91.63	91.23	48.64	38.05	**42.70**

Methods	SUN397			DTD			EuroSAT			UCF101		
	Base	Novel	H.M	Base	Novel	H.M	Base	Novel	H.M	Base	Novel	H.M
CLIP [ICML2021] [2]	69.36	75.35	72.23	53.24	59.90	56.37	56.48	64.05	60.03	70.53	77.50	73.85
CoOp [IJCV2022] [6]	80.60	65.89	72.51	79.44	41.18	54.24	92.19	54.74	68.69	84.69	56.05	67.46
CoOpOp [CVPR2022] [7]	79.74	76.86	78.27	77.01	56.00	64.85	87.49	60.04	71.21	82.33	73.45	77.64
ProDA [CVPR2022] [44]	78.67	76.93	77.79	80.67	56.48	66.44	83.90	66.00	73.88	85.23	71.97	78.04
KgCoOp [CVPR2023] [8]	80.29	76.53	78.36	77.55	54.99	64.35	85.64	64.34	73.48	82.89	76.67	79.65
MaPLe [CVPR2023] [14]	80.82	78.70	79.75	80.36	59.18	68.16	94.07	73.23	82.35	83.00	78.66	80.77
LASP-V [CVPR2023] [9]	80.70	79.30	80.00	81.10	62.57	70.64	95.00	83.37	**88.86**	85.53	78.20	81.70
RPO [ICCV2023] [15]	80.60	77.80	79.18	76.70	62.13	68.61	86.63	68.97	76.79	83.67	75.43	79.34
PromptSRC [ICCV2023] [16]	82.67	78.47	80.52	83.37	62.97	71.75	92.90	73.90	82.32	87.10	78.80	82.74
DePT [CVPR2024] [19]	83.27	78.97	**81.06**	84.80	61.20	71.09	93.23	77.90	84.88	87.73	77.70	82.46
DPC [CVPR2025] [47]	83.63	78.08	80.76	86.88	54.31	66.84	96.25	74.73	84.13	88.99	78.13	83.21
SkipT [CVPR2025] [42]	82.40	79.03	80.68	83.77	67.23	**74.59**	92.47	83.00	**87.48**	87.30	82.47	**84.81**
MMRL [CVPR2025] [46]	83.20	79.30	81.20	85.67	65.00	73.82	95.60	80.17	87.21	88.10	80.07	83.89
MMA [CVPR2024] [29]	82.27	78.57	80.38	83.20	65.63	**73.38**	85.46	82.34	83.87	86.23	80.03	82.20
MMA++ [This Work]	82.80	79.57	**81.15**	82.30	64.30	72.19	94.55	75.97	84.25	88.30	80.17	**84.04**

TABLE II
BASE-TO-NOVEL GENERALIZATION WITH LLM FOR TEXT AUGMENTATION. METRICS ARE THE SAME TO TABLE I. **RED** AND **BLUE** ARE THE BEST AND SECOND BEST ACCORDING TO THE H.M.

(a) Average rank (Avg.Rank) of the H.M of all compared methods on each dataset.

Method	CoPrompt	HPT	LLaMP	LwEIB	MMA++
Avg.Rank	3.64	4.00	**2.09**	3.36	**1.91**

(b) Detail comparisons on all 11 datasets of state-of-the-art methods.

Method	Avg			ImageNet			Caltech101			Pets		
	Base	Novel	H.M Base	Novel	H.M	Base	Novel	H.M	Base	Novel	H.M	
CoPrompt [ICLR2024] [18]	84.00	77.23	80.48	77.67	71.27	74.33	98.27	94.90	96.55	95.67	98.10	96.87
HPT [AAAI2024] [17]	84.32	76.86	80.23	77.95	70.74	74.17	98.37	94.98	96.65	95.78	97.65	96.71
LLaMP [CVPR2024] [35]	85.16	77.71	**81.27**	77.99	71.27	**74.48**	98.45	95.85	**97.17**	96.31	97.74	**97.02**
LwEIB [IJCV2025] [30]	84.45	78.21	81.21	76.64	71.64	74.06	98.47	95.47	96.95	95.70	97.40	96.54
MMA++ [This Work]	85.78	78.67	**82.07**	78.37	72.03	**75.07**	98.60	95.37	**96.97**	95.80	97.33	**96.56**

Methods	Cars			Flowers			Food101			Aircraft		
	Base	Novel	H.M	Base	Novel	H.M	Base	Novel	H.M	Base	Novel	H.M
CoPrompt [ICLR2024] [18]	76.97	74.40	75.66	97.27	76.60	85.71	90.73	92.07	91.40	40.20	39.33	39.76
HPT [AAAI2024] [17]	76.95	74.23	75.57	98.17	78.37	**87.16**	90.46	91.57	91.01	42.68	38.13	40.28
LLaMP [CVPR2024] [35]	81.56	74.54	**77.89**	97.82	77.40	86.42	91.05	91.93	**91.47**	47.30	37.61	41.90
LwEIB [IJCV2025] [30]	80.07	74.01	76.92	97.53	77.50	86.37	90.63	91.73	91.18	45.11	42.60	**43.82**
MMA++ [This Work]	82.77	73.87	**78.07**	98.10	78.13	**86.98**	90.83	91.93	**91.38**	49.30	43.33	**46.12**

Methods	SUN397			DTD			EuroSAT			UCF101		
	Base	Novel	H.M	Base	Novel	H.M	Base	Novel	H.M	Base	Novel	H.M
CoPrompt [ICLR2024] [18]	82.63	80.03	81.31	83.13	64.73	72.79	94.60	78.57	85.84	86.90	79.57	83.07
HPT [AAAI2024] [17]	82.57	79.26	80.88	83.84	63.33	72.16	94.24	77.12	84.82	86.52	80.06	83.16
LLaMP [CVPR2024] [35]	83.41	79.90	**81.62**	83.49	64.49	72.77	91.93	83.66	**87.60**	87.13	80.66	83.77
LwEIB [IJCV2025] [30]	81.10	79.80	80.44	82.87	67.83	**74.60**	95.00	80.01	**86.86**	85.73	82.37	**84.02**
MMA++ [This Work]	83.23	80.57	**81.88**	84.43	70.43	**76.80**	93.70	78.83	85.62	88.40	83.60	**85.93**

MMA++'s good ability to balance adaptation to base classes and generalization to novel classes.

Second, the average rank comparison in Table I(a) further confirms the robustness of MMA++. Our MMA++ achieve the same overall rank (2.82) with the latest published SkipT, and outperforms DPC (4.18), DePT (4.00), and LASP-V (5.55), underscoring its stability and effectiveness across heterogeneous datasets. Moreover, both DPC and DePT achieve such results with separate models or distinct parameters for base and novel classes, which necessitates prior knowledge of the class type during testing. In contrast, our MMA++ introduces a train-test decoupling paradigm that does not require such strong assumptions about the test data.

Third, while no single method dominates all benchmarks, MMA++ consistently ranks among the top performers. Specifically, it achieves the highest H.M on 4 out of 11 datasets (ImageNet, Cars, Aircraft, and SUN397) and secures the second-best score on another two datasets. These results underscore the inherent challenges of the Base-to-Novel generalization task, highlighting that achieving robust performance across diverse datasets and classes remains a demanding problem.

2) Base-to-Novel Generalization With LLM: Some recent methods [17], [18], [30], [35] leverage LLMs for texture augmentation. For a more comprehensive comparison, we incorporate GPT-3 generated texts [38] into our MMA++ and compare ours with other related methods in Table II.

With LLM-enhanced prompts, MMA++ further improves the performance and achieves the best H.M (82.07) across all methods. In terms of average rank (Table II(a)), MMA++ achieves the lowest (best) average ranking of 1.91, outperforming the second-best LLaMP (2.09). Moreover, MMA++ attains the best or second-best H.M on 10 out of 11 datasets, demonstrating both strong base accuracy and superior generalization to novel categories. For example, it achieves state-of-the-art results on challenging datasets such as SUN397 (81.88), DTD (76.80), Aircraft (46.12), and UCF101 (85.93). Even on datasets where the gap among methods is narrow (e.g., Caltech101, Flowers102, Food101), MMA++ still performs competitively to other methods.

Building on the above results, and to ensure a contemporary comparison with many recent methods, we adopt the full version of our approach—MMA++ augmented with LLM-generated descriptions—for all evaluations in the Cross-Dataset and Domain Generalization settings.

3) Cross-Dataset Evaluation: In this study, we also conduct experiments under the Cross-Dataset setting. Specifically, all models are trained on the full 1000-class ImageNet dataset with 16 shots per category, and directly evaluated on ten target datasets used in the previous Base-to-Novel setting. For a fair comparison, the methods using dataset-specific tuning [46] or training with testing data [20] are not included.

The results are summarized in Table III. Compared with existing methods, our MMA++ achieves the highest average accuracy (69.66) and the second best average rank (3.00), clearly outperforming both prompt tuning baselines (e.g., CoOp [6] and CoCoOp [7]) and more recent LLM-augmented approaches

TABLE III
CROSS-DATASET EVALUATION ACROSS TEN TARGET DATASETS. ALL MODELS ARE TRAINED ON IMAGENET-1 K WITH 16 SHOTS PER CLASS AND DIRECTLY EVALUATED ON TEN DATASETS. OUR MMA++—AN ENHANCED VERSION OF MMA [29]—ACHIEVES THE HIGHEST AVERAGE ACCURACY (69.78) AND BEST AVERAGE RANK (1.80), DEMONSTRATING STRONG GENERALIZATION. METHODS LIKE MMRL [46] AND PROMPTKD [20] ARE EXCLUDED DUE TO DATASET-SPECIFIC TUNING OR TRANSDUCTIVE SETTINGS, MAKING THEM INCOMPATIBLE WITH THE OTHER METHODS.

Method	ImageNet	Caltech101	Pets	Cars	Flowers	Food101	Aircraft	SUN397	DTD	EuroSAT	UCF101	Avg	Avg.Rank
CoOp [IJCV2022] [6]	71.51	93.70	89.14	64.51	68.71	85.30	18.47	64.15	41.92	46.39	66.55	63.88	10.40
CoCoOp [CVPR2022] [7]	71.02	94.43	90.14	65.32	71.88	86.06	22.94	67.36	45.73	45.37	68.21	65.74	8.50
MaPLe [CVPR2023] [14]	70.72	93.53	90.49	65.57	72.23	86.20	24.74	67.01	46.49	48.06	68.69	66.30	7.00
PromptSRC [ICCV2023] [16]	71.27	93.60	90.25	65.70	70.25	86.15	23.90	67.10	46.87	45.50	68.75	65.81	7.80
CoPrompt [ICLR2024] [18]	70.80	**94.50**	90.73	65.67	72.30	**86.43**	24.00	67.57	47.07	51.90	69.73	67.00	4.30
DePT [CVPR2024] [19]	71.60	93.80	90.13	66.00	70.93	86.27	24.30	67.23	46.60	45.83	69.10	66.02	6.70
HPT [AAAI2024] [17]	71.72	94.20	**92.63**	**66.33**	**74.84**	86.21	25.68	68.75	**50.87**	47.36	**70.50**	67.74	3.10
SkipT [CVPR2025] [42]	**72.77**	93.43	90.10	65.37	71.97	86.17	25.13	67.33	48.00	54.27	68.23	67.00	7.10
LwEIB [IJCV2025] [30]	71.31	**94.51**	92.50	**66.58**	73.03	**86.37**	**27.70**	**69.33**	50.63	**55.37**	70.03	**68.61**	**2.20**
MMA [CVPR2024] [29]	71.00	93.80	90.30	66.13	72.07	86.12	25.33	68.17	46.57	49.24	68.32	66.61	5.30
MMA++ [This Work]	**73.34**	94.30	**92.70**	65.47	**73.83**	85.87	**29.10**	**70.13**	**54.13**	**60.20**	**70.87**	**69.66**	**3.00**

(e.g., CoPrompt [18] and HPT [17]). Notably, MMA++ ranks first on 6 out of the 10 transfer datasets, including challenging domains such as Aircraft (29.10), SUN397 (70.13), DTD (54.13), EuroSAT (60.20), and UCF101 (70.87).

Compared to our earlier version MMA [29], MMA++ achieves a +3.05 absolute improvement in average accuracy (from 66.61 to 69.66) and advancing its average rank from 5.30 to 3.00. It is also worth noting that although methods like HPT and LwEIB attain competitive performance on certain datasets, they exhibit more fluctuation across others (e.g., relatively lower performance on EuroSAT or Aircraft), whereas MMA++ maintains better and stable results. This suggests that beyond LLM augmentation, our choices–particularly the multi-modal fusion with α-consistency framework–play a crucial role in promoting better generalization.

4) Domain Generalization: We further evaluate the robustness of the compared methods under domain shift using the Domain Generalization setting, where models are tuned on ImageNet and directly tested on four ImageNet-derived out-of-distribution (OOD) datasets: ImageNet-V2, -Sketch (-S), -Adversarial (-A), and -Rendition (-R). These datasets differ in texture, style, or acquisition conditions, making them a standard benchmark for assessing model generalization.

The results are reported in Table IV. Our proposed MMA++ achieves the best average performance (61.30) and the lowest (best) average rank (1.75) across 4 OOD datasets. Compared to earlier approaches, prompt tuning baselines such as CoOp [6] and CoCoOp [7] show limited generalization. While LLM-enhanced methods (e.g., CoPrompt [18], HPT [17], and LwEIB [30]) achieve moderate improvements over early prompt-tuning baselines, and SkipT [42] demonstrates the effectiveness of skipped fine-tuning strategies, their performance tends to fluctuate across different datasets. In contrast, MMA++ delivers more stable performance, achieving competitive or best results on most benchmarks. The consistent improvements over MMA [29] highlight the benefits of α-consistency framework, providing a more robust and reliable solution for handling domain shifts.

5) Summary of All Prior Comparisons: The comparisons across the three widely adopted experimental settings clearly validate the effectiveness of MMA++. Notably, our method achieves leading performance without relying on prior class-type knowledge at test time, which is commonly used in other methods [19], [20], [46], [47]. Unlike most existing approaches, MMA++ is grounded in a rigorous theoretical framework that not only informs its design but also elucidates its strong ability to balance adaptation and generalization within a unified model. This principled foundation distinguishes MMA++ from prior works and highlights its advantages in real-world deployment, especially for few-shot generalization under practical constraints.

TABLE IV
DOMAIN GENERALIZATION RESULTS ON FOUR OOD IMAGENET. WE EVALUATE THE SAME MODELS AS IN TABLE III ON IMAGENET-V2, -SKETCH, -A, AND -R, INCLUDING RECENT LLM-AUGMENTED METHODS [17], [18], [30]. MMA++ ACHIEVES THE HIGHEST AVERAGE ACCURACY (61.03) AND RANK (1.50). WE EXCLUDE [20], [46] BASED METHODS AS THEY RELY ON TEST-SPECIFIC TUNING OR DIFFERENT BACKBONES TRAINED WITH FULL TARGET DATA, DEVIATING THE ZERO-SHOT SETTING USED BY ALL OTHERS.

Methods	ImageNet	-V2	-S	-A	-R	Avg	Avg.Rank
CLIP [ICML2021] [2]	66.73	60.83	46.15	47.77	73.96	57.17	11.00
CoOp [IJCV2022] [6]	71.51	64.20	47.99	49.71	75.21	59.28	9.50
CoCoOp [CVPR2022] [7]	71.02	64.07	48.75	50.63	76.18	59.90	8.75
MaPLe [CVPR2023] [14]	70.72	64.07	49.15	50.90	76.98	60.26	7.25
PromptSRC [ICCV2023] [16]	71.27	64.35	49.55	50.90	77.80	60.65	4.25
CoPrompt [ICLR2024] [18]	70.80	64.25	49.43	50.50	77.51	60.42	6.50
HPT [AAAI2024] [17]	71.72	65.25	49.36	50.85	77.38	60.71	5.50
SkipT [CVPR2025] [42]	**72.77**	**65.67**	**49.73**	**51.13**	**78.27**	61.20	**2.00**
LwEIB [IJCV2025] [30]	71.31	64.47	**50.07**	51.00	**77.81**	**60.84**	2.75
MMA [CVPR2024] [29]	71.00	64.33	49.13	51.12	77.32	60.48	6.00
MMA++ [This Work]	**73.34**	**65.73**	**50.07**	**51.73**	77.67	**61.30**	**1.75**

B. Ablation Experiments

Here, we conduct all ablation studies under the Base-to-Novel generalization using handcrafted texture prompts, with detailed results and analysis reported in the following sections.

1) Variants of Adding MMA: We begin by evaluating different strategies for integrating our MMA units into the encoder. Specifically, we consider two types of insertion approaches: from X_{Embed} to the k-th layer, and from the k-th layer to the

TABLE V
ABLATION EXPERIMENTS OF OUR **MMA** OVER 11 DATASETS USED IN BASE-TO-NOVEL GENERALIZATION. "10→12" DENOTES FINE-TUNING THE LAST THREE LAYERS. MMA ACHIEVES THE HIGHEST H.M, DEMONSTRATING BETTER GENERALIZATION COMPARD WITH OTHER CHOICES.

(a) Performance with different model variants

Model Variants	Base	Novel	H.M
Only L-Adapter	80.36	75.81	78.02
Only V-Adapter	80.39	74.18	77.16
No SharedProj	82.43	76.21	79.20
FCAA [4]	79.11	75.64	77.34
MMA	**83.20**	**76.80**	**79.87**

(b) Dimensions of shared layers

Dims	Base	Novel	H.M
8	82.66	76.17	79.28
16	82.80	76.48	79.52
32	**83.20**	**76.80**	**79.87**
64	83.41	76.17	79.63
128	82.98	76.54	79.58

(c) MMA *vs*. Last few layers fine-tuning

Layer	Base	Novel	H.M
12	80.77	74.08	77.28
10→12	83.02	74.55	78.56
8→12	**83.77**	73.77	78.45
5→12	83.21	70.95	76.59
MMA	83.20	**76.80**	**79.87**

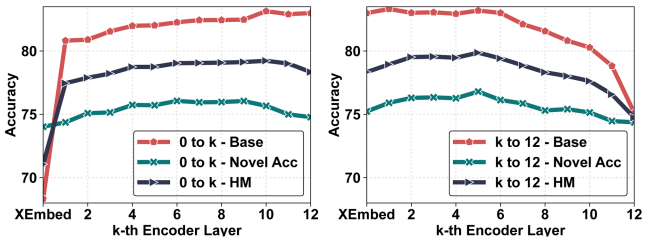

Fig. 6. Ablation studies on different insertion strategies (**left:** from 0 to k, **right:** from k to 12) for our multi-modal units. Average Base, Novel, and H.M scores are reported across 11 datasets. Insertion after $k = 5$ offers the best trade-off between discrimination and generalization (best viewed in PDF format).

final layer, where $k \in \{X_{Embed}, 1, \ldots, 12\}$. The results are shown in Fig. 6. For the first strategy, increasing k tends to enhance performance on base classes while degrading accuracy on novel classes. In contrast, for the second strategy, base class performance remains relatively stable when k is set around 5, while novel class accuracy improves significantly as k approaches 5. Notably, the highest H.M of 79.87 is achieved at $k = 5$, supporting our earlier observations regarding the optimal trade-off between discrimination and generalization.

2) Adapting Variant Options: We evaluate the effectiveness of several design choices for our MMA. These include uni-modal adapters added solely to the vision (V-) or language (L-) branches, as well as the removal of the shared projection layer (denoted as No SharedProj). To further assess adapter efficiency, we replace each MMA module with a Flamingo-style cross-attention block [4], referred to as FCAA. The average performance across 11 recognition datasets is summarized in Table V(a). Our results show that uni-modal adapters consistently underperform compared to the bi-modal configuration. In addition, incorporating the shared projection layer improves the H.M from 79.20 to 79.87, highlighting its role in promoting effective feature alignment.

3) Dimension of the Shared Layer: The dimension of shared layers in our MMA determines the number of parameters to extract relationships between the features from the two modalities. We systematically varies the dimensions of the shared layers to investigate its effects. As depicted in Table V(b), accuracy on base class is highest with an increment in the middle dimension, but the novel accuracy performance reaches a saturation point at approximately 32. This may be because a larger dimension of the shared layers incurs more trainable parameters, increasing the risk of overfitting.

4) Fine-Tuning the Last Few Layers: We further compare our MMA with a baseline approach that fine-tunes only the last few layers of the pre-trained VLM. The results are shown in Table V(c). As expected, fine-tuning more layers (e.g., 8→12) improves the performance on base classes, reaching the highest base accuracy of 83.77. However, this comes at the cost of reduced generalization to novel classes (73.77). This trend is more pronounced when fine-tuning from layer 5, where novel class accuracy further declines to 70.95. This degradation can be attributed to the overfitting of model parameters to base class distributions and the potential erosion of general knowledge encoded in the pre-trained VLM. In contrast, our MMA achieves the highest H.M (79.87), striking a better balance between base class discrimination and novel class generalization without altering the core pre-trained backbone.

5) Ablation Studies of α-Consistency Framework: To better understand the contribution of each component in MMA++, we conduct a step-by-step ablation study as shown in Table VI(a). Starting from the original MMA [29] trained for 5 epochs, we first extend the training schedule to 50 epochs, as the short training in the original setting may not sufficiently fit the training data, as confirmed in Fig. 4. While this leads to a notable improvement on the base classes (from 83.20 to 86.02), performance on novel classes drops due to potential overfitting. We then apply our proposed α-consistency training to encourage alignment between adaptation scales and sample regimes. It results in a clear better performance on novel classes (75.52) and an improved H.M (80.58). Finally, by introducing α train-test decoupling, we further enhance generalization to novel classes, achieving the best overall H.M of 80.81. These results demonstrate the effectiveness and complementarity of each design component in MMA++.

6) Different λ Values in (19): We here investigate the impact of the loss weighting parameter λ in (19). As shown in Table VI(b), we evaluate $\lambda \in \{1.0, 2.5, 5.0, 10.0\}$ to examine its influence on the balance between base and novel class performance. A smaller value of $\lambda = 1.0$ achieves the best on base classes (85.43), but underperforms on novel classes. Increasing λ to 2.5 leads to a better trade-off, significantly boosting novel class accuracy to 76.80 and yielding the highest H.M (80.81). Further increasing λ (to 5.0 and 10.0) leads to lower H.M. These results suggest that $\lambda = 2.5$ provides a stable and effective balance between adaptation and generalization.

7) Different α Values During Testing: We examine the impact of varying the test-time fusion scale α while fixing the

TABLE VI
ABLATION EXPERIMENTS OF OUR **MMA++** OVER 11 DATASETS USED IN BASE-TO-NOVEL GENERALIZATION

(a) Progressive component addition in MMA++

Method	Base	Novel	H.M
MMA (5 Epochs) [29]	83.20	76.80	77.87
+ 50 Epochs	86.02	71.25	77.94
+ α-Consistency Training	**86.38**	75.52	80.58
+ α Train-Test Decoupling	85.27	**76.80**	**80.81**

(b) Different λ in Eq. (19)

Value	Base	Novel	H.M
1.0	**85.43**	76.60	80.77
2.5	85.27	**76.80**	**80.81**
5.0	85.01	76.61	80.60
10.0	84.33	76.46	80.20

(c) Decoupled α during testing

Value	Base	Novel	H.M
0.0015	73.65	75.85	74.73
0.0045	81.01	76.63	78.76
0.0065	85.27	**76.80**	**80.81**
0.01 (training α)	**86.38**	75.52	80.58

TABLE VII
COMPARISON OF ADDITIONAL PARAMETERS (+PARAMS) AND RUNNING SPEED (FPS) USING IMAGENET DATASET. TRAIN (TEST) FPS IS TESTED WITH A BATCH SIZE OF 16 (100). ALL SPEEDS ARE TESTED ON A SINGLE GTX 8000 GPU. WE ALSO REPORT THE H.M SCORE IN THE BASE-TO-NOVEL SETTING.

Method	+Params	Train FPS	Test FPS	H.M
MMA	675 K	46	263	79.87
MMA++	675 K	31	262	80.81

training α_1 at 0.01. A small α (0.0015) leads to weak base performance (73.65) and the lowest HM (74.73), suggesting underuse of adapted features. As α increases, novel accuracy improves, with the highest HM (80.81) at $\alpha = 0.0065$. Using the training α (0.01) at test time slightly favors base classes but results in a lower HM (80.58). These results support our theoretical analysis that α should be adapted according to the number of adapted samples.

C. Computational Costs

Table VII compares the computational costs of MMA++ with the baseline MMA. MMA++ incurs a lower training FPS while maintaining similar test efficiency. This is because MMA++ introduces extra training time overhead from the α-consistency strategy, which requires two forward passes and auxiliary consistency losses during optimization. At inference, these auxiliary components are disabled, and the adapter outputs are fused using a fixed, decoupled scale with a single forward pass, resulting in the same inference procedure as MMA and thus unchanged test speed.

V. CONCLUSION

In this work, we proposed MMA and its enhanced version MMA++, a multi-modal adaptation framework designed to improve the few-shot generalization of CLIP. Our study has four main innovations: (1) a dataset-driven analysis that guides the selective insertion of adapters into higher layers, (2) a shared projection space that enhances cross-modal alignment, and (3) the α-consistency framework, which offers a theoretical grounding for a α-consistency training strategy and a train-test scale decoupling mechanism. Through comprehensive evaluations across base-to-novel generalization, cross-dataset transfer, and domain generalization, MMA++ consistently demonstrates strong performance and robustness.

These findings point to several promising directions for future research. A particularly interesting path is to establish deeper theoretical connections between multi-modal alignment and generalization error bounds, which may inspire new learning objectives or model structures. Another avenue is to explore adaptive, data-aware mechanisms for dynamically tuning the fusion scale, reducing manual effort while improving robustness. By integrating theory-driven insights with practical system design, MMA++ moves toward making large-scale vision-language models more flexible, efficient, and deployable in real-world applications.

REFERENCES

[1] C. Jia et al., "Scaling up visual and vision-language representation learning with noisy text supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 4904–4916.
[2] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.
[3] L. Yao et al., "FILIP: Fine-grained interactive language-image pre-training," in *Proc. Int. Conf. Learn. Representations*, 2021.
[4] J.-B. Alayrac et al., "Flamingo: A visual language model for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 23716–23736.
[5] C. Schuhmann et al., "LAION-5B: An open large-scale dataset for training next generation image-text models," in *Proc. 36th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track*, 2022, pp. 25278–25294. [Online]. Available: https://openreview.net/forum?id=M3Y74vmsMcY
[6] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *Int. J. Comput. Vis.*, vol. 130, no. 9, pp. 2337–2348, 2022.
[7] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Conditional prompt learning for vision-language models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16816–16825.
[8] H. Yao, R. Zhang, and C. Xu, "Visual-language prompt tuning with knowledge-guided context optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 6757–6767.
[9] A. Bulat and G. Tzimiropoulos, "LASP: Text-to-text optimization for language-aware soft prompting of vision & language models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 23232–23241.
[10] H. Yao, R. Zhang, and C. Xu, "TCP: Textual-based class-aware prompt tuning for visual-language model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 23438–23448.
[11] M. U. Khattak, M. F. Naeem, M. Naseer, L. Van Gool, and F. Tombari, "Learning to prompt with text only supervision for vision-language models," in *Proc. AAAI Conf. Artif. Intell.*, 2025, pp. 4230–4238.
[12] M. Jia et al., "Visual prompt tuning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 709–727.
[13] A. Bar, Y. Gandelsman, T. Darrell, A. Globerson, and A. Efros, "Visual prompting via image inpainting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 25005–25017.
[14] M. U. Khattak, H. Rasheed, M. Maaz, S. Khan, and F. S. Khan, "MaPLe: Multi-modal prompt learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 19113–19122.
[15] D. Lee, S. Song, J. Suh, J. Choi, S. Lee, and H. J. Kim, "Read-only prompt optimization for vision-language few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 1401–1411.

[16] M. U. Khattak, S. T. Wasim, M. Naseer, S. Khan, M.-H. Yang, and F. S. Khan, "Self-regulating prompts: Foundational model adaptation without forgetting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 15190–15200.

[17] Y. Wang, X. Jiang, D. Cheng, D. Li, and C. Zhao, "Learning hierarchical prompt with structured linguistic knowledge for vision-language models," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 5749–5757.

[18] S. Roy and A. Etemad, "Consistency-guided prompt learning for vision-language models," in *Proc. Int. Conf. Learn. Representations*, 2024.

[19] J. Zhang, S. Wu, L. Gao, H. T. Shen, and J. Song, "DePT: Decoupled prompt tuning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 12924–12933.

[20] Z. Li et al., "PromptKD: Unsupervised prompt distillation for vision-language models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 26617–26626.

[21] N. Houlsby et al., "Parameter-efficient transfer learning for NLP," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2790–2799.

[22] E. J. Hu et al., "LoRa: Low-rank adaptation of large language models," in *Proc. Int. Conf. Learn. Representations*, 2021.

[23] S. Chen et al., "AdaptFormer: Adapting vision transformers for scalable visual recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 16664–16678.

[24] A. C. Stickland and I. Murray, "BERT and PALs: Projected attention layers for efficient adaptation in multi-task learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5986–5995.

[25] P. Gao et al., "CLIP-adapter: Better vision-language models with feature adapters," *Int. J. Comput. Vis.*, vol. 132, pp. 581–595, 2023.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[27] A. Dosovitskiy et al., "An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2020.

[28] Z. Liu et al., "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[29] L. Yang, R.-Y. Zhang, Y. Wang, and X. Xie, "MMA: Multi-modal adapter for vision-language models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 23826–23837.

[30] L. Yang, R.-Y. Zhang, Q. Chen, and X. Xie, "Learning with enriched inductive biases for vision-language models," *Int. J. Comput. Vis.*, vol. 133, pp. 3746–3761, 2025.

[31] X. Zhai et al., "LiT: Zero-shot transfer with locked-image text tuning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18123–18133.

[32] C. M. Beaumont R et al., "Reproducible scaling laws for contrastive language-image learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 2818–2829.

[33] K. Kim, M. Laskin, I. Mordatch, and D. Pathak, "How to adapt your large-scale vision-and-language model," 2021.

[34] R. Zhang et al., "Tip-adapter: Training-free adaption of CLIP for few-shot classification," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 493–510.

[35] Z. Zheng, J. Wei, X. Hu, H. Zhu, and R. Nevatia, "Large language models are good prompt learners for low-shot image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 28453–28462.

[36] Z. Ma et al., "Where and what matters: Sensitivity-aware task vectors for many-shot multimodal in-context learning," in *Proc. AAAI Conf. Artif. Intell.*, 2025, pp. 76930–76966.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Ch. Assoc. Comput. Linguistics: Human Lang. Technol.*, 2018, vol. 1, pp. 4171–4186.

[38] S. Pratt, I. Covert, R. Liu, and A. Farhadi, "What does a platypus look like? Generating customized prompts for zero-shot image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 15691–15701.

[39] Y. Zhang, W. Zhu, H. Tang, Z. Ma, K. Zhou, and L. Zhang, "Dual memory networks: A versatile adaptation approach for vision-language models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 28718–28728.

[40] C. Zhou, C. C. Loy, and B. Dai, "Extract free dense labels from CLIP," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 696–712.

[41] J. Zhang, J. Song, L. Gao, N. Sebe, and H. T. Shen, "Reliable few-shot learning under dual noises," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 10, pp. 9005–9022, Oct. 2025.

[42] S. Wu, J. Zhang, P. Zeng, L. Gao, J. Song, and H. T. Shen, "Skip Tuning: Pre-trained vision-language models are effective and efficient adapters themselves," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2025, pp. 14723–14732.

[43] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, vol. 1, pp. 4582–4597.

[44] Y. Lu, J. Liu, Y. Zhang, Y. Liu, and X. Tian, "Prompt distribution learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5206–5215.

[45] H. Jiang et al., "Cross-modal adapter for text-video retrieval," *J. Pattern Recognit.*, vol. 159, 2025, Art. no. 111144.

[46] Y. Guo and X. Gu, "MMRL: Multi-modal representation learning for vision-language models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2025, pp. 25015–25025.

[47] H. Li, L. Wang, C. Wang, J. Jiang, Y. Peng, and G. Long, "DPC: Dual-prompt collaboration for tuning vision-language models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2025, pp. 25623–25632.

[48] B. Zhu, Y. Niu, Y. Han, Y. Wu, and H. Zhang, "Prompt-aligned gradient for prompt tuning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 15659–15669.

[49] A. V. D. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[50] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1521–1528.

[51] Y. Rao et al., "DenseCLIP: Language-guided dense prediction with context-aware prompting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18082–18091.

[52] P. L. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *J. Mach. Learn. Res.*, vol. 3, pp. 463–482, 2002.

[53] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2018.

[54] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15750–15758.

[55] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[56] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2004, pp. 178–178.

[57] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, "Cats and dogs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3498–3505.

[58] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2013, pp. 554–561.

[59] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. 6th Indian Conf. Comput. Vis., Graph. Image Process.*, 2008, pp. 722–729.

[60] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101–mining discriminative components with random forests," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 446–461.

[61] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," 2013, *arXiv:1306.5151*.

[62] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3485–3492.

[63] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3606–3613.

[64] P. Helber, B. Bischke, A. Dengel, and D. Borth, "EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 7, pp. 2217–2226, Jul. 2019.

[65] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.

[66] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do imagenet classifiers generalize to imagenet," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5389–5400.

[67] H. Wang, S. Ge, Z. Lipton, and E. P. Xing, "Learning robust global representations by penalizing local predictive power," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 10506–10518.

[68] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15262–15271.
[69] D. Hendrycks et al., "The many faces of robustness: A critical analysis of out-of-distribution generalization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8340–8349.

Lingxiao Yang (Member, IEEE) received the PhD degree from Hong Kong Polytechnic University, China, in 2020. He is currently an associate professor with Sun Yat-sen University. He has authored or coauthored about 40 papers in prestigious international journals and conferences including IJCV, ICML, and CVPR. His research interests include computer vision and machine learning with a focus on object recognition, video understanding, and brain-inspired computational model.

Ru-Yuan Zhang received the BS degree in psychology from Peking University in 2010, and the PhD degree in brain and cognitive sciences from the University of Rochester, USA, in 2016. He is currently a tenure-track associate professor with the School of Psychological and Cognitive Sciences and IDG McGoven Institute for Brain Research, Peking University. His interests include human visual neuroscience, human decision making and learning, computer vision, and machine reasoning. He has authored or co-authored more than 50 papers in prestigious international journals and conferences in neuroscience and machine learning, including *Nature Human Behavior*, PNAS, and *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yanchen Wang received the bachelor's degree in computer science from the University of Rochester. He was a research data analyst with Stanford University with prof. Ehsan Adeli and Feng Vankee Lin on AI, brain and cognitive science. He is currently a research staff assistant with the Columbia University Center for Theoretical Neuroscience, where he is advised by prof. Liam Paninski. He was advised by prof. Christopher Kanan and worked on continual learning. His research interests include computational neuroscience and computational cognitive science. He is passionate about understanding the inner workings of the brain.

Xiaohua Xie (Member, IEEE) received the BS degree in mathematics and applied mathematics from Shantou University in 2005, the MS degree in information and computing science and the PhD degree in applied mathematics from Sun Yat-sen University, China, in 2007 and 2010, respectively. He was an associate professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. He is currently a professor with Sun Yat-sen University. His research interests include image processing, computer vision, pattern recognition, and computer graphics. He has authored or coauthored more than 70 papers in prestigious international journals and conferences.