Conditional Prompt Tuning for Multimodal Fusion

Ruixiang Jiang, Lingbo Liu, Changwen Chen
The Hong Kong Polytechnic University
Hong Kong SAR, China
rui-x.jiang@connect.polyu.hk {lingbo.liu, changwen.chen}@polyu.edu.hk

Abstract

We show that the representation of one modality can effectively guide the prompting of another modality for parameter-efficient multimodal fusion. Specifically, we first encode one modality and use its representation as a prior to conditionally prompt all frozen layers of the other modality. This is achieved by disentangling the vanilla prompt vectors into three types of specialized prompts that adaptively capture global-level and instance-level features. To better produce the instance-wise prompt, we introduce the mixture of prompt experts (MoPE) to dynamically route each instance to the most suitable prompt experts for encoding. We further study a regularization term to avoid degenerated prompt expert routing. Thanks to our design, our method can effectively transfer the pretrained knowledge in unimodal encoders for downstream multimodal tasks. Compared with vanilla prompting, we show that our MoPE-based conditional prompting is more expressive, thereby scales better with training data and the total number of prompts. We also demonstrate that our prompt tuning is architecture-agnostic, thereby offering high modularity. Extensive experiments over three multimodal datasets demonstrate state-of-the-art results, matching or surpassing the performance achieved through fine-tuning, while only necessitating 0.7% of the trainable parameters. Code will be released.

1. Introduction

Empowered with billion-scale training data and highly scalable model architectures, recent unimodal pre-trained large models [5, 6, 15, 24, 29], also known as foundation models, have demonstrated their powerfulness that are transferrable to various downstream tasks [22, 26, 46]. Transferring multimodally-pretrained foundation models for a specific multimodal task, however, is less flexible. Recent explorations, such as CLIP [30], employ two tower designs to contrastively pre-train two encoders together. Despite its success, the joint pre-training entangles the two encoders, meaning that replacing either one would necessitate expensive paired pre-training from scratch. This limitation restricts the broader application of multimodal foundation models for downstream tasks that would benefit from a specific unimodal architecture. Therefore, a compelling question arises: *Is it feasible to combine unimodal foundation models for downstream multimodal tasks?*

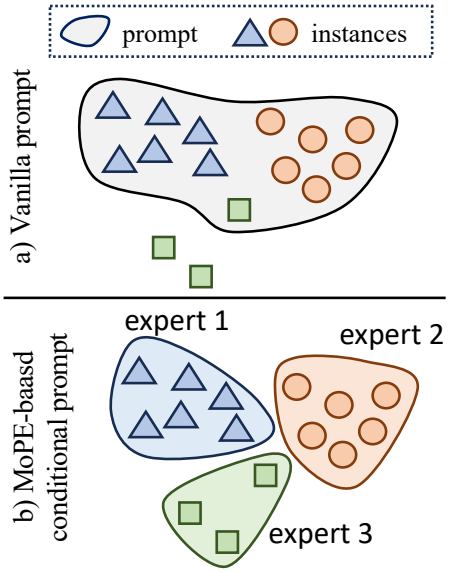

Figure 1. **High-level motivation of MoPE-based conditional prompting**. a) vanilla prompt-tuning learn a shared prompt for all instances, which would be influenced by majority classes; b) MoPE-based conditional prompt optimize multiple specialized prompt experts to better handle per-instance shift.

Two challenges emerge when transferring unimodal foundation models for multimodal task. The first challenge stems from the specialized architecture of foundation models across different modalities, each incorporating unique, modality-specific designs. This diversity complicates the design of an optimal fusion method for each combination of model architectures. Secondly, compared to unimodal datasets, the limited availability of multimodal data complicates the tuning of foundation models. As pre-trained

models continue to increase in size, the standard fine-tuning technique may not be as efficient, given the large volumes of data needed to achieve optimal performance.

We address above challenges by proposing a conditional prompt-tuning method. Specifically, we adopt a sequential pipeline to segregate the architecture details of each modalities. This design is architecture-agnostic, resulting in high modularity. To enhance the parameter-efficiency, we develop a fusion method based on the recently emerged prompt-tuning technique [10, 18], where we condition prompting of one modality on the other. Compared with fine-tuning, our method achieves similar or superior results while using significantly fewer trainable parameters.

Despite the effectiveness of prompt-tuning in various transfer learning setups, its application to multimodal tasks with large datasets is far from straightforward. Notably, prompt-tuning generally performs well in low-shot scenarios but can be less effective in full-shot training on the entire dataset [8, 20, 33, 38]. This reduced efficacy in large-scale scenarios could be explained by at least two factors: 1) Previous prompt-tuning methods optimize a globally-shared prompt for all instances [10, 18], which is not necessarily the local optimal for each input instance, and 2) the small amount of trainable parameters (compared with finetuning) can lead to underfitting on large datasets. Increasing the prompt length is a potential solution to incorporate more trainable parameters and better capture minority classes, but it often results in degraded outcomes instead [10, 12, 14, 18, 38]. In this paper, we aim to address these challenges by developing an instance-specific prompting technique and enhancing the expressiveness of vanilla prompt tuning.

To incorporate instance-specific prior into the framework of prompt-tuning, we augment the vanilla global-shared (*i.e.*, static) prompt with instance-wise (*i.e.*, dynamic) information to better handle per-instance shift. Specifically, we introduce two additional types of prompts: the *dynamic prompt* augments the static prompt by capturing instance-level shift, while the *mapped prompt* inject fine-grained information for multimodal prediction. We generate the two prompts in an instance-wise manner, utilizing two encoders for each modality in a sequential pipeline, where the output of the complementary modality is used to guide the prompting of the main modality encoder. Such design segregates the architectural details of encoders from each other, thereby permitting each modality to be independently substituted.

To further scale up the expressiveness of prompting, our key motivation is to fit multiple prompts to deal with the shift across instances, as illustrated in Fig. 1. To be more specific, we introduce the **M**ixture **o**f **P**rompt **E**xpert (MoPE) method for dynamic prompt generation, where a pool of prompt experts is optimized per-layer. For a specific instance, we utilize a learned router to predict a routing score, which is used to weigh all experts for synthesizing the dynamic prompt for this specific instance. We also study the effect of a regularization term, which prevents degenerated routing across instances. With the combination of those modules, we observe specialized experts spontaneously emerge after training, each focusing on a specific group of instances. Moreover, we demonstrate that increasing the number of experts could be an effective way to scale up the model capacity compared with simply increasing prompt length. This allows for learning more complex mapping for down-stream tasks with a large dataset, while still keeping fixed sequence length during self-attention. In conclusion, we summarize our contribution as follows:

- We propose to augment the vanilla prompt tuning with the dynamic and mapped prompt, utilizing the paired modality as a prior, to better adapt the pretrained model to each instance.
- We elaborate a mixture of prompt expert design for dynamic prompt, which scales up the expressiveness of prompt tuning for transfer-learning.
- We study the effect of a regularization term for avoiding degenerated expert routing.
- Extensive experiments on UPMC_Food-101, SNLI-VE and MM-IMDB dataset demonstrate state-of-the-art performance for multimodal fusion.

2. Related Works

Unimodal Prompt Tuning. Prompt tuning emerges as a parameter-efficient method of transferring large pretrained models for down-stream tasks. This involves freezing the pretrained model and inserting additional learnable prompt tokens into the input of the model for solving specific tasks. This tuning scheme was first popularized in natural language processing (NLP) [17, 18], then quickly introduced to computer vision (CV) society [3, 10, 21, 23]. For either modality, this tuning scheme achieves good transfer learning performance in low-data regime, yet its performance is less comparable to fine-tuning when abundant training instances are available [8, 18]. Moreover, simply increasing prompt length could result in performance saturation, and over-length prompts might have an negative impact on performance [10, 14, 38]. Following works scale up the vanilla prompt tuning by increasing their diversity through chain-of-thought [43], adaptive prompting [2, 9, 36, 39], or ensembling [28]. In this work, we scale up the expressiveness of vanilla prompting with a MoE design.

Multimodal Prompt Tuning. There are two prominent streams of work in multimodal prompting, depending on whether pre-training of the foundation model is multimodal or unimodal. The first stream applies prompt tuning to transfer *multimodal pretrained models* for down-

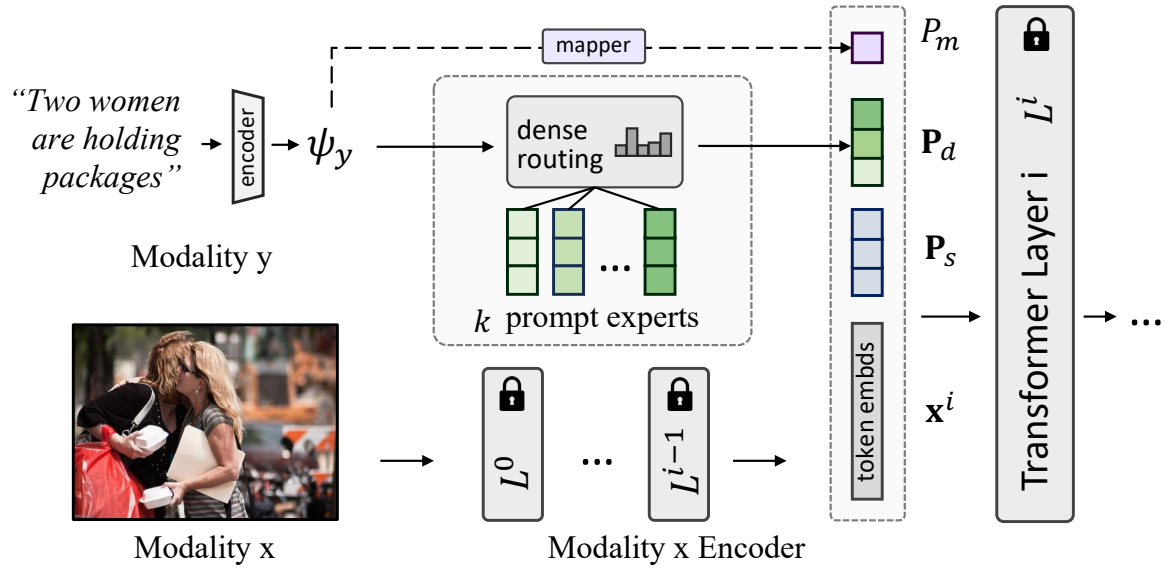

Figure 2. **Overview of instance-wise conditional prompting.** We illustrate the proposed conditional prompt tuning method for multimodal fusion, applied to one Transformer layer. Three types of prompts are concatenated to the token embeddings: (1) We first extract features from the complimentary modality ψ_y, which is used to guide the mixture of expert routing of instance-wise dynamic prompt $\mathbf{P}_d$. (2) A lightweight mapper is applied to map ψ_y into a single prompt P_m. (3) the static prompt $\mathbf{P}_s$ will also be used, which is not conditional on ψ_y. All of the pretrained Transformer parameters are frozen, and only the router, mapper and prompt are trainable.

stream tasks, primarily focusing on Vision-Language Models (VLMs) such as CLIP [30]. Existing methods of this stream either optimize prompts on one branch (*e.g.*, CoOp [45], CPT [41]), or tune both branches with different designs [11, 12, 42, 44]. More related to our work is the second stream, which focus on using prompt to bridge *unimodal pretrained models* for multimodal task. Frozen [33] first introduces a method where the visual representation are mapped as two input tokens to query frozen language models (LM). PromptFuse and BlindPrompt [20] improved upon this by introducing tunable prompts to the LM only for alignment. PICa [40] translate images into text captions to prompt frozen Large LMs. PMF [19] further introduces interactive prompt-tuning only in the deep layers for memory efficiency. Our work extends the second branch by further studying the interplay of prompting across modalities.

MoE in Transformers. MoE is a powerful technique for scaling Transformers up to billion scale [16, 27, 31]. The fundamental approach involves inserting MoE layers into the standard Transformer architecture, composed of multiple feed-forward networks (FFNs) acting as experts. A router is learned to route each token embedding to the most suitable expert(s). Due to the significant computational cost associated with FFN forwarding, a sparse gating function is typically employed to limit the number of experts used per token [28, 31]. Thanks to those designs, the MoE layer can efficiently scale up model capacity. Inspired by previous FFN-based MoE, we explore scaling up the prompt tuning method with an MoE design.

3. Method

In this section, we elaborate on the proposed method for multimodal fusion via conditional prompting. In Sec. 3.1 we establish the basic notation of prompt tuning. We explain how we sequentially fuse two unimodal encoders via prompting, as well as the three types of prompts in Sec. 3.2. In Sec. 3.3 we introduce the MoPE-based conditional prompting technique. Finally, we introduce the regularization method in Sec. 3.4.

3.1. Preliminary: Prompt Tuning

Consider using a Transformer [34] or its variants to extract features from (embedded) input sequence $\mathbf{x}^0 \in \mathbb{R}_{s\times d}$ where s is the sequence length and d is hidden dimension of the Transformer. The input sequence of i-th layer L^i could be further denoted as:

$$\mathbf{x}^i = [x_0^i, \mathbf{T}^{i-1}] \tag{1}$$

where x_0^i denote the [CLS] token, $\mathbf{T}^{i-1}$ is the token embedding from the previous layer, and $[,]$ denotes the concatenation operation.

Instead of fine-tuning all of the parameters, prompt-tuning freezes all pre-trained model weights and insert a small amount of trainable prompts $\mathbf{P} \in \mathbb{R}_{l\times d}$ to the input

Figure 3. **Examples of expert routing.** We visualize the result of last-layer expert routing on the SNLI-VE test set, by showing the images (left) and the paired word clouds (right) routed to expert 4 and 12. The expert ID of an instance is determined by its highest routing score for visualization purposes only. In this example, expert-4 is specialized for children while expert-12 focuses on crowds.

$\mathbf{x}^0$ or internal layers $\mathbf{x}^i$, where l is the number of prompts. The input of layer L^i now becomes:

$$\hat{\mathbf{x}}^i = [x_0^i, \mathbf{P}, \mathbf{T}^{i-1}] \tag{2}$$

In the self-attention process, the learned prompt $\mathbf{P}$ attends to the whole sequence $\hat{\mathbf{x}}^i$ to achieve parameter-efficient transfer learning. Unless otherwise specified, we assume that prompt-tuning append learnable prompts to all of the transformer layers, this corresponds to "VPT-deep" in CV and "prefix-tuning" in NLP.

3.2. Conditional Prompt Tuning

Our objective is to transfer unimodal pretrained models for solving multimodal tasks via prompt tuning. With paired multimodal input data as a prior, we aim to condition the prompting of one modality on the other for more effective and adaptive transfer learning. To achieve this goal, we disentangle the vanilla prompt vector $\mathbf{P}$ into three types of specialized prompts $[\mathbf{P}_s, \mathbf{P}_d, P_m]$. The static prompt $\mathbf{P}_s \in \mathbb{R}_{l\times d}$ is a globally-shared prompt vector that is non-conditional to input. The dynamic prompt $\mathbf{P}_d \in \mathbb{R}_{l\times d}$ and the mapped prompt $P_m \in \mathbb{R}_d$ are instance-wise adaptive, which are conditionally synthesized based on the representation of the other modality(ies).

A sequential pipeline is adopted to achieve prompt conditioning. More specifically, let (x, y) be a multimodal input pair, with $\mathcal{E}_x$, $\mathcal{E}_y$ be the encoders of each modality. Depending on the intrinsic of the specific task, we assign a fusion direction from the complementary modality to the main modality, and the proposed prompt tuning method will only be used for tuning the main modality. Without loss of generality, we treat x as the main modality, so the y would be complementary. To synthesize the instance-wise prompt vector $\mathbf{P}_d$ and P_m, we first extract the global-level feature of complementary modality $\psi_y = \mathcal{E}_y(y)$ and apply a router $R(\cdot)$ to synthesize the instance-wise dynamic prompt vectors $\mathbf{P}_d$ to tune $\mathcal{E}_x$ (entailed in Sec 3.3). To further inject information from the complimentary modality, we apply a lightweight mapper to map the complimentary feature as a single prompt: $P_m = f_m(\psi_y)$. Finally, the static prompt $\mathbf{P}_s$ will also be used to tune the main modality encoder $\mathcal{E}_x$. In summary, the input of layer L^i of $\mathcal{E}_x$ becomes:

$$\hat{\mathbf{x}}^i = [x_0^i, \mathbf{P}_s, R(\psi_y), f_m(\psi_y), \mathbf{T}^{i-1}] \tag{3}$$

It is important to note that our three types of prompting serve as a plug-in module to replace the vanilla prompt vector of the main modality $\mathcal{E}_x$. We do not impose any constraints on the architecture, capacity, or tuning method of the complementary modality $\mathcal{E}_y$.

3.3. Mixture of Prompt Experts

We propose learning multiple prompts as experts for generating the dynamic prompt in a per-instance manner to handle per-instance shift. Specifically, For each prompt-tuned Transformer layer L^i, we randomly initialize k prompt experts $\{\mathbf{E}_i\}_{i=1}^k$ to be optimized end-to-end, where $\mathbf{E}_i \in \mathbb{R}_{l\times d}$. For a specific instance, its dynamic prompt at this layer would be synthesized based on all of the available experts. This is accomplished by learning a $\mathrm{Softmax}$ layer to predict a routing score $r = \mathrm{Softmax}(\mathbf{W}_r\psi_y/\tau + \epsilon)$, where $\mathbf{W}_r$ is layer-specific linear transformation, $\tau = 0.1$ is the temperature hyper-parameter, and ϵ is sampled noise to encourage routing diversity [31]. The routing process $R(\psi_y)$ is to synthesize the dynamic prompt by a convex combination of all experts according to the routing score:

$$\mathbf{P}_d = \sum_i^k r_i \mathbf{E}_i \tag{4}$$

Unlike previous MLP-based MoE methods [31], we do

not insist the routing score be sparse (*i.e.*, we do not use a TOP-K gate). This is because all prompts are the leaf nodes in the computation graph, and densely fusing them does not result in high computational cost, while we empirically find it gives better results.

3.4. Regularizing Expert Routing

To avoid specific experts being dominant across all instances, we add an additional importance loss [31] to encourage balanced expert routing. Specifically, for a batch of input $\mathbf{Y}$, the importance of expert-i is defined as the summed routing score of all inputs.

$$\mathrm{Imp}(E_i) = \sum_{y \in \mathbf{Y}} \mathrm{Softmax}(\mathbf{W}_r \psi_y / \tau)_i \tag{5}$$

The importance loss is defined as coefficient of variation of all experts' importance:

$$\mathcal{L}_{imp} = \mathrm{stopgrad}\left(\left(\frac{\mathrm{std}(\{\mathrm{Imp}(\mathbf{E}_i)\}_i^k)}{\mathrm{mean}(\{\mathrm{Imp}(\mathbf{E}_i)\}_i^k)}\right)^2 ; \gamma\right) \tag{6}$$

where $\mathrm{stopgrad}(\cdot)$ is the stop-gradient operator, which prevents error propagation of this loss term when the coefficient of variation is less than a pre-defined threshold $\gamma = 0.05$. This loss encourages balanced utilization of experts across all instances. The inclusion of the additional threshold constraint is due to our instance-wise routing. Unlike the previous per-token routing, instance-wise routing uses a smaller batch size for importance value calculation, thereby increasing the likelihood of a larger coefficient of variation.

4. Experiment and Results

4.1. Implementation Details

Architecture Details. For all experiments, unless otherwise specified, we use Swin-B [24] as the vision encoder and Bert-base [5] as the text encoder. Following the experiment setup in [10, 19], we also finetune a linear head for each dataset. We implement the mapper $f_m(\cdot)$ as a two-layer MLP with GELU nonlinearily. Regarding prompt tuning, we use $l = 6$ prompts and $k = 16$ experts by default, and the tunable prompt is applied to all layers of the main modality encoder. We employ vanilla prompt tuning [10] to tune the complementary modality.

Training Detail. All images are resized and cropped to the size of 224×224. All models are trained for 12 epoches, using the AdamW [25] optimizer with a learning rate of $4e-4$ for vision and $5e-4$ for text. We use RandAug [4] with default setup to augment the training images, with the exception of the SNLI-VE dataset. We use a constant step decay scheduler that halves the learning rate at epoch = 2 and 5.

4.2. Datasets

UPMC_Food101 [35] serves as a comprehensive multimodal dataset designed for fine-grained recipe classification. The dataset contains 90,840 image-text pairs for 101 food classes. We follow previous methods [13, 19] to create a validation set of 5000 samples.

MM-IMDB [1] is a multimodal movie classification dataset. It comprises 25,956 pairs of images and texts, each pair including a movie poster and a plot summary. The dataset supports multi-label genre classification across a spectrum of 23 genres with imbalanced classes.

SNLI-VE [37] is a large-scale multimodal dataset with 565,286 image-text pairs. The task for this dataset is visual entailment, which means that the model should decide whether a hypothesis matches the given premise. Following [19], we only use the image premise, while in other works the text premise might also be used.

4.3. Compared Methods

First, we compare our method with several baseline methods for multimodal fusion.

ImgOnly / TextOnly. Only fine-tune one encoder, and the input of the other modality is discarded.

P-ImgOnly / P-TextOnly. Only prompt-tune one encoder, and the input of the other modality is discarded. In our experiment, we use VPT-deep [10] style prompt-tuning for the vision and text transformer.

LateConcat. This baseline involves fine-tuning both encoders, concatenating their features, and appending a linear head for classification.

P-LateConcat. Similar as **LateConcat** but prompt-tune each encoder instead of fine-tuning. However, the linear head is still fine-tuned.

SequentialFuse. This method first extracts features from the complementary modality and maps them to the embedding space of the main modality encoder in the same way as our mapped prompt for end-to-end training. Both encoders are fine-tuned. This is a strong baseline and can be seen as our method without MoPE and routers, but with all parameters fine-tuned.

P-SequentialFuse. Similar as **SequentialFuse** but prompt-tune each encoder. This baseline is comparable to CoCoOp [44] but with additional static prompts.

To ensure a fair comparison, all the above methods with prompt-tuning use the same prompt length as our method. In addition to these baselines, we also compare our methods with existing prompt-based fusion methods, including MMBT [13], Frozen [33], PromptFuse [20], BlindPrompt [20], and PMF [19].

4.4. Main Results

The quantitative results of all baselines, compared methods, and our method with different expert number k are summa-

	Method	Param	SNLI-VE ↑	Food-101 ↑	MM-IMDB ↑
finetuning	ImgOnly	86.9M	33.37	75.84	39.31/53.85
	TextOnly	109.0M	69.69	86.46	58.80/65.67
	LateConcat	196.0M	70.01	93.29	59.56/64.92
	SequentialFuse	197.0M	74.44	87.38	59.53/66.55
	MMBT [13]	196.5M	67.58	94.10	60.80/66.10
prompt-tuning	P-ImgOnly	0.1M	33.30	75.84	32.83/49.47
	P-TextOnly	-	69.69	81.20	51.84/61.81
	P-LateConcat	1.0M	63.05	89.03	53.91/59.93
	P-SequentialFuse	1.1M	65.26	81.50	55.57/63.98
	P-MMBT [19]	0.9M	67.58	81.07	52.95/59.30
	PromptFuse [20]	-	64.53	82.21	48.59/54.49
	BlindPrompt [20]	-	65.54	84.56	50.18/56.46
	PMF [19]	2.5M	71.92	91.51	58.77/64.51
	PMF-Large [19]	4.5M	72.10	91.68	61.66/66.58
	Ours ($k = 4$)	1.5M	73.32	91.54	61.54/67.49
	Ours ($k = 16$)	2.6M	**73.85**	**91.74**	**62.01/68.25**

Table 1. **Quantitative results on three multimodal dataset.** Quantitative result of all of the baseline methods, compared methods, and our method with different expert numbers. The metric on SNLI-VE and Food-101 is accuracy (%), and MM-IMDB is F1-Macro and F1-Micro. We also list the total number of trainable parameters (million) of each method, where '-' means parameter less than 0.1 million. The best prompt-based results are in bold.

Prompt	SNLI-VE ↑	Food-101 ↑	MM-IMDB ↑
$[\mathbf{P}_s]$	33.30	75.2	24.69/45.11
$[\mathbf{P}_d]$	64.52	74.22	46.90/60.17
$[P_m]$	33.94	72.70	24.69/45.11
$[\mathbf{P}_s, \mathbf{P}_d]$	66.92	74.24	48.26/60.96
$[\mathbf{P}_s, P_m]$	65.26	81.40	55.57/63.98
$[\mathbf{P}_d, P_m]$	71.81	91.13	60.42/66.88
$[\mathbf{P}_s, \mathbf{P}_d, P_m]$	**73.85**	**91.74**	**62.01/68.25**

Table 2. **Ablation on each of the prompt types.** The results from all combinations of mapped prompt, dynamic prompt, and static prompt are presented. Our full method with all prompts achieves the best result.

rized in Tab 1. We also report the total trainable parameters of all methods.

Our method outperforms all the listed prompt-based methods and is competitive with fine-tuning. Specifically, when compared with the fine-tuning baselines, SequentialFuse and LateConcat, our method delivers competitive accuracy on the Food-101 dataset and superior results on the SNLI-VE and MM-IMDB datasets, while requiring as few as 0.7% of the trainable parameters. Our method also matches the performance achieved by MMBT [13], an early fusion method, which suggests that the proposed sequential pipeline could be a promising fusion paradigm.

Compared to the prompt-tuning baseline, our method demonstrates superior parameter-efficiency. Our MoPE design enables us to achieve significant gains of 12.34%, 12.32% and 10.75%/5.48% respectively on all metrics, outperforming P-SequentialFuse. We also surpass all existing prompt-based fusion methods, including PromptFuse [20], BlindPrompt [20], and PMF [19]. Notably, our method with $k = 4$ delivers similar results to PMF-Large, while only requiring 33% of the trainable parameters, and our method with $k = 16$ establishes a new state-of-the-art on the three datasets.

4.5. Qualitative Result of MoPE Routing

Specialized prompt experts naturally emerge during training. We visualize two of them in Fig 3. Specifically, we manually designate the expert with the highest routing score as the expert for each instance for visualization. By doing so, we are able to collect all text that would be routed to this expert, as well as the paired images. For each expert, we create a word cloud of all the texts, with common stop words (*e.g.*,*"an" "the"*) removed. We also visualize some images that are paired with the text. In the provided example, we observe that expert-4 appears to specialize in children, while expert-12 focuses on crowds.

5. Analysis and Discussion

Ablation: Effect of each prompt types. We ablate each of the prompt types and report the metrics on three datasets, the result is summarized in Tab. 2. Our full method with all types of prompts achieves the best result, indicating that each prompt collaborates with the others. Specifically, with-

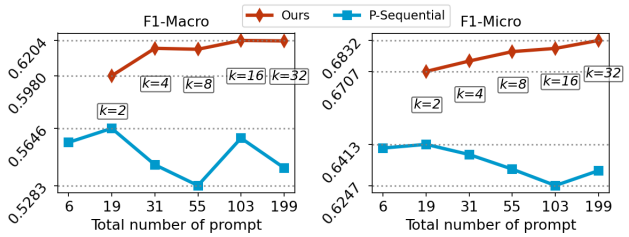

Figure 4. **More experts v.s. longer prompts.** We compare the effects of increasing the number of experts, k, versus increasing the prompt length, l. Increasing the number of experts consistently outperforms the strategy of lengthening the prompts, exhibiting a linear growth trend. Conversely, excessive prompt length detrimentally affect the model's performance

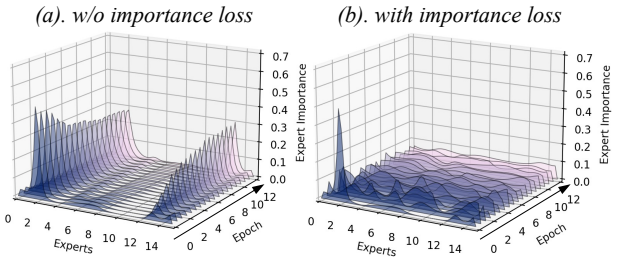

Figure 5. **Effect of the importance loss**. We visualize how importance (z-axis) of all experts (x-axis) in the last Transformer layer change during training (y-axis). (a) without importance loss, only a few experts are used throughout training (b) the importance loss ensures balanced utilization of all experts.

out the mapped prompt, the fine-grained information from complimentary modality is lost, leading to a significant drop of 10.3%, 23.6% and 28.4%/11.9% on all metrics, respectively. When the dynamic prompt is not used, the method simply find a global prompt, which may not handle every instance well. This is supported by a drop of 11.6% F1-Macro, which is known to be sensitive to class-imbalance. Without the static prompt, the model already achieves satisfactory result, due to the fact that dynamic prompts with a degenerated router could also act as static to learn feature across all instances. However, adding the static prompt relieves the stress of dynamic prompts for global feature modeling, allowing them to focus more on capturing instance-level features. This leads to an improvement of 2.84%, 0.67%, and 2.63%/2.04%, respectively.

Ablation: Effect of MoPE. We study the effectiveness of MoPE by increasing the number of experts k v.s. scaling up prompt length l. Specifically, our starting point is $l = 6$ prompts and $k = 2$ experts, which account for $(2 + 1) \times 6 + 1 = 19$ tunable prompts in total per-layer. We increase the number of k, and at each step we also report the result of simply increasing l to the same total number of prompts. The results, measured as F1-Macro and F1-Micro on the MM-IMDB dataset, are summarised in Fig. 4.

For both metrics, our results indicate that merely augmenting the number of prompts l does not lead to a steady performance improvement. Such observation is in-line with previous studies [10, 12, 18, 38]. Moreover, the increase in prompt length will slow down the model due to the quadratic time complexity of self-attention. On the other hand, our MoPE scales better with the total number of prompts, as indicated by the linear growth of performance with respect to total trainable prompts. This improvement is achieved by conditioning the dynamic prompt on more experts to enhance the prompt's expressiveness. Since the number of prompts used in the forward pass remains fixed, our method maintains a constant time complexity.

Ablation: Effect of the importance loss. The importance loss is crucial for avoiding degenerated routing solution. In Fig. 5, we visualize how the importance of each expert (*i.e.*, average routing score) changes when training on the SNLI-VE dataset. Without the importance loss, routing adheres to its initial state, leading to a skewed distribution where few experts dominate. This phenomenon aligns with observations from previous FFN-based MoE methods [7, 32]. The importance loss alleviates this by penalizing unbalanced expert importance, which results in balanced expert utilization of all experts.

Ablation: Dense routing vs. sparse routing. Previous MoE with FFN experts usually employ a sparse gating function that only select one or a few experts for token, while we use a dense routing to allow all experts to make contributions. To compare the effect, we also implement a sparse router following [31], which takes form:

$$r' = \text{TOP}_1(\text{Softmax}(\mathbf{W}_r \psi_y / \tau + \epsilon)) \tag{7}$$

The results of using sparse routing versus dense routing are reported in Tab. 3. Our experiments indicate that switching between either routing scheme does not induce significant performance gaps within the MoPE setting, while we favor the dense one as it gives marginally better results.

To further understand how dense routing helps the model, we evaluated the degree of uncertainty in routing, utilizing the empirical entropy of the routing score, averaged across training batches and all layers. Basically, high entropy indicate that routing behaves more randomly. The entropy is defined as $\mathcal{H}(r) = -\sum_{i=1}^{n} r_i log_2(r_i)$, where r corresponds to the raw SoftMax probability regardless of the application of the TOP-1 gate. The relationship between entropy and optimization steps is illustrated in Fig. 6. As demonstrated in the figure, dense routing results in less random expert decision, suggesting that the model better leverages the complementary modality for routing.

Scalability with dataset size. The performance of previous prompting method does not scale well with respect to increased training data [8, 20, 33, 38]. We study how the

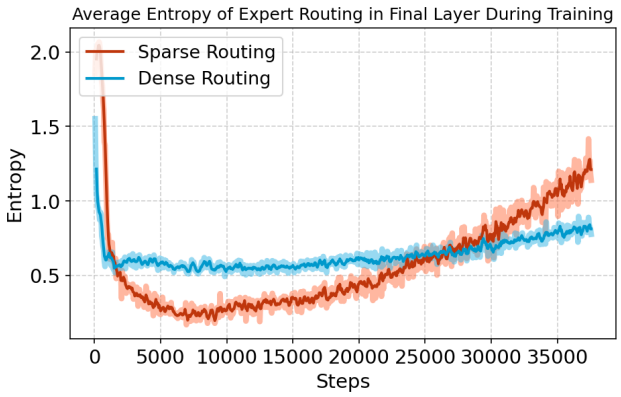

Figure 6. **Routing randomness of dense vs. sparse routing.** The average routing entropy throughout the optimization process for both dense and sparse routing is visualized. Dense routing exhibits less randomness

Routing	SNLI-VE ↑	Food-101 ↑	MM-IMDB ↑
Dense	**73.85**	**91.74**	**62.01/68.25**
Sparse	73.00	91.20	61.95/68.11

Table 3. **Result of dense routing v.s. sparse routing**. Dense routing achieves slightly better results on all three datasets.

proposed method performs on different data scales. Specifically, we sub-sample the training set to 64, 256, 512, and 1024 shots, and train our method and other prompt-tuning and fine-tuning methods with the same subsampled data. The mean accuracy and F1-Micro score on SNLI-VE and MM-IMDB, respectively, are reported in Fig. 7.

We observe that the two compared prompting methods, PromptFuse and P-SequentialFuse, provide satisfactory results in low-data situations. However, when a larger quantity of training samples is available, as is the case with 1024-shot and full-dataset training, they exhibit a significant performance gap compared to the fine-tuning method, SequentialFuse. In contrast, our method consistently matches or exceeds the results of fine-tuning across all data scales. We also note that our method significantly outperforms all compared methods on the low-shot MM-IMDB F1-micro metric. This superior performance is linked to the observation that the routing entropy is high in the low-shot regime. This suggests that our MoPE could adaptively behave like ensembling to prevent overfitting.

Modularity. The proposed fusion method abstracts the complimentary modality as a representation, allowing high modality of both modalities. In particular, our method allow arbitrary models to be seamless plug-in for multimodal fusion. Such modularity is at least three fold: model architecture, the pre-training scheme, and the specific transfer learning method of the model. We exemplify each in Tab. 4.

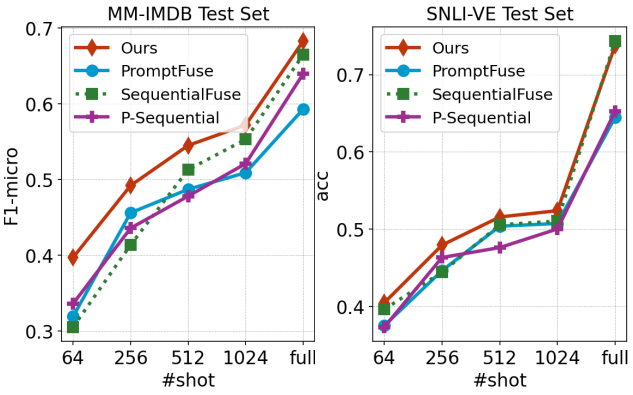

Figure 7. **Scaling performance with increased training data**. This chart showcases the comparative performance of our method and representative methods as we progressively increase the amount of training data, or 'shots'. The proposed method consistently achieves superior results, outperforming other prompt-tuning methods at all data scales, and remains competitive with the full-tuning method, *PromptFuse*. Note: *P-Sequential* means *P-SequentialFuse*.

Architecture	Pretraining	Transfer	MM-IMDB ↑
BoW	Bert*	Fine-tuning	48.20/57.50
Transformer	Bert [5]	Frozen	58.86/66.13
Transformer	GPT-2 [29]	Frozen	34.03/50.84
Transformer	Bert [5]	Fine-tuning	60.34/67.27

Table 4. **Our prompt tuning method are highly modular.** We offer flexibility in at least three dimensions: model architecture, pretraining scheme, as well as transfer learning technique. (*): Bag-of-word initialized with Bert word embeddings.

Limitation and future work. The proposed method relies on a single global-level representation for the complimentary modality, which might overlook the spatial information useful for specific tasks such as segmentation. Future works could study how to allow arbitrary sequence length from the complementary modality for fusion.

6. Conclusion

In this paper, we present a conditional prompting method for parameter-efficient multimodal fusion. Our method involves augmenting the vanilla static prompt with dynamic and mapped prompt for instance-adaptive prompt learning. We also introduce the mixture-of-prompt-expert technique to improve the expressiveness of prompt-tuning. Extensive experiments demonstrate our method is parameter-efficient, scales better with dataset size and number of prompts, and is highly modular.

References

[1] John Arevalo, Thamar Solorio, Manuel Montes-y Gómez, and Fabio A González. Gated multimodal units for information fusion. *arXiv preprint arXiv:1702.01992*, 2017. 5

[2] Akari Asai, Mohammadreza Salehi, Matthew E Peters, and Hannaneh Hajishirzi. Attentional mixtures of soft prompt tuning for parameter-efficient multi-task knowledge sharing. *arXiv preprint arXiv:2205.11961*, 3, 2022. 2

[3] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022. 2

[4] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 5

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 5, 8

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[7] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013. 7

[8] Yunhe Gao, Xingjian Shi, Yi Zhu, Hao Wang, Zhiqiang Tang, Xiong Zhou, Mu Li, and Dimitris N Metaxas. Visual prompt tuning for test-time domain adaptation. *arXiv preprint arXiv:2210.04831*, 2022. 2, 7

[9] Qidong Huang, Xiaoyi Dong, Dongdong Chen, Weiming Zhang, Feifei Wang, Gang Hua, and Nenghai Yu. Diversity-aware meta visual prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10878–10887, 2023. 2

[10] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. 2, 5, 7

[11] Ruixiang Jiang, Lingbo Liu, and Changwen Chen. Clip-count: Towards text-guided zero-shot object counting. *arXiv preprint arXiv:2305.07304*, 2023. 3

[12] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023. 2, 3, 7

[13] Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, Ethan Perez, and Davide Testuggine. Supervised multimodal bitransformers for classifying images and text. *arXiv preprint arXiv:1909.02950*, 2019. 5, 6

[14] Youngeun Kim, Yuhang Li, Abhishek Moitra, and Priyadarshini Panda. Do we really need a large number of visual prompts? *arXiv preprint arXiv:2305.17223*, 2023. 2

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[16] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020. 3

[17] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 2

[18] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 2, 7

[19] Yaowei Li, Ruijie Quan, Linchao Zhu, and Yi Yang. Efficient multimodal fusion via interactive prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2604–2613, 2023. 3, 5, 6

[20] Sheng Liang, Mengjie Zhao, and Hinrich Schütze. Modular and parameter-efficient multimodal fusion with prompting. *arXiv preprint arXiv:2203.08055*, 2022. 2, 3, 5, 6, 7

[21] Junfan Lin, Jianlong Chang, Lingbo Liu, Guanbin Li, Liang Lin, Qi Tian, and Chang-wen Chen. Being comes from not-being: Open-vocabulary text-to-motion generation with wordless training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23222–23231, 2023. 2

[22] Lingbo Liu, Jiaqi Chen, Hefeng Wu, Guanbin Li, Chenglong Li, and Liang Lin. Cross-modal collaborative representation learning and a large-scale rgbt benchmark for crowd counting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4823–4833, 2021. 1

[23] Lingbo Liu, Jianlong Chang, Bruce XB Yu, Liang Lin, Qi Tian, and Chang-Wen Chen. Prompt-matched semantic segmentation. *arXiv preprint arXiv:2208.10159*, 2022. 2

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1, 5

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[26] Jun Ma and Bo Wang. Segment anything in medical images. *arXiv preprint arXiv:2304.12306*, 2023. 1

[27] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. *Advances in Neural Information Processing Systems*, 35:9564–9576, 2022. 3

[28] Guanghui Qin and Jason Eisner. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*, 2021. 2, 3

[29] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1, 8

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 3

[31] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021. 3, 4, 5, 7

[32] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 7

[33] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021. 2, 3, 5, 7

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[35] Xin Wang, Devinder Kumar, Nicolas Thome, Matthieu Cord, and Frederic Precioso. Recipe recognition with large multimodal food dataset. In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2015. 5

[36] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 2

[37] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019. 5

[38] Hao Yang, Junyang Lin, An Yang, Peng Wang, Chang Zhou, and Hongxia Yang. Prompt tuning for generative multimodal pretrained models. *arXiv preprint arXiv:2208.02532*, 2022. 2, 7

[39] Xianjun Yang, Wei Cheng, Xujiang Zhao, Linda Petzold, and Haifeng Chen. Dynamic prompting: A unified framework for prompt tuning. *arXiv preprint arXiv:2303.02909*, 2023. 2

[40] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3081–3089, 2022. 3

[41] Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Cpt: Colorful prompt tuning for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*, 2021. 3

[42] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022. 3

[43] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022. 2

[44] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 3, 5

[45] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 3

[46] Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*, 2020. 1