Proceedings of CONF-MLA 2025 Symposium: Applied Artificial Intelligence Research
DOI: 10.54254/2755-2721/2025.BJ26532

A Multi-Task Learning Framework Based on CLIP and Adapter Modules

Jiazhi Han

School of Computer Science and Technology, Harbin University of Science and Technology, Harbin, China
hanjiazhi04@163.com

Abstract. In recent years, with the rapid development of cross-modal learning, pretrained models such as CLIP have demonstrated powerful zero-shot capabilities in image-text alignment tasks, making them central to multimodal research. However, a key challenge remains: how to effectively transfer these capabilities while preserving the strengths of CLIP. To address this, we propose a parameter-efficient multi-task fine-tuning framework—Multi-Task CLIP-Adapter. By inserting lightweight Adapter modules after the frozen CLIP encoder, our method enables unified adaptation across multiple tasks, including classification, image-text retrieval, and regression. Experimental results show that our approach achieves an 8%–12% performance improvement with less than 0.2% additional parameters, while maintaining the original model's zero-shot capability. Compared to the original CLIP and conventional transfer strategies, the Multi-Task CLIP-Adapter offers significant advantages in parameter efficiency and task generalization, paving a new path for scalable applications of large multimodal models.

Keywords: CLIP, Adapter, Multi-task Learning, Zero-shot, Fine-tuning

1. Introduction

With the advancement of artificial intelligence, research in the field has gradually shifted from unimodal approaches to multimodal information fusion involving text, images, and speech [1]. Meanwhile, fueled by self-supervised pretraining and the increasing availability of computational resources, multimodal models have evolved from task-specific networks to unified vision-language frameworks [2]. Ensuring that such models perform robustly and effectively across a wide range of downstream tasks has become a central focus in both academia and industry. Early multimodal methods typically employed a dual-encoder architecture, extracting image features via ViT [3] and text features via BERT [4], with retrieval achieved through contrastive learning in a shared embedding space. Subsequently, a series of large-scale pretrained models significantly advanced cross-modal alignment. CLIP [3] pioneered the contrastive learning paradigm between images and text with natural language prompts, pretraining on 400 million image-text pairs. This enabled a single static model to perform zero-shot classification on hundreds of datasets, ushering in the era of prompt-driven visual modeling. BLIP [5] further introduced Bootstrapping Language-Image Pre-training, markedly enhancing performance on image captioning and visual question answering

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.BJ26532

(VQA). BLIP-2 [6] froze the vision encoder and incorporated a lightweight query transformer (Q-Former) to achieve efficient image-text alignment, laying the interface foundation for large multimodal language models. ALIGN [7] trained on 1.8 billion noisy web image-text pairs, demonstrated that large-scale data could compensate for label noise in retrieval tasks, achieving strong cross-domain generalization. Flamingo [8] froze the ViT-G/14 backbone and trained only the cross-modal causal attention layers, enabling the model to quickly adapt to open-domain image-text dialogues and multi-image scene understanding with as few as 16 shots, showcasing the potential for rapid instruction tuning.

As multimodal applications continue to expand, there is a growing demand for foundational models to adapt flexibly across multiple tasks. In response to this need, the Adapter technique [9] has emerged. By inserting small-scale bottleneck layers into frozen large models, Adapters enable efficient adaptation to various downstream tasks with minimal trainable parameters. This approach significantly reduces the computational cost of fine-tuning while maintaining strong performance, making it a powerful tool for enhancing the scalability of large models.Neil Houlsby et al. demonstrated that inserting Adapter layers into a frozen BERT model could achieve performance comparable to full fine-tuning on natural language inference (NLI) tasks, while drastically reducing the number of trainable parameters [10]. Similarly, Hao Chen et al. introduced convAdapter modules into the attention and MLP layers of ViT, specifically tailored for visual tasks, and achieved performance gains on the ImageNet classification benchmark [11]. Yi-Lin Sung further applied Adapters to ViLT [12], enhancing vision-language joint understanding with lightweight parameter updates, and achieved near full fine-tuning performance on VQA tasks [13].However, these improvements have largely focused on optimizing performance for single tasks, such as classification, retrieval, image captioning, or VQA, making it difficult to maintain consistent effectiveness across multiple tasks in a unified setting.

To address this limitation, we propose Multi-Task CLIP-Adapter, which introduces lightweight bottleneck adapter layers after the frozen CLIP vision and text encoders. By adding only ~0.2% trainable parameters, our method establishes a compact mapping between shared representations and task-specific semantics. Through few-shot fine-tuning, we achieve an average performance improvement of 8%–12% across three diverse tasks: image classification, image-text retrieval, and age regression, while preserving the original model's strong zero-shot capability. Extensive experiments show that our approach outperforms the original CLIP model with single-task fine-tuning in terms of parameter efficiency and multi-task generalization. This work offers a novel and practical path toward building efficient and scalable large multimodal models.

DOI: 10.54254/2755-2721/2025.BJ26532

2. Method

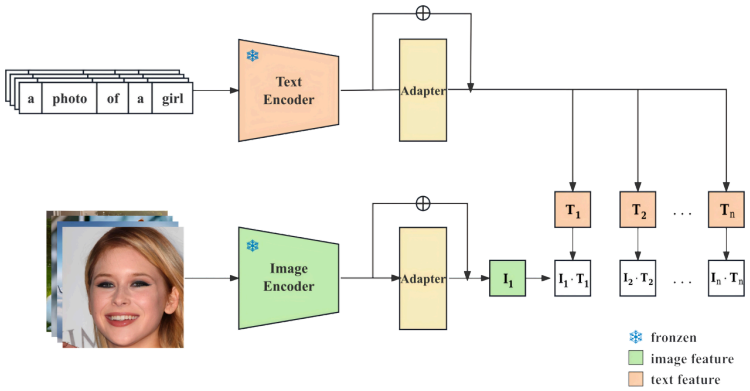

Figure 1. Structure of the Multi-Task CLIP-Adapter Mode

This study aims to achieve efficient transfer and performance enhancement of CLIP in downstream tasks while preserving the cross-modal capabilities of the pretrained model. Specifically, our approach is built upon the CLIP architecture and incorporates lightweight Adapter modules to enable parameter-efficient fine-tuning. In the following sections, we provide a detailed explanation of the CLIP backbone and the underlying principles of the Adapter mechanism.

2.1. CLIP

The CLIP framework [1], proposed by OpenAI, consists of an image encoder and a text encoder. Both encoders output d-dimensional feature representations and are jointly pretrained on large-scale image-text pairs using contrastive learning, thereby acquiring cross-modal alignment capabilities. The image encoder maps images into feature vectors, while the text encoder maps textual descriptions into corresponding feature vectors. By projecting both modalities into a shared embedding space, CLIP enables cross-modal interaction and fusion of information.

aDuring the pretraining phase, given a set of N paired samples $< \mathrm{v_i}, \mathrm{t_i} >$, each image $\mathrm{v_i}$ and its corresponding text $\mathrm{t_i}$ are encoded into feature vectors through their respective encoders:

$$\boldsymbol{z_i^v} = \boldsymbol{E_v}(\boldsymbol{v_i}) \in \mathbb{R}^{\boldsymbol{d}}, \boldsymbol{z_i^t} = \boldsymbol{E_t}(\boldsymbol{t_i}) \in \mathbb{R}^{\boldsymbol{d}}$$

To mitigate scale discrepancies between modalities, we apply L2 normalization to the image and text projection vectors independently:

$$\widehat{\mathbf{Z}}_i^v = \frac{\mathbf{Z}_i^v}{\| \mathbf{Z}_i^v \|_2}, \quad \widehat{\mathbf{Z}}_i^t = \frac{\mathbf{Z}_i^t}{\| \mathbf{Z}_i^t \|_2}$$

Given a batch containing N image-text pairs $\{(v_i, t_i)\}_{i=1}^N$, a pairwise similarity matrix is defined as follows:

DOI: 10.54254/2755-2721/2025.BJ26532

$$S_{ij} = \frac{\widehat{\mathbf{Z}}_i^v \cdot \widehat{\mathbf{Z}}_j^t}{\tau}$$

Here, $\tau > 0$ is a learnable temperature parameter. CLIP adopts a symmetric cross-entropy loss:

$$\ell_{\text{CLIP}} = -\frac{1}{2N} \sum_{i=1}^{N} \left[\log \frac{\exp\left(S_{ii}\right)}{\sum_{j=1}^{N} \exp\left(S_{ij}\right)} + \log \frac{\exp\left(S_{ii}\right)}{\sum_{j=1}^{N} \exp\left(S_{ji}\right)} \right]$$

Here, N denotes the number of image-text pairs within a training batch, and $\mathrm{S_{ij}}$ represents the similarity between the i-th image encoding and the j-th text encoding.

The symmetric cross-entropy loss encourages high similarity for matched image-text pairs while suppressing the similarity of mismatched pairs.After large-scale pretraining on 400 million image-text pairs collected from the web, CLIP demonstrates strong zero-shot capabilities in classification, retrieval, and semantic alignment tasks, laying a solid foundation for efficient downstream fine-tuning.

2.2. Adapter

Traditional full fine-tuning strategies require updating all parameters of the entire model, which incurs high computational costs, large memory consumption, and often leads to catastrophic forgetting—undermining the performance of the pretrained model on foundational tasks.To address these issues, this study draws on successful practices from the field of natural language processing and introduces lightweight adapter modules to enable efficient fine-tuning of the CLIP model.

In order to retain CLIP's zero-shot generalization capabilities while adapting effectively to a variety of downstream tasks—including image classification, cross-modal retrieval, and age regression—lightweight bottleneck adapters are inserted after both the image encoder and text encoder of CLIP.These adapters allow for parameter-efficient fine-tuning of the large-scale pretrained model by introducing only a small number of trainable parameters, while keeping the original weights frozen. This approach not only captures task-specific information but also preserves the original cross-modal alignment to the greatest extent, thereby achieving both efficiency and performance gains. The implementation details are as follows:

After the raw input is processed by either the image encoder or the text encoder, it produces an output feature vector $\mathrm{h} \in \mathbb{R}^{\mathrm{d}}$. The Adapter module first applies a down-projection to h:

$$\mathbf{a} = \mathrm{ReLU}\left(W\,\mathbf{h} + b\right), \quad W \in \mathbb{R}^{r \times d},\, b \in \mathbb{R}^{r},$$

Here, r is the bottleneck dimension, which is much smaller than d, and a ReLU activation is applied to introduce non-linear capability. Subsequently, a is up-projected and added to the original feature via a residual connection:

DOI: 10.54254/2755-2721/2025.BJ26532

$$\mathbf{h}\prime = \mathbf{h} + W\,\mathbf{a} + b, \quad W \in \mathbb{R}^{d\times r},\, b \in \mathbb{R}^{d}.$$

By introducing non-linear transformations and residual connections, the Adapter module not only enhances the model's representational capacity but also effectively mitigates issues such as gradient vanishing and performance degradation. Compared to full fine-tuning, Adapter-based tuning introduces less than 0.2% additional parameters, significantly reducing training overhead and making it well-suited for large-scale deployment and multi-task parallel scenarios.

Unlike prompt-based tuning methods such as CoOp, which rely on learnable prompts, Adapter modules are inserted after both the image encoder and the text encoder to enable the capture of task-specific features across different modalities. The overall model architecture is illustrated in Figure 1.

During fine-tuning, all original weights of CLIP—including those of the encoders and the projection heads—remain frozen, and only the parameters of the Adapter modules and the task-specific heads are updated. This design not only preserves the strong semantic alignment capabilities of the pretrained model but also significantly improves generalization performance on downstream tasks.

3. Experiments

3.1. Experimental setup

The experiments were conducted on a Windows operating system with an NVIDIA RTX 3060 GPU (8GB memory). The software environment included Python 3.10 and the deep learning framework PyTorch 2.1.2 with CUDA 12.1 support.

3.2. Evaluation metrics

3.2.1. Evaluation metrics for classification tasks

For classification tasks, we evaluate model performance using four metrics: Accuracy, Precision, Recall, and F1-score.

Accuracy measures the proportion of correctly classified samples to the total number of samples, and is defined as:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

Here, TP denotes the number of true positives correctly predicted as positive, TN denotes the number of true negatives correctly predicted as negative, FP refers to the number of false positives (incorrectly predicted as positive), and FN refers to the number of false negatives (incorrectly predicted as negative).

Precision measures the proportion of correctly predicted positive instances among all predicted positives, and is defined as:

DOI: 10.54254/2755-2721/2025.BJ26532

$$Precision = \frac{TP}{TP + FP}$$

Recall measures the proportion of actual positive instances that are correctly predicted as positive, and is defined as:

$$Recall = \frac{TP}{TP + FN}$$

F1-score measures the harmonic mean of Precision and Recall, aiming to balance both metrics. It is defined as:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

3.2.2. Evaluation metrics for text-to-image retrieval

In image-text retrieval tasks, we primarily evaluate model performance using Precision@K and mean Average Precision at K (mAP@K).

Precision@K measures the proportion of relevant items among the top-K retrieved results. Specifically, Precision@5 and Precision@10 used in this study indicate the proportion of relevant (i.e., correctly matched) items within the top 5 and top 10 retrieved results, respectively.

A higher Precision@K value implies that the model retrieves more relevant content in the top results, reflecting better real-world retrieval effectiveness. The definition of Precision@K is as follows:

$$Precision@K = \frac{T}{K}$$

Here, T denotes the number of relevant items among the top-K retrieved results.

mAP@K (mean Average Precision at K) represents the mean of the average precision scores across all queries. For each query, the average precision is calculated based on the top-K retrieved results, and then the mean is taken over all queries.

This metric considers both the relevance and the ranking position of retrieved items, making it a comprehensive indicator of retrieval performance. The definition of mAP@K is as follows:

$$MAP = \frac{1}{Q} \sum_{q=1}^{Q} AP(q)$$

where $AP(q)$ is the average precision for the q-th query within the top-K results, defined as:

DOI: 10.54254/2755-2721/2025.BJ26532

$$AP = \frac{\sum_{k=1}^{n} (Precision@k \times rel(k))}{number\ of\ relevant\ items}$$

$rel(k)$ is an indicator function that equals 1 if the item at position k is relevant, and 0 otherwise..

3.2.3. Regression task metrics

For regression tasks, we evaluate model prediction performance using the following four metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

MAE measures the average absolute difference between the predicted values and the true values, and is defined as:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

where y_i is the ground truth value of the i-th sample, $\hat{y}_i$ is its predicted value, and n is the total number of samples.

MSE calculates the mean of the squared differences between predicted and true values, and is defined as:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{y}_i \right)^2$$

RMSE is the square root of MSE. It is also used to quantify the difference between predicted values and actual values, but by converting the error into the same unit as the target variable, it provides a more intuitive interpretation of the model's prediction error. RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{y}_i \right)^2}$$

3.3. Comparative experiments

To verify the effectiveness of the proposed method in multi-task scenarios, we use the original CLIP model as a baseline and introduce Adapter modules to construct the Multi-Task CLIP-Adapter model. Comparative experiments are conducted on three tasks: image classification, image-text retrieval, and age regression. The experimental results are summarized in Table 1.

DOI: 10.54254/2755-2721/2025.BJ26532

Table 1. Comparative results between the proposed method and the baseline model

Method	Classification				Text-to-Image Retrieval			Age Regression		
	Accuracy	Precision	Recall	F1-score	Precision@5	Precision@10	mAP@10	MAE	MSE	RMSE
CLIP	0.8779	0.8817	0.8779	0.8779	0.96	0.96	0.9801	6.18	67.14	8.19
Multi-Task CLIP-Adapter	0.9435	0.9438	0.9435	0.9435	0.98	0.99	0.9835	5.28	49.37	7.03

The evaluation of the model's classification capability was conducted on the CIFAR-100 dataset, with Accuracy, Precision, Recall, and F1-score on the test set as the main metrics. Experimental results show that the original CLIP model achieved an accuracy of 0.8779, a precision of 0.8817, a recall of 0.8779, and an F1-score of 0.8779, indicating stable overall performance.

After introducing Adapter modules and freezing all parameters of the original model (training only the Adapter for 5 epochs), the Multi-Task CLIP-Adapter model demonstrated significant improvements across all four metrics. The accuracy and F1-score both increased to 0.9435, suggesting that the lightweight Adapter mechanism effectively enhances the model's feature extraction and discrimination capabilities in classification tasks.

For the image-text retrieval task, experiments were also conducted on the CIFAR-100 dataset. Given a textual description, the model is required to retrieve the most relevant image from a collection. We adopt Precision@5, Precision@10, and mAP@10 as evaluation metrics to assess retrieval relevance in the top-ranked results. The original CLIP model achieved 0.96 for both Precision@5 and Precision@10, and 0.9801 for mAP@10. In contrast, after training under the same conditions, the Multi-Task CLIP-Adapter improved to 0.98 on Precision@5, 0.99 on Precision@10, and 0.9835 on mAP@10. These results demonstrate that the introduction of Adapters not only preserves the cross-modal alignment capability of CLIP, but also enhances retrieval performance in top-ranked results, indicating stronger practical retrieval ability.

In the age regression task, the CelebA dataset was used for training and evaluation. The goal is to predict the numerical age from a given facial image. The performance was assessed using MAE, MSE, and RMSE. With a simple linear head trained for 5 epochs, the original CLIP model achieved an MAE of 6.18, MSE of 67.14, and RMSE of 8.19. In comparison, the Multi-Task CLIP-Adapter, incorporating both Adapter and Linear modules and trained for the same number of epochs, reduced the MAE to 5.28, and the MSE and RMSE to 49.37 and 7.03, respectively, showing superior numerical regression fitting ability.

Overall, across the three tasks, the Multi-Task CLIP-Adapter significantly improves accuracy in image classification and image-text retrieval tasks, and reduces prediction error in the age regression task, all without requiring full-parameter updates of the original large-scale model. This method effectively balances computational efficiency with performance gains, demonstrating strong generalization and transferability, and providing an efficient and practical solution for lightweight optimization in multi-task scenarios.

DOI: 10.54254/2755-2721/2025.BJ26532

3.4. Ablation study

Table 2. Comparison between CLIP and ResNet18

Method	Classification				Age Regression		
	Accuracy	Precision	Recall	F1-score	MAE	MSE	RMSE
CLIP	0.8779	0.8817	0.8779	0.8779	6.18	67.14	8.19
ResNet18	0.1316	0.1074	0.1316	0.1005	40.39	1930.69	43.94

In the classification task, we employed the classical convolutional neural network ResNet18, which was pretrained on the CIFAR-100 dataset. After pretraining, the model was transferred to the CIFAR-10 dataset, with the output layer replaced by a 10-class classifier. Under zero-shot conditions —i.e., without any further fine-tuning—the model achieved only 13.16% accuracy on the CIFAR-10 test set.In contrast, we evaluated the CLIP model on the same test set under zero-shot inference conditions. Without any fine-tuning, CLIP achieved a classification accuracy of 87.79%, with Precision, Recall, and F1-score all exceeding 87%, significantly outperforming the traditional convolutional model.

In the facial age regression task, we transferred the pretrained ResNet18 model to the CelebA dataset, using facial images as input to predict corresponding ages. The performance was evaluated using MAE, MSE, and RMSE. Results show that the CLIP model also maintained strong performance in the regression task, achieving an MAE of 6.18 and an RMSE of 8.19, indicating low prediction error and high stability.In contrast, the ResNet18 model performed poorly, with an MAE as high as 40.39 and an RMSE of 43.94, reflecting a significant level of error that fails to meet even the minimum usability standards.

These experimental results demonstrate that traditional models exhibit poor generalization when confronted with datasets involving distribution shifts or semantic transfer. In contrast, the CLIP model, pretrained through large-scale image-text contrastive learning, successfully aligns visual and textual semantic spaces. This endows CLIP with strong semantic generalization and zero-shot inference capabilities, enabling accurate classification without the need for additional training.Overall, the experiments highlight the potential of CLIP for cross-semantic understanding in vision transfer tasks.

4. Conclusions

This paper addresses the adaptation challenges of current large-scale multimodal models in multi-task transfer scenarios by proposing the Multi-Task CLIP-Adapter approach. Without compromising the original pretrained capabilities of CLIP, the proposed method enables efficient transfer and performance enhancement across three representative tasks: image classification, image-text retrieval, and age regression.Experimental results demonstrate that the proposed method significantly outperforms the original CLIP model in all multi-task scenarios, achieving consistent improvements across all evaluation metrics. Notably, while preserving CLIP's zero-shot capabilities, our method introduces only 0.2% additional trainable parameters and achieves over 10% average performance gain. Furthermore, ablation studies confirm the advantages of our approach in terms of model efficiency and transferability, showcasing its promising potential for real-world applications.In future work, we plan to explore the applicability of this method to more complex tasks such as multilingual understanding and cross-modal generation. We also aim to further

DOI: 10.54254/2755-2721/2025.BJ26532

improve the flexibility and generalization of multi-task learning by integrating mechanisms such as prompt tuning and LoRA.

References

[1] Baltrušaitis T, Ahuja C, Morency L P. Multimodal machine learning: A survey and taxonomy [J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(2): 423-443.
[2] Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision [C]//International conference on machine learning. PmLR, 2021: 8748-8763.
[3] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale [J]. arXiv preprint arXiv: 2010.11929, 2020.
[4] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding [C]//Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). 2019: 4171-4186.
[5] Li J, Li D, Xiong C, et al. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation [C]//International conference on machine learning. PMLR, 2022: 12888-12900.
[6] Li J, Li D, Savarese S, et al. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models [C]//International conference on machine learning. PMLR, 2023: 19730-19742.
[7] Jia C, Yang Y, Xia Y, et al. Scaling up visual and vision-language representation learning with noisy text supervision [C]//International conference on machine learning. PMLR, 2021: 4904-4916.
[8] Alayrac J B, Donahue J, Luc P, et al. Flamingo: a visual language model for few-shot learning [J]. Advances in neural information processing systems, 2022, 35: 23716-23736.
[9] Gao P, Geng S, Zhang R, et al. Clip-adapter: Better vision-language models with feature adapters [J]. International Journal of Computer Vision, 2024, 132(2): 581-595.
[10] Houlsby N, Giurgiu A, Jastrzebski S, et al. Parameter-efficient transfer learning for NLP [C]//International conference on machine learning. PMLR, 2019: 2790-2799.
[11] Chen H, Tao R, Zhang H, et al. Conv-adapter: Exploring parameter efficient transfer learning for convnets [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 1551-1561.
[12] Kim W, Son B, Kim I. Vilt: Vision-and-language transformer without convolution or region supervision [C]//International conference on machine learning. PMLR, 2021: 5583-5594.
[13] Sung Y L, Cho J, Bansal M. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks [C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 5227-5237.