Information Fusion 104 (2024) 102204

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full Length Article

Instruction-ViT: Multi-modal prompts for instruction learning in vision transformer

Zhenxiang Xiao [a,b,1], Yuzhong Chen [a,b,1], Junjie Yao [a,b], Lu Zhang [d], Zhengliang Liu [c], Zihao Wu [c], Xiaowei Yu [d], Yi Pan [a,b,e], Lin Zhao [c], Chong Ma [f], Xinyu Liu [g], Wei Liu [h], Xiang Li [i], Yixuan Yuan [g], Dinggang Shen [j,k,l], Dajiang Zhu [d], Dezhong Yao [a,m,n], Tianming Liu [c], Xi Jiang [a,b,*]

[a] The Clinical Hospital of Chengdu Brain Science Institute, MOE Key Laboratory for NeuroInformation, School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, 611731, China
[b] High-Field Magnetic Resonance Brain Imaging Key Laboratory of Sichuan Province, University of Electronic Science and Technology of China, Chengdu, 611731, China
[c] School of Computing, The University of Georgia, Athens, 30602, USA
[d] Department of Computer Science and Engineering, The University of Texas at Arlington, Arlington, 76019, USA
[e] Glasgow College, University of Electronic Science and Technology of China, Chengdu, 611731, China
[f] School of Automation, Northwestern Polytechnical University, Xi'an, 710072, China
[g] Department of Electronic Engineering, Chinese University of Hong Kong, 999077, Hong Kong, China
[h] Department of Radiation Oncology, Mayo Clinic, Phoenix AZ, 85054, USA
[i] Department of Radiology, Massachusetts General Hospital and Harvard Medical School, Boston, 02115, USA
[j] School of Biomedical Engineering, ShanghaiTech University, Shanghai, 201210, China
[k] Shanghai United Imaging Intelligence Co., Ltd., Shanghai, 200230, China
[l] Shanghai Clinical Research and Trial Center, Shanghai, 201210, China
[m] Research Unit of NeuroInformation (2019RU035), Chinese Academy of Medical Sciences, Chengdu, 611731, China
[n] School of Electrical Engineering, Zhengzhou University, Zhengzhou, 450001, China

ARTICLE INFO

Keywords:
Multi-modal information fusion
Multi-modal prompt
Instruction learning
Vision transformer

ABSTRACT

Prompts play a crucial role in enhancing the control, adaptability, and scalable application of large language models. In recent years, strategies involving prompts have also been applied to visual models. However, the extent to which the fusion of multi-modal prompts (e.g., text or image prompts) can improve downstream task performance in visual models has not been systematically investigated. To address this issue, this paper focuses on adapting the design of prompts based on instruction tuning in a vision transformer model for visual tasks, which we have named Instruction-ViT. The key idea involves implementing and fusing multi-modal prompts (either text or image prompts) related to category information, guiding the fine-tuning of the model. Based on the experiments conducted on several image understanding tasks, including classification, segmentation, image captioning, and object detection, we observe consistently improved performance and domain adaptability. Our work presents an innovative strategy for fusing multi-modal prompts, enhancing performance and adaptability in visual models.

1. Introduction

A long-standing goal of humanity has been to develop Artificial General Intelligence (AGI) that exhibits human-level intelligence or surpasses it. A key characteristic of human intelligence is its ability to process information from multiple modalities, enabling individuals to understand their surroundings through various information sources and communicate effectively with others [1]. Similarly, AGI systems are also expected to efficiently handle, integrate, and utilize multimodal data to solve real-world problems. Recent breakthroughs in Large Language Models (LLMs) have provided new insights toward realizing this goal. Prompt tuning and instruction tuning were initially introduced in the field of Natural Language Processing (NLP) to address a variety of complex tasks. Subsequently, LLMs have demonstrated remarkable capabilities in both learning and reasoning. Unlike traditional language

[*] Corresponding author at: The Clinical Hospital of Chengdu Brain Science Institute, MOE Key Laboratory for NeuroInformation, School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, 611731, China.
E-mail address: xijiang@uestc.edu.cn (X. Jiang).

[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.inffus.2023.102204
Received 22 July 2023; Received in revised form 13 December 2023; Accepted 15 December 2023
Available online 18 December 2023

models, generative LLMs employ a novel prompt methodology that enables pre-trained models to facilitate downstream tasks without the need for further model fine-tuning. Through flexible prompt and instruction designs, as illustrated in Fig. 1(a–b), these models can be pre-trained on vast amounts of raw text and perform few-shot or even zero-shot learning. This enables them to adapt to new scenarios with minimal or no labeled data [2]. For instance, the in-context learning capabilities demonstrated by GPT-3 [3] and ChatGPT [4] enable these models to produce ideal results for previously unseen tasks without the need to update any parameters [5].

While large-scale uni-modal (images or texts) models have demonstrated impressive performance in a variety of tasks [3,6], the complexity and diversity of many real-world problems in artificial intelligence often require the integration of information from multiple modalities, such as text, image, and audio. Multi-modal models employ various methods to integrate data from different modalities. These methods are typically categorized as early (feature) fusion, late (decision) fusion, or intermediate (hybrid) fusion, depending on the stage in the network at which representations are combined [7]. However, the choice of the fusion method remains highly dependent on the specific domain, data, and task, and there are currently no universal fusion rules. Multi-modal models have demonstrated much potential in improving performance across a variety of tasks, such as speaker diarization [8], text-to-image generation [9], and image description [10]. However, they fall short in their capabilities for some specific tasks, such as segmentation [6].

Recently, several studies have focused on introducing prompts into vision and multi-modal models [11]. For example, as demonstrated in Fig. 1(c), using a pair of input–output images as visual prompts, the model can automatically generate an output image consistent with these examples for any new input image [12], and it can perform multiple tasks, such as segmentation. The Segment Anything Model (SAM) demonstrates its impressive segmentation capabilities across numerous tasks by incorporating segmentation prompts including points, boxes, text, and masks [6,13,14]. Additionally, the Context Optimization (CoOp) approach [15] has improved the CLIP model [16], aligning the image and text more effectively, as shown in Fig. 1(d), by adding an additional learnable context prompt to enhance zero-shot learning capabilities. In our study, we propose the Instruction-ViT model, as shown in Fig. 1(e). This model fuses multi-/single-modal prompts with input images and employs distinct task modules to accomplish a variety of downstream tasks.

Our contributions and main findings are summarized as follows:

(1) We have successfully incorporated instruction tuning into the vision transformer model. This approach allows the use of both images and text as instruction prompts to guide the tuning process.

(2) In conducting experiments on various image captioning tasks, our Instruction-ViT model demonstrated improved performance and stronger adaptability.

2. Related work

2.1. Large-scale multi-modal Models

Large-scale multi-modal Models (LMMs) are developed using extensive datasets, enabling them to effectively process multiple modalities, especially in visual-language downstream tasks. For example, UNITER achieves state-of-the-art performance on various downstream tasks by jointly encoding textual and visual information in a shared representation space [17]. CLIP utilizes different encoders for images and text, matching them in the latent space to achieve a powerful multi-modal encoder model [16]. ALIGN also uses a dual-encoder architecture to align visual and language representations by training image–text pairs without manual annotations [18]. BLIP pre-trains a multi-modal mixture of encoder–decoder model to tackle both understanding-based and generation-based tasks [19]. Flamingo is a family of visual language models trained on large-scale multi-modal web corpora, and can easily adapt to both classification and generation tasks [20]. GPT-4, the latest version of GPT models, is a large-scale multi-modal model that is expected to be able to process multiple types of data, including texts, images, audio, and video [10]. The use of these LMMs' pre-trained visual-language models has become increasingly significant. In contrast to the common practice of directly aligning text and images in existing models, our method uniquely focuses on utilizing multi-modal prompts, aiming to significantly enhance the performance of visual tasks.

2.2. Multi-modal prompt tuning

Prompt tuning, a technique initially utilized in NLP, is designed to improve the performance of language models [21]. This method involves fine-tuning a pre-trained language model for a specific task by using a set of relevant prompts or examples [22]. Unlike conventional fine-tuning, which often requires modifying the weights or parameters of the pre-trained model, prompt tuning does not necessitate any alterations to these pre-existing model weights. Moreover, prompt tuning shows competitive advantages, especially at larger scales, in an era where models with billions of parameters are becoming increasingly common.

Visual Prompt Tuning [23] introduces the prompts to vision models, which only trains very few parameters to achieve higher classification accuracy than the full fine-tuning method. The ViPT uses other tracking modal images to improve the performance of object tracking tasks based on RPG modal data [24]. While these approaches achieve better results compared to a single visual modal, the text modal is ignored. The VPTG fuses the visual modal data to the text modal and obtains superior performance than the text-only model in the visual dialogue task [25]. Similarly, the V2P uses the Swin Transformer based image encoder to generate the attributes of the image as the prompt [26]. These attribute prompts are utilized to complete summary generation tasks resulting in better performance. While these works demonstrate that methods utilizing visual data as prompts achieve better performances than those merely using single-modal data, the existing methods do not consider the situation using multi-modal prompts. The NewsMEP based on concatenated multi-modal prompts achieves better results in image captioning task [27]. For vision-language models, CoOp adds an additional learnable context prompt to the input of the text encoder to enhance the zero-shot learning capability [15]. To further improve the class shift robustness of CoOp, the CoCoOp embeds the instance-conditional token on the image encoder features from the basis of the context token [28]. However, these methods only use multi-modal prompts for a single downstream task or only generate a single-modal prompt.

2.3. Instruction tuning

Instruction fine-tuning, also known as instruction tuning, is a fine-tuning technique initially introduced for LLMs [29]. Rather than fine-tuning on a specific downstream task as in BERT-based [30] model tuning, instruction tuning employs data comprising concise instructions and corresponding outputs across a diverse range of tasks and domains. In the field of Natural Language Processing (NLP), Ouyang et al. [31] built upon GPT-3 [3] and utilized instruction tuning with Reinforcement Learning from Human Feedback (RLHF) [32] to develop InstructGPT. This model better aligns its responses with user intent and minimizes the output of untruthful and toxic content. Further application of instruction tuning led OpenAI to introduce ChatGPT and GPT-4 [10], representing significant advancements towards AGI models [1] and potential applications in various fields such as computer-aided diagnosis in medical imaging [33,34]. Another recently released instruction-tuned model, Alpaca, leveraged GPT-3.5 to generate a 52k instruction-following dataset [35], which was then used to fine-tune LLaMA 7b [36]. This approach achieves comparable performance to GPT-3.5 on a smaller scale and with fewer computational resources [37].

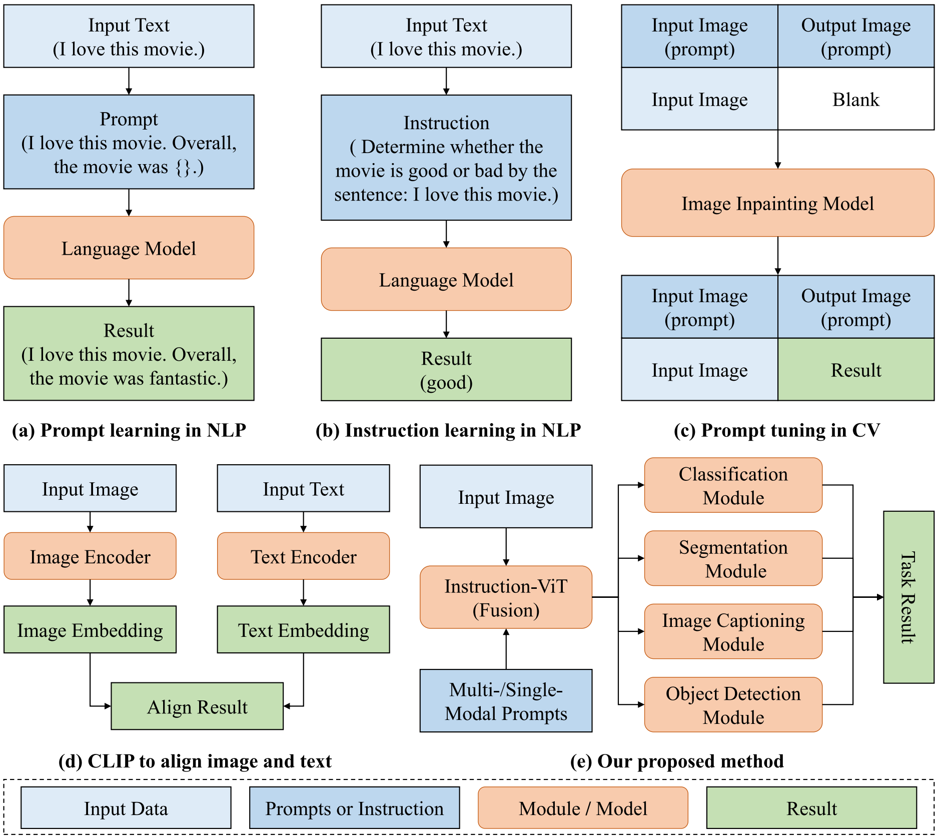

Fig. 1. Comparison of our method with other established methods includes: (a) A common method of prompt learning in Natural Language Processing (NLP). (b) A typical approach to instruction learning in NLP. (c) An application of visual prompts in computer vision (CV), utilizing an image inpainting model to generate missing parts of images and unify multiple CV tasks. (d) The pipeline of the CLIP model for aligning information from text and image. (e) Our proposed Instruction-ViT model, which fuses multi-/single-modal prompts with the input image and adopts different task modules to achieve various downstream tasks.

Beyond NLP, recent work has expanded the use of instruction tuning to multi-modal model fine-tuning. Liu et al. [38] leveraged GPT-4 to generate instruction-following data [39] based on images and the corresponding captions. In addition, the GPT-4V demonstrates amazing visual reasoning ability under the guidance of text instruction in both the natural and medical image domains [40,41]. The resulting LLaVA model demonstrates competitive results with GPT-4 on visual and language understanding tasks. By simply modifying the module of LLaVA, LLaVA-1.5 [42] achieved improved performance on various tasks. Similar to the approach described above, our work utilizes instruction tuning to fine-tune our model for specific visual tasks and to emphasize the adaptability and efficacy of multi-modal prompts.

3. Methods

We propose Instruction-ViT, a unified framework that aligns the input of images and prompts. In this section, we will first introduce the proposed method for creating prompt tokens, outline the backbone of the model, and discuss how we perform various downstream tasks. Then, we will introduce our strategy for selecting prompts and for interpretability analysis.

3.1. Prompt

We construct the prompt as shown in the bottom right of Fig. 2. In our work, we use the text of the class name, the corresponding image in the training dataset, as well as the combination of text and image as our prompts, respectively. For the text prompt, we use 80 sentence templates with the form like $a\ photo\ of\ a\ \{Class\ Name\}$ in the same way as OpenCLIP [43]. After that, we use the pre-trained CLIP text encoder as our prompt encoder, the 80 constructed text prompts as input, and obtain the average result as the prompt token of text x_{pt} as shown in the formula:

$$x_{pt} = \frac{\sum_{k=1}^{80} x_{pt_k}}{80} \tag{1}$$

where x_{pt_k} represents the prompt generated from kth text template. CLIP's pre-trained image encoder is used as the Prompt Encoder for generating image prompt tokens x_{pi}. Specifically, we randomly select an image from the training set as the prompt image and embed the image to tokens by the encoder. By averaging the text and image prompt tokens together [44], we obtain the mixed prompt tokens x_{pm} for each class, as shown in the formula:

$$x_{pm} = \frac{x_{pi} + x_{pt}}{2} \tag{2}$$

For comparison, we also use BERT as the Prompt Encoder [30] with the same process as constructing the text prompt tokens, and end up with the prompt tokens x_{pB}. Finally, we generate four types of prompts, and the input prompt tokens are represented as $x_p = \{x \mid x \in x_{pt}\ or\ x \in x_{pi}\ or\ x \in x_{pm}\ or\ x \in x_{pB}\}$ in this work.

3.2. Instruction prompt in vision transformer

As shown in Fig. 2, we adopt ViT as the backbone of our model [45]. For the input of the transformer module, we create a learnable [CLS] token x_{cls} to represent global image features and to extract prompt features. The input image, which is divided into patches and encoded into a sequence of patch embeddings x_{im} by the Embed module, forms the other part. Additionally, we add positional embeddings to x_{cls} and x_{im} to retain positional information. The final part is the instruction

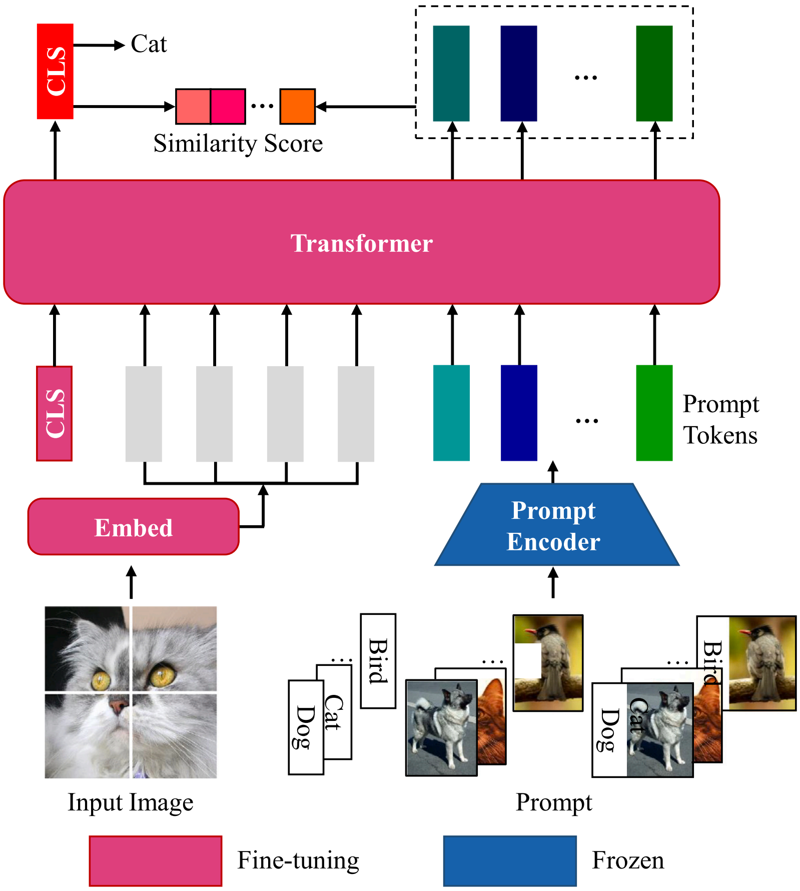

Fig. 2. The overall framework of Instruction-ViT for classification. For each image input, the corresponding latent text or visual features are considered as prompts. We utilize the Transformer's attention mechanism to combine the features of the input image and prompts. The CLS token is used to complete the downstream classification task, and the similarity scores computed by the CLS and prompt tokens are used to assist in the fine-tuning of the model. During the training stage, the pink module is fine-tuned while the navy blue module remains frozen.

Algorithm 1 The pseudo code of Instruction-ViT for classification

```
Input: Images, Prompt
Output: SimScore, Pred
if Prompt not suitable to Images then
    PromptEmb ← ResetPrompt(Prompt);
end
for each Im in Images do
    ImPatchs ← ImageToPatch(Im);
    ClsToken ← Zeros(SameEmbSize(ImPatchs));
    x ← Concate(ClsToken, ImPatchs, PromptEmb);
    x ← Transformer(x);
    ClsToken, ImPatchs, PromptEmb ← x;
    Pred ← ClassificationHead(ClsToken);
    SimScore ← Distance(ClsToken, PromptEmb);
end
```

prompt x_p. Therefore, we can represent the input of our Transformer module as:

$$x_{in} = \begin{bmatrix} x_{cls} & x_{im} & x_p \end{bmatrix} \tag{3}$$

where x_{cls}, x_{im}, and x_p represent the [CLS] token, input image patch embeddings, and prompt tokens, respectively. The input x_{in} is then fed into the Transformer module and uses the self-attention mechanism so that the [CLS] token can utilize features from both x_{im} and x_p. We demonstrate the core implementation of our work in Algorithm 1.

3.3. Downstream task and loss construction

Our model is built upon the ViT-based backbone, enabling it to effectively perform various tasks including classification, segmentation, image captioning, and object detection via using different function heads. In our work, we have successfully implemented and accomplished these tasks using our model, demonstrating its flexibility and adaptability.

3.3.1. Classification

In our work, we add a classification head after the CLS token to accomplish the classification task. For the predicted result y_{pred}, we use the cross-entropy loss as the loss function in the classification task, which is defined as:

$$loss_{pred} = CELoss(y_{pred}, target) \tag{4}$$

where $target$ is the ground truth, and $CELoss$ is the function to calculate cross-entropy loss.

Similarly to VPT [23], we evaluate the overall performance of our model by fine-tuning only a subset of parameters. Specifically, in line with VPT, we fine-tune the head function and the prompt part in our model solely. Furthermore, considering the specific context in which VPT is employed, we also train our model using only a small number of samples from each class for training.

Furthermore, we conduct additional comparisons by replacing our backbone with DeiT [46] as well as using a BERT-generated prompt [30]. In the case of backbone replacement, the structure of DeiT closely resembles ViT except for the distillation token, resulting in seamlessly replacement of ViT backbone with DeiT. Similarly, we directly replace the prompt with the one from BERT with same dimension.

3.3.2. Segmentation

We perform the segmentation task by replacing the function head with a segmentation head. Specifically, our head function is composed of two CNN layers and one interpolation layer, which can predict patch embedding to a segmentation result with the original image size. For the segmentation task, we use DiceLoss as the loss function which is represented as:

$$loss_{pred} = DiceLoss(y_{seg}, targets) \tag{5}$$

where y_{seg} is the predicted segmentation result and $targets$ is the ground truth of segmentation.

3.3.3. Image captioning

To perform the image captioning task, we employ LSTM as our decoder, which is a widely adopted choice to effectively generate captions for the input images [47]. For the training process, we utilize the cross-entropy function *CELoss* as our loss function:

$$loss_{IC} = CELoss(y_{IC}, targets) \tag{6}$$

where y_{IC} is the prediction score of image captioning, and $targets$ is the score of ground truth.

3.3.4. Object detection

To perform the object detection task, we follow a previously established method [48] that utilizes ViT as the backbone for object detection. We simply switch out the backbone with our Instruction-ViT to effectively detect and identify targets in the given input image.

3.3.5. Loss construction

To optimize the participation of different modal prompts in the classification process, we calculate the cosine similarity score [45] between the output [CLS] token and prompt tokens to improve the alignment and uniformity, which are the important properties in contrastive learning [49]. The similarity score is calculated as:

$$Score = y_{cls} \cdot y_p^T \tag{7}$$

where y_{cls} represents the output of [CLS] token, and y_p represents the output of prompt tokens. Both y_{cls} and y_p are L2 regularized. For image segmentation, the similarity score is calculated as:

$$Score = y_{patch} \cdot y_p^T \tag{8}$$

where y_{patch} is the patch embedding of the output of Transformer with L2 regularization. Since the class having the largest similarity score with the CLS token is the targeted class, we use the similarity score as part of the loss which is defined as:

$$loss_{score} = -\log \frac{\exp(z_+)}{\sum \exp(z_i)} \tag{9}$$

where z_+ represents the similarity score of the target sample and z_i represents each similarity score. The final loss is then represented as:

$$loss = loss_{pred} + loss_{score} \tag{10}$$

3.4. Prompt selection

To ensure that the input image remains the main focus and to improve computational efficiency, we limit the number of input prompts during validation. We carry out an initial filtering of potential classes and illustrate the overall framework of prompt selection in Fig. 3. Specifically, we extract features from both the input image I_{im} and text templates I_{text} using the pre-trained CLIP image encoder ENC_{image} and text encoder ENC_{text} respectively, and calculate their similarity score, which is represented as follows:

$$F_{image} = L_2(ENC_{image}(I_{im})) \tag{11}$$

$$F_{text} = L_2(ENC_{text}(I_{text})) \tag{12}$$

$$Score = F_{image} \cdot F_{text}^T \tag{13}$$

where L_2 represents the *L2*-regularization, $F_{text} \in \mathbb{R}^{N\times Dim}$ represents the extracted text features and $F_{image} \in \mathbb{R}^{1\times Dim}$ represents the extracted image features. We then select K prompts with the highest similarity as the input prompts to the next module. For the other N-K prompt tokens, we calculate their average value and utilize it as an additional input prompt. By selecting K+1 prompt token, we can effectively reduce the computational time.

3.5. Interpretability analysis

We use the Grad-CAM method to generate visual explanations from our model [50]. Specifically, we extract the forward propagation parameters P_{for} and backward propagation parameters P_{back} of the last attention layer of Transformer [45]. The visual explanations result L is represented as:

$$L = Func_i(ReLU(\sum_{head=1}^{n}(P_{for} \odot P_{back})/n)) \tag{14}$$

where $Func_i$ is the bilinear interpolate function, ReLU is the rectified linear unit, n is the number of heads in the last layer, and $\odot$ is the element-wise operation. We further employ three quantitative metrics including Accuracy Drop, Average Drop of Confidence, and Increase in Confidence to evaluate the model interpretability between our model and the ViT model [51]. Specifically, the visualization results generated from the proposed Instruction-ViT and ViT are input into CLIP as the discriminative model, and the evaluation metrics based on CLIP's output is calculated. The Accuracy Drop is defined as:

$$metric = \sum_{i=1}^{N}(ACC_{Orig} - ACC_{Vis})/N \tag{15}$$

where ACC_{Orig} and ACC_{Vis} represent the accuracy of original input and visualized input, respectively, and N represents the Validate set size. The Confidence Drop is defined as:

$$metric = 100 * (\sum_{i=1}^{N}(max(0, O_i - V_i)/O_i)) \tag{16}$$

where O_i and V_i represent the output of original input and visualized input, respectively. The Increase in Confidence is defined as:

$$metric = 100 * (\sum_{i=1}^{N} logic(O_i < V_i)/N) \tag{17}$$

where $logic(O_i < V_i)$ is to set to 1 if the input O_i and V_i meet the condition $O_i < V_i$.

4. Experiments

4.1. Datasets

We use 4 image classification datasets from different domains including Caltech-101 [52], Oxford-III Pets [53], AgriNet [54] and Oxford Flowers 102 [55]. We also use the data from PASCAL VOC2012 [56], Microsoft COCO 2014 [57] and Microsoft COCO 2017 [57] for the segmentation, image captioning, and object detection tasks, respectively.

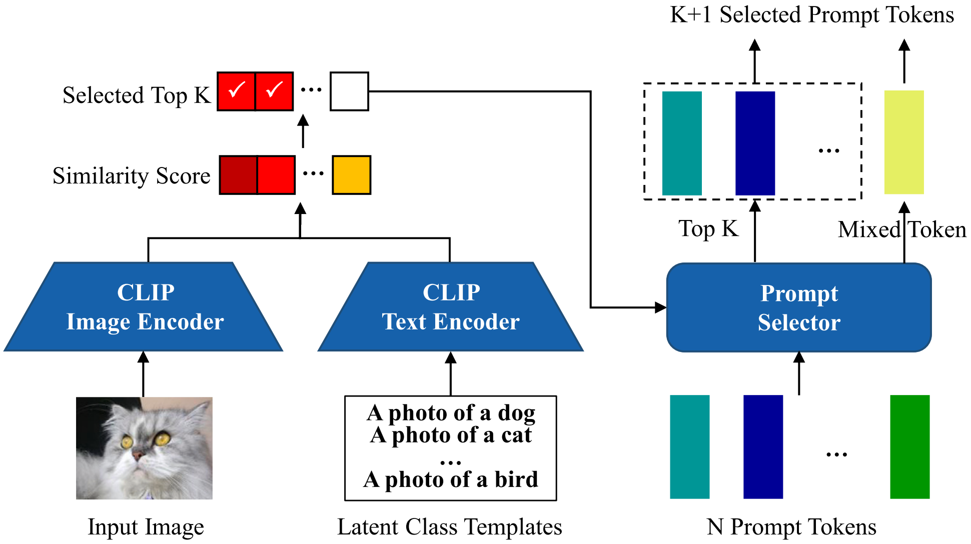

Fig. 3. The overall framework of prompt selection in validation studies. For an input image from the validation set, feature extraction is performed using the zero-shot CLIP model for the potentially possible class and the image. Then, a similarity score is calculated, in which the K prompt tokens with the highest similarity, along with the average value of the remaining N-K prompt tokens, are selected for the next module.

4.2. Experiments settings

4.2.1. Model setup

In our work, we follow the previous study [45] and adopt a network architecture consisting of 12-layer Transformer blocks with 768 hidden sizes and 12 attention heads. Additionally, the input image is partitioned into 196 patches with a patch size of 16. For the DeiT backbone experiment [46], we use the same network structure as ViT with a distillation head. For the creation of prompt tokens, we adopt the image and text encoders from the pre-trained parameters of CLIP [16] and a text encoder from BERT model [30]. In the segmentation, image captioning, and object detection tasks, we employ our Instruction-ViT as the backbone and leverage distinct decoders for each specific task. Specifically, the segmentation task is facilitated by a two-layer CNN, the image captioning task utilizes an LSTM [47], and the object detection task follows established methodologies as described in prior research [48].

4.2.2. Training setup

During the training stage in classification, the model is trained with a batch size of 256 and 20 epochs using a server equipped with 8 NVIDIA GeForce RTX 3090 GPUs. We use the Adam optimizer [58] with a learning rate of 1e-4. The foot learning rate is set to 1e-5, incorporating a linear warm-up over the first 5 epochs as part of the cosine decay strategy. For data augmentation, we adopt the RandAugment [59] and Mixup methods [60]. For few-shot learning, the maximum epoch is set to 100 for 2/4/8/16 shots and 60 for 1 shot by a single NVIDIA 3090 GPU, while other parameters remain the same. For the image captioning and object detection tasks, we fine-tune the models for 20 epochs and 10 epochs with a single NVIDIA 3090 GPU, respectively.

4.3. Result

4.3.1. Fine-tuning result in classification

As reported in Table 1, we compare our model with other models in the classification task, including ViT [45], DeiT [46], CaiT [61], PiT [62], ResNet [63] and EfficientNet [64]. The comparisons of model parameter size, GFLOPs, and FPS for different models are shown in Fig. 4. Additionally, we conduct an ablation study to compare the performance of our approach when utilizing the DeiT as backbone [46] and replacing prompts generated by BERT [30]. As a general observation, the average accuracy of our proposed model outperforms other models including both ViT-based models and CNN-based models in fine-tuning performance. Experimental results demonstrate the ability of our method to improve the classification accuracy by introducing features from the text. In addition, our method with ViT and DeiT as backbone achieves better results when using CLIP as the prompt generator compared to ViT and DeiT. Our method with BERT prompts yields a higher accuracy value compared to using the DeiT backbone, while the ViT backbone with selected CLIP prompts obtains the highest accuracy of 97.71%. The results demonstrate that the prompt embedding generated by the CLIP model, which specializes in aligning text and images, exhibits superior performance compared to the BERT model which focuses more on NLP. Moreover, employing DeiT as the backbone model performs better on the Oxford-III Pets dataset owing to the distillation token, while failing to outperform on the other datasets. In conclusion, the experimental results demonstrate the effectiveness and superiority of our method when fine-tuning the global parameters.

Table 1
Comparison of the fine-tuning classification performance of our model with other models in 4 datasets. The best values are highlighted in bold.

Model	Caltech101	Pets	AgriNet	Flowers	Average
ResNet-50	95.55	92.47	97.73	82.93	92.17
ResNet-101	96.72	93.18	97.91	85.76	93.39
ResNet-152	97.08	93.04	97.92	88.64	94.17
EfficientNet-B0	88.74	86.41	96.78	68.98	85.23
EfficientNet-B1	91.59	87.99	96.40	75.28	87.82
EfficientNet-B2	93.48	89.17	96.66	78.11	89.36
EfficientNet-B3	97.59	88.43	96.67	80.08	90.69
EfficientNet-B4	94.21	89.52	96.69	81.18	90.40
EfficientNet-B5	91.37	84.50	96.10	78.26	87.56
EfficientNet-B6	93.19	88.78	96.53	82.71	90.30
EfficientNet-B7	88.67	76.23	96.02	79.02	84.99
EfficientNet-B8	88.94	72.41	96.10	78.68	84.03
CaiT-S-24	96.56	94.27	98.55	96.43	96.45
PiT-B	96.73	95.29	98.98	97.25	97.06
ViT-B	97.61	94.19	99.06	**99.58**	97.61
DeiT-B	96.87	94.71	98.93	96.28	96.70
Ours(BERT prompt)	97.54	94.24	98.86	99.53	97.54
Ours(DeiT backbone)	96.96	**94.73**	98.88	96.52	96.77
Ours(CLIP prompt)	**97.68**	94.39	99.06	**99.58**	97.68
Ours(prompt select)	**97.68**	94.49	**99.07**	**99.58**	**97.71**

4.3.2. Visual prompt tuning result in classification

We additionally compare the model performance between our training method and the VPT method based on ViT [23]. We keep most of

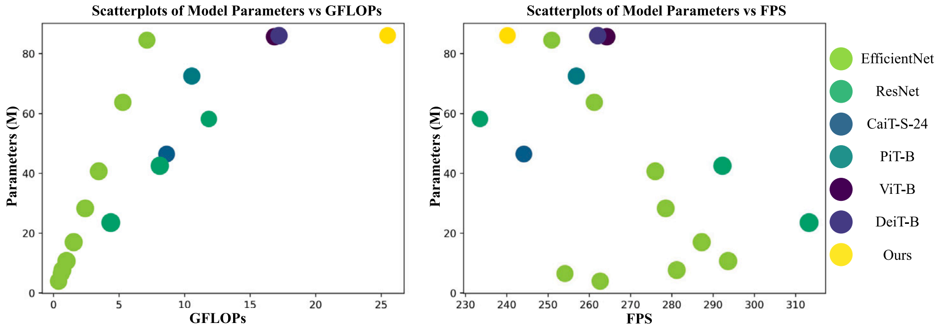

Fig. 4. Comparison of model size, GFLOPs, and FPS for different models.

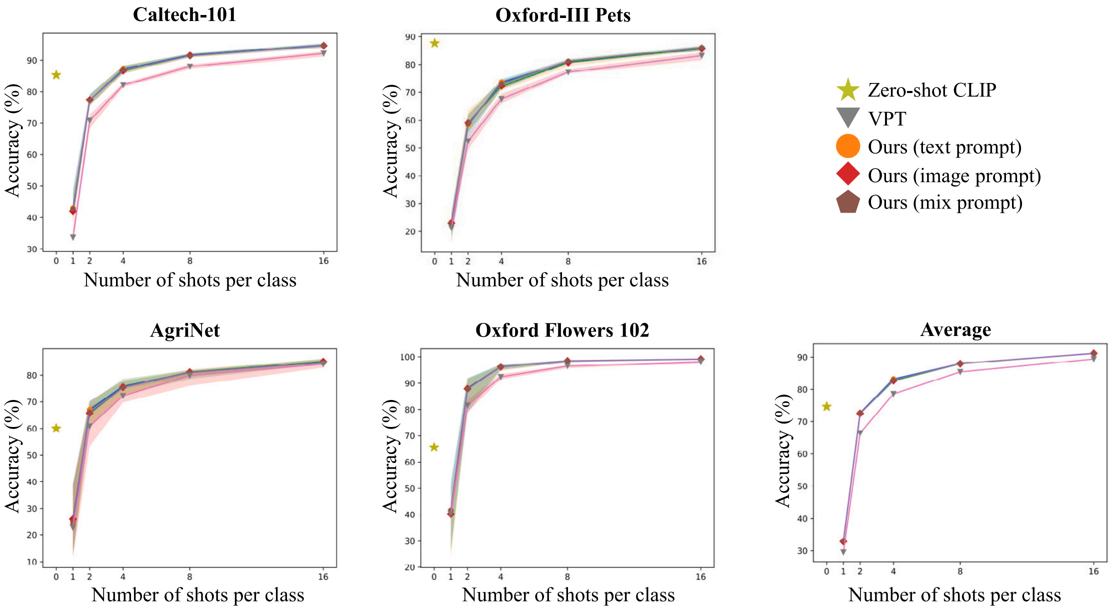

Fig. 5. The results of few-shot learning on the 4 datasets. Overall, our method effectively improves the accuracy compared with the VPT method, regardless of which prompt is used.

Table 2
Comparison of the fine-tuning classification performance between our method and ViT using the VPT training strategy in 4 datasets. The best values are highlighted in bold.

Model	Caltech101	Pets	AgriNet	Flowers	Average
VPT(ViT-B)	73.83	81.69	91.25	58.76	76.38
Ours (text prompt)	79.81	84.74	92.57	**63.20**	**80.08**
Ours (image prompt)	80.85	84.58	92.71	60.42	79.64
Ours (mix prompt)	79.78	84.39	92.56	61.10	79.46
Ours (Bert prompt)	**83.05**	84.77	**93.16**	55.53	79.13
Ours (DeiT backbone)	62.50	**90.75**	91.44	54.98	74.92

the model parameters frozen and only fine-tune part of the parameters. Specifically, we fine-tune the classification heads and the prompt embedding layer, while the VPT method fine-tunes the head and visual prompt. As reported in Table 2, our proposed method achieves superior performance compared to VPT. The experimental results demonstrate the feasibility of our proposed method in generating special prompts and also prove that the other modal prompts such as text prompts can improve the performance of visual tasks based on our method. Furthermore, we conduct a comparative analysis to assess the disparity among the three proposed modal prompts. In the four datasets of our experiments, the three modal prompts have their own advantages. In Oxford-III Pets and Oxford Flowers 102, the text prompt yields the highest accuracy of 84.74% and 63.20%, respectively. In the Caltech-101 and AgriNet datasets, the image prompt achieves the optimal accuracy of 80.85% and 92.71%. Although the highest classification accuracy is not achieved by using the mix prompt, the effect of the mix prompt which only represents the values of simple images and text prompts is better than that of merely using the image prompt in the AgriNet dataset, suggesting the importance of using multi-modal prompts in different scenarios. We also compare the model performance when using BERT output as prompts and using DeiT as the backbone. The results show that the method using the BERT output as prompt achieves superior performance than VPT as well as best performance on Caltech-101 and AgriNet datasets. In addition, the method using DeiT as the backbone achieves the highest accuracy on the Oxford-III Pets dataset, despite its underperformance on the other datasets. In conclusion, our method based on ViT backbone achieves superior performances compared to VPT when fine-tuning a small part of the parameters.

4.3.3. *Visual prompt tuning result in few-shot learning*

The overall performance of few-shot learning with only some fine-tuned parameters is shown in Fig. 5. We present a comparative analysis of the performance between our proposed method and the VPT method on four different datasets. Our method consistently outperforms the VPT method in terms of mean accuracy across individual datasets as

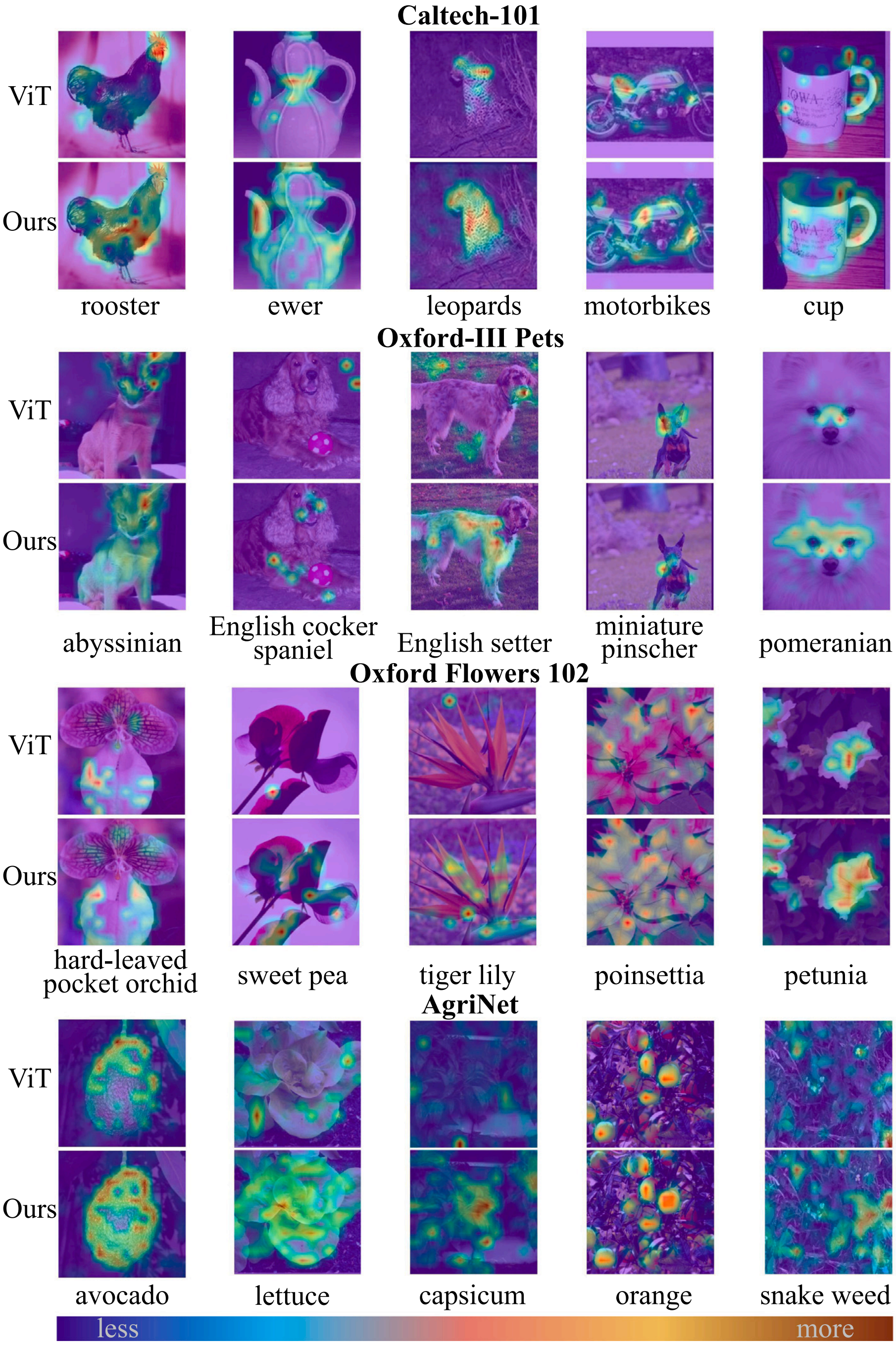

Fig. 6. Visual explanations generated by Grad-CAM. The visualization results for each of the five classes within a dataset are provided in ViT and Instruction-ViT, respectively. The areas where the model pays more attention are highlighted by warmer color, and vice versa.

well as the average results. In each dataset and each shot, we train our model five times with the randomly selected images and compare the performance of our method and VPT by the independent samples t-test. In the Caltech-101 dataset, our method exhibits significantly superior performance in the range of 1-shot to 16-shot compared to the VPT method. In both Oxford-III Pets and Oxford Flowers 102 datasets, our method shows a significantly superior performance over other methods from 2-shot to 16-shot. In the AgriNet dataset, our method shows superior mean performance compared to the VPT method. However, the significant results are only observed in the 4-shot, potentially due to the non-uniform distribution of the dataset. In summary, we demonstrate the effectiveness of our method in tackling the challenges of few-shot learning.

4.3.4. *Results of downstream tasks*

We conduct a comparative analysis between our method and the ViT model as the backbone for different tasks including segmentation, image captioning, and object detection. As reported in Table 3, our model achieves a higher mean Dice score of 56.12% and mean IoU of 45.41% in the segmentation task while ViT achieves a mean Dice score of 54.93% and mean IoU of 43.70%. The experimental results show that our proposed method can optimize the current ViT-based approach by introducing additional information in prompts. In the image captioning task, our method obtains 46.84% top 1 accuracy and 71.12% top 5 accuracy, which are higher than 46.34% and 70.56% of the ViT-based model, respectively. In the object detection task, our method and ViT-based method achieve approximate performance with

Table 3
Comparison of the fine-tuning performance in segmentation, image captioning, and object detection tasks. The best values are highlighted in bold.

Task	Segmentation		Image captioning		Object detection
metrics	mean Dice	mean IoU	TOP-1 ACC	TOP-5 ACC	bbox mAP
ViT-B	54.93	43.70	46.34	70.56	15.7
Ours	**56.12**	**45.41**	**46.84**	**71.12**	**16.2**

Table 4
Comparison of our method with the YOLO framework models in the object detection task. The mAP means the average precision (AP) at IoU ∈ $[0.5, 0.95]$, AP_{50} means the AP at IoU = 0.5, AP_{75} means the AP at IoU = 0.75. AP_{small}, AP_{medium} and AP_{large} mean the AP for small, medium, and large objects. The best values are highlighted in bold.

Model	mAP	AP_{50}	AP_{75}	AP_{small}	AP_{medium}	AP_{large}
YOLOv3	9.5	23.5	5.9	1.4	9.1	16.5
YOLOX-S	7.0	14.4	6.0	0.8	7.0	13.2
YOLOX-X	10.8	20.6	10.5	1.3	10.7	21.5
ViT	15.7	27.9	15.7	**2.5**	16.6	28.9
Ours	**16.2**	**28.8**	**16.4**	2.3	**17.1**	**29.1**

Table 5
Comparison of model interpretability of our model with ViT in each dataset.

Metric	Method	Caltech101	Pets	AgriNet	Flowers	Average
Accuracy drop↓	ViT	40.32	**65.49**	30.23	57.49	48.38
	ours	**27.86**	74.08	**28.79**	**57.08**	**46.95**
Average Drop of Confidence↓	ViT	47.03	**61.98**	**32.82**	51.79	48.40
	ours	**33.83**	68.43	32.95	**51.66**	**46.72**
Increase in Confidence↑	ViT	11.45	**3.95**	20.41	11.45	11.60
	ours	**19.78**	3.65	**22.45**	**19.78**	**14.20**

bbox mAP of 16.2% and 15.7%, respectively. We further compare the object detection ability between our model and representative YOLO framework models, i.e., the YOLOv3 [65] and the YOLOX [66], under the same training conditions. Our method consistently outperforms the two YOLO framework models as reported in Table 4. We also find that in our experimental conditions, i.e., using images with size of 224*224 as input and only 10 epochs of training, the overall performance of the model is different from the results reported in previous studies [48, 65,66]. It is worth noting that the improvement of our model does not intentionally focus on the key module of the object detection task, which may lead to comparable performances between ours and ViT-based model in the object detection task. Overall, our approach still exhibits superior performance in most of different downstream tasks.

4.3.5. *Interpretability analysis result*

We present visual explanations results in Fig. 6 using Grad-CAM. For each dataset, we display the visualization results from both ViT and Instruction-ViT across five different classes. We see that our model tends to focus on a broader area of the target subject, while ViT concentrates more on the subject's local information across most classes. In well-defined and common classes (e.g., classes in Caltech-101), Instruction-ViT can precisely focus more on the entire target subject, while in more fine-grained classification tasks such as classes in Oxford-III Pets, Instruction-ViT's attention centers more on local features. Quantitatively, as reported in Table 5, our model outperforms ViT in terms of the three metrics in Caltech-101 and Oxford Flowers 102 datasets. In the AgriNet dataset, our method performs better in the metrics of Increase in Confidence and Accuracy Drop, while ViT yields superior results in the Oxford Pet dataset. Overall, the average metrics of Instruction-ViT still surpass those of ViT in both qualitative and quantitative analyses.

5. Conclusion

In this work, we introduce Instruction-ViT, a simple and effective approach that aligns the input and prompts across distinct modalities. It leverages the pre-trained parameters from ViT-B as the backbone, and combines them with CLIP encoders as well as a flexible head module to complete various downstream tasks including image classification, segmentation, image captioning, and object detection. We show that Instruction-ViT can effectively use uni-modal prompts (e.g., images or texts) as well as multi-modal prompts (e.g., combined image and text features). Experimental results demonstrate that Instruction-ViT enhances the performance of the ViT-based model by incorporating prompts in different modalities, which can further improve the effectiveness of model with fewer parameter training requirements.

The current study has several limitations which can be categorized into two aspects: (1) Our work solely utilizes image and text data, neglecting continuous modal data such as audio and video. (2) The employed method to generate multi-modal prompts (text and image) relies on a basic weighted average of embedding generated by different modality encoders. Moving forward, we aim to refine the proposed Instruction-ViT from several perspectives of prompts. Considering the flexibility of our proposed prompt approach, we intend to further investigate how to design prompts that yield better results within our proposed framework. We plan to test different types of prompts such as using image descriptions as text prompts or employing prompts from other modalities like audio. Furthermore, we will investigate the optimal method for multi-modal prompt fusion in our framework.

CRediT authorship contribution statement

Zhenxiang Xiao: Methodology, Software, Visualization, Writing – original draft. **Yuzhong Chen:** Methodology, Software, Writing – original draft. **Junjie Yao:** Writing – original draft. **Lu Zhang:** Writing – original draft. **Zhengliang Liu:** Writing – review & editing. **Zihao Wu:** Writing – original draft. **Xiaowei Yu:** Writing – original draft. **Yi Pan:** Writing – review & editing. **Lin Zhao:** Writing – review & editing. **Chong Ma:** Writing – review & editing. **Xinyu Liu:** Writing review & editing. **Wei Liu:** Writing – review & editing. **Xiang Li:** Writing review & editing. **Yixuan Yuan:** Writing – review & editing. **Dinggang Shen:** Writing – review & editing. **Dajiang Zhu:** Writing – review & editing. **Dezhong Yao:** Funding acquisition, Writing – review & editing. **Tianming Liu:** Conceptualization, Project administration, Supervision, Writing – review & editing. **Xi Jiang:** Funding acquisition, Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This study was partly supported by the National Natural Science Foundation of China (61933003, 62276050, 61976045).

References

[1] L. Zhao, L. Zhang, Z. Wu, Y. Chen, H. Dai, X. Yu, Z. Liu, T. Zhang, X. Hu, X. Jiang, et al., When brain-inspired ai meets AGI, Meta-Radiology (2023) 100005.

[2] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, ACM Comput. Surv. 55 (9) (2023) 1–35.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, in: Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877–1901.

[4] OpenAI, Introducing chatgpt — openai.com, 2022, https://openai.com/blog/chatgpt. (Accessed 11 July 2023).

[5] Y. Zhang, K. Zhou, Z. Liu, What makes good examples for visual in-context learning? 2023, arXiv preprint arXiv:2301.13670.

[6] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A.C. Berg, W.-Y. Lo, et al., Segment anything, 2023, arXiv preprint arXiv:2304.02643.

[7] T. Baltrušaitis, C. Ahuja, L.-P. Morency, Multimodal machine learning: A survey and taxonomy, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2) (2018) 423–443.

[8] I.D. Gebru, S. Ba, X. Li, R. Horaud, Audio-visual speaker diarization based on spatiotemporal Bayesian fusion, IEEE Trans. Pattern Anal. Mach. Intell. 40 (5) (2017) 1086–1099.

[9] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.

[10] OpenAI, GPT-4 technical report, 2023, arXiv:2303.08774.

[11] J. Wang, Z. Liu, L. Zhao, Z. Wu, C. Ma, S. Yu, H. Dai, Q. Yang, Y. Liu, S. Zhang, et al., Review of large vision models and visual prompt engineering, 2023, arXiv preprint arXiv:2307.00855.

[12] A. Bar, Y. Gandelsman, T. Darrell, A. Globerson, A. Efros, Visual prompting via image inpainting, Adv. Neural Inf. Process. Syst. 35 (2022) 25005–25017.

[13] H. Dai, C. Ma, Z. Liu, Y. Li, P. Shu, X. Wei, L. Zhao, Z. Wu, D. Zhu, W. Liu, et al., SAMAug: Point prompt augmentation for Segment Anything Model, 2023, arXiv preprint arXiv:2307.01187.

[14] L. Zhang, Z. Liu, L. Zhang, Z. Wu, X. Yu, J. Holmes, H. Feng, H. Dai, X. Li, Q. Li, et al., Segment Anything Model (SAM) for radiation oncology, 2023, arXiv preprint arXiv:2306.11730.

[15] K. Zhou, J. Yang, C.C. Loy, Z. Liu, Learning to prompt for vision-language models, Int. J. Comput. Vis. 130 (9) (2022) 2337–2348.

[16] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.

[17] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, J. Liu, Uniter: Universal image-text representation learning, in: European Conference on Computer Vision, Springer, 2020, pp. 104–120.

[18] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, T. Duerig, Scaling up visual and vision-language representation learning with noisy text supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 4904–4916.

[19] J. Li, D. Li, C. Xiong, S. Hoi, BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022, arXiv preprint arXiv:2201.12086.

[20] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al., Flamingo: A visual language model for few-shot learning, Adv. Neural Inf. Process. Syst. 35 (2022) 23716–23736.

[21] L. Floridi, M. Chiriatti, GPT-3: Its nature, scope, limits, and consequences, Minds Mach. 30 (2020) 681–694.

[22] Y. Gu, X. Han, Z. Liu, M. Huang, Ppt: Pre-trained prompt tuning for few-shot learning, 2021, arXiv preprint arXiv:2109.04332.

[23] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, S.-N. Lim, Visual prompt tuning, in: European Conference on Computer Vision, Springer, 2022, pp. 709–727.

[24] J. Zhu, S. Lai, X. Chen, D. Wang, H. Lu, Visual prompt multi-modal tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 9516–9526.

[25] M. Zhu, Y. Weng, B. Li, S. He, K. Liu, J. Zhao, Knowledge transfer with visual prompt in multi-modal dialogue understanding and generation, in: Proceedings of the First Workshop on Transcript Understanding, 2022, pp. 8–19.

[26] X. Song, L. Jing, D. Lin, Z. Zhao, H. Chen, L. Nie, V2P: Vision-to-prompt based multi-modal product summary generation, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 992–1001.

[27] J. Zhang, S. Fang, Z. Mao, Z. Zhang, Y. Zhang, Fine-tuning with multi-modal entity prompts for news image captioning, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 4365–4373.

[28] K. Zhou, J. Yang, C.C. Loy, Z. Liu, Conditional prompt learning for vision-language models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16816–16825.

[29] J. Wei, M. Bosma, V.Y. Zhao, K. Guu, A.W. Yu, B. Lester, N. Du, A.M. Dai, Q.V. Le, Finetuned language models are zero-shot learners, 2021, arXiv preprint arXiv:2109.01652.

[30] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[31] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, Adv. Neural Inf. Process. Syst. 35 (2022) 27730–27744.

[32] P.F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, D. Amodei, Deep reinforcement learning from human preferences, in: Advances in Neural Information Processing Systems, vol. 30, 2017.

[33] N. Pan, K. Qin, A. Shekara, M.P. DelBello, ChatGPT: A promising AI technology for psychoradiology research and practice, Psychoradiology 3 (2023) kkad018.

[34] X. Jiang, X.-J. Shou, Z. Zhao, Y. Chen, F.-C. Meng, J. Le, T.-J. Song, X.-J. Xu, W. Guo, X. Ke, et al., A brain structural connectivity biomarker for autism spectrum disorder diagnosis in early childhood, Psychoradiology 3 (2023) kkad005.

[35] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N.A. Smith, D. Khashabi, H. Hajishirzi, Self-instruct: Aligning language model with self generated instructions, 2022, arXiv preprint arXiv:2212.10560.

[36] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, 2023, arXiv preprint arXiv:2302.13971.

[37] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T.B. Hashimoto, Stanford alpaca: An instruction-following LLaMA model, 2023, GitHub repository, GitHub, https://github.com/tatsu-lab/stanford_alpaca.

[38] H. Liu, C. Li, Q. Wu, Y.J. Lee, Visual instruction tuning, 2023, arXiv preprint arXiv:2304.08485.

[39] B. Peng, C. Li, P. He, M. Galley, J. Gao, Instruction tuning with GPT-4, 2023, arXiv preprint arXiv:2304.03277.

[40] Z. Yang, L. Li, K. Lin, J. Wang, C.-C. Lin, Z. Liu, L. Wang, The dawn of LMMS: Preliminary explorations with GPT-4V (ISION), 2023, p. 1, arXiv preprint arXiv:2309.17421, 9.

[41] Z. Liu, H. Jiang, T. Zhong, Z. Wu, C. Ma, Y. Li, X. Yu, Y. Zhang, Y. Pan, P. Shu, Y. Lyu, L. Zhang, J. Yao, P. Dong, C. Cao, Z. Xiao, J. Wang, H. Zhao, S. Xu, Y. Wei, J. Chen, H. Dai, P. Wang, H. He, Z. Wang, X. Wang, X. Zhang, L. Zhao, Y. Liu, K. Zhang, L. Yan, L. Sun, J. Liu, N. Qiang, B. Ge, X. Cai, S. Zhao, X. Hu, Y. Yuan, G. Li, S. Zhang, X. Zhang, X. Jiang, T. Zhang, D. Shen, Q. Li, W. Liu, X. Li, D. Zhu, T. Liu, Holistic evaluation of GPT-4V for biomedical imaging, 2023, arXiv:2312.05256.

[42] H. Liu, C. Li, Y. Li, Y.J. Lee, Improved baselines with visual instruction tuning, 2023, arXiv preprint arXiv:2310.03744.

[43] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al., Laion-5b: An open large-scale dataset for training next generation image-text models, 2022, arXiv preprint arXiv:2210.08402.

[44] A. Jangra, S. Mukherjee, A. Jatowt, S. Saha, M. Hasanuzzaman, A survey on multi-modal summarization, ACM Comput. Surv. 55 (13s) (2023) 1–36.

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[46] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: International Conference on Machine Learning, PMLR, 2021, pp. 10347–10357.

[47] M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli, G. Fiameni, R. Cucchiara, From show to tell: A survey on deep learning-based image captioning, IEEE Trans. Pattern Anal. Mach. Intell. 45 (1) (2022) 539–559.

[48] Y. Li, H. Mao, R. Girshick, K. He, Exploring plain vision transformer backbones for object detection, in: European Conference on Computer Vision, Springer, 2022, pp. 280–296.

[49] T. Wang, P. Isola, Understanding contrastive representation learning through alignment and uniformity on the hypersphere, in: International Conference on Machine Learning, PMLR, 2020, pp. 9929–9939.

[50] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[51] A. Chattopadhay, A. Sarkar, P. Howlader, V.N. Balasubramanian, Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks, in: 2018 IEEE Winter Conference on Applications of Computer Vision, WACV, IEEE, 2018, pp. 839–847.

[52] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories, in: 2004 Conference on Computer Vision and Pattern Recognition Workshop, 2004, p. 178, http://dx.doi.org/10.1109/CVPR.2004.383.

[53] O.M. Parkhi, A. Vedaldi, A. Zisserman, C. Jawahar, Cats and dogs, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3498–3505.

[54] H. Dai, L. Zhang, L. Zhao, Z. Wu, Z. Liu, D. Liu, X. Yu, Y. Lyu, C. Li, N. Liu, et al., Hierarchical semantic tree concept whitening for interpretable image classification, 2023, arXiv preprint arXiv:2307.04343.

[55] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, 2008, pp. 722–729, http://dx.doi.org/10.1109/ICVGIP.2008.47.

[56] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2010) 303–338.

[57] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.

[58] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[59] E.D. Cubuk, B. Zoph, J. Shlens, Q.V. Le, RandAugment: Practical data augmentation with no separate search, 2019, arXiv preprint arXiv:1909.13719.

[60] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, 2017, arXiv preprint arXiv:1710.09412.

[61] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, H. Jégou, Going deeper with image transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 32–42.

[62] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, S.J. Oh, Rethinking spatial dimensions of vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 11936–11945.

[63] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[64] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.

[65] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, 2018, arXiv preprint arXiv:1804.02767.

[66] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, YOLOX: Exceeding YOLO series in 2021, 2021, arXiv preprint arXiv:2107.08430.