Deep Correlated Prompting for Visual Recognition with Missing Modalities

Lianyu Hu, Tongkai Shi, Wei Feng, Fanhua Shang, Liang Wan[*]
College of Intelligence and Computing, Tianjin University
{hly2021,stk,wfeng,fhshang,lwan}@tju.edu.cn
https://github.com/hulianyuyy/Deep_Correlated_Prompting

Abstract

Large-scale multimodal models have shown excellent performance over a series of tasks powered by the large corpus of paired multimodal training data. Generally, they are always assumed to receive modality-complete inputs. However, this simple assumption may not always hold in the real world due to privacy constraints or collection difficulty, where models pretrained on modality-complete data easily demonstrate degraded performance on missing-modality cases. To handle this issue, we refer to prompt learning to adapt large pretrained multimodal models to handle missing-modality scenarios by regarding different missing cases as different types of input. Instead of only prepending independent prompts to the intermediate layers, we present to leverage the correlations between prompts and input features and excavate the relationships between different layers of prompts to carefully design the instructions. We also incorporate the complementary semantics of different modalities to guide the prompting design for each modality. Extensive experiments on three commonly-used datasets consistently demonstrate the superiority of our method compared to the previous approaches upon different missing scenarios. Plentiful ablations are further given to show the generalizability and reliability of our method upon different modality-missing ratios and types.

1 Introduction

Our human beings typically perceive information of multiple modalities such as visual, linguistic and audio signals to understand the world, where different signals drawn from various perspectives are inherently complementary. Thus, modeling and coordinating multimodal information is of great value for large-scale models to reason about real-world scenarios. Recently, multimodal models [34, 1, 22, 29] have developed fast powered by the large collection of multimodal data pairs and evolution of model architectures (e.g., Transformer [31]), which demonstrate promising performance across a series of downstream tasks, such as cross-model retrieval [8, 18, 29, 9], image captioning [24, 1] and image/video generation [23, 22]. Supported by their large capacity and general knowledge acquired by training upon web-scale data, these models have been more and more applied to daily work in our life (e.g., GPT-4 is used by millions of people).

However, there exist two major concerns that may hinder these impressive models from broader applications. First, a common assumption of previous methods is that the input data is modality-complete, which may not always hold in the real world due to privacy considerations, collection difficulty and security issues [27, 19]. When an input modality is missing in general real-world conditions, the performance of these models usually degrades a lot (regardless of training or testing settings) [27], which is easily influenced by the input completeness. Second, these large models are usually parameter-abundant [1, 6, 21] and require heavy computations [30, 45] to pretrain and

---

[*]Corresponding author

finetune on downstream tasks, whose computational demands may not always be available in most real-world applications due to limited computing resources. It's necessary to develop a new method to efficiently adapt these powerful methods to perform robustly against missing-modality scenarios.

Previous works [27, 42, 32, 28, 19] of multimodal learning have considerably explored the missing-modality issues. Earlier works [27, 42, 32, 28] mostly reconstruct the absent information of missing modalities or use other modalities to augment the missing modalities. MMP [42] has first introduced prompt learning to handle missing-modality scenarios by regarding different missing cases as different types of input. The multimodal backbone is kept frozen and only the newly introduced prompts are updated in the fine-tuning process, thus only incurring a few extra computations. However, MMP [19] simply inserts independent prompts into each layer, and overlooks the relationships among prompts and input features. The prompts across different layers lack cooperation to aggregate beneficial information to well guide the model predictions. The fixed prompts are used for different input samples which fail to consider the characteristics of various inputs. The complementary multimodal information of different input modalities is overlooked in the fine-tuning process.

To better adapt large multimodal models to missing-modality scenarios, we introduce deep correlated prompting (DCP) by capturing different types of correlations between prompts and input features. Specifically, to leverage the hierarchical semantics of different layers, we propose correlated prompts by perceiving beneficial information from preceding layers to instruct the features of the current layer. We further propose to dynamically generate the prompts according to the input features to better fit the characteristics of different inputs. To leverage the complementary information of multimodal inputs, we decompose the prompts into modal-common and modal-specific parts to guide each encoder to focus on its unique features. The proposed prompts are concatenated and prepended to the input and intermediate features of the multimodal backbone to instruct the model to alleviate the performance drop caused by the missing modality. The multimodal backbone keeps frozen during training and only the learnable prompts are tuned, thus offering high training efficiency. Extensive experiments on three widely-used datasets including MM-IMDb [2], UPMC Food-101 [33] and Hateful Memes [17] demonstrate consistently superior performance compared to other methods across all benchmarks, which verify the effectiveness of our proposed method. Ablation studies are further given to verify the generalizability and reliability of our method upon different missing-modality types and ratios.

2   Related Work

2.1   Missing-Modality for Multimodal Learning.

Earlier works of missing modalities for multimodal learning mostly generate the missing modality based on other modalities [42], or align features of latent representations for multiple modalities to help recognition [38, 15]. Recently, multimodal transformers [1, 22, 23] emerge as an effective tool to model information from various modalities and process them into a robust representation, which have shown impressive performance over a series of tasks [8, 18, 29, 9, 24, 11, 12]. However, these methods typically assume that the inputs are modality-complete, which may not always hold in real-world scenarios. When a modality is missing, these methods usually demonstrate degraded accuracy and lead to unstable performance [27].

To deal with missing modalities in multimodal transformers, MMIN [42] predicts the intermediate features of the missing modality based on other available modalities, by learning a common multimodal representation. SMIL [28] proposes a Bayesian meta-learning framework to estimate the latent features of the modality-incomplete data. It further explores the effects faced with severe modality-incomplete samples (e.g., 90% missing ratio). Ma et al. [27] test the robustness of multimodal transformers to missing modalities and their missing types, and propose a multitask optimization framework to improve it via modality fusion. Zeng et al. [40] propose a tag-assisted transformer Encoder network to handle the problem of missing uncertain modalities. ShaSpec [32] employs a shared head across different tasks to aggregate information from various input samples, which complements the features for missing modalities. However, these methods mostly assume that a fixed modality is missing, which is known in advance. Besides, these methods still require updating most parameters of the model, which consume heavy computations in downstream tasks. MMP [19] first introduced missing-aware prompts to handle scenarios with missing modalities with minimal extra computational costs. However, it simply inserts independent prompts into the intermediate features of the multimodal backbone, without considering the relationships between prompts of different

layers as well as the correlations between prompts and input features. In contrast, we carefully design the prompts by exploring the correlations between prompts and input features, and achieves much superior performance across all missing-modality scenarios upon three commonly-used datasets.

2.2 Prompt Learning

Prompt Learning is first explored in neural language processing (NLP), which adopts a "prompt" to modify the input text to instruct the pre-trained model for downstream tasks. Earlier works [6, 14] usually adopt manually designed prompts to improve the generalizability of large models over downstream tasks. Later, prompt tuning methods [26, 20, 25] emerge by prepending learnable prompts to the input features in the training phase to automate the optimization process. Recently, prompt learning is also introduced into computer vision tasks [4, 13, 37] and multimodal learning tasks [36, 44, 35, 16, 43]. CoOp [44] is first proposed to insert learnable soft prompts besides input images to adapt vision-language models to various vision tasks. CoCoOp [43] generates an image-conditional prompt to utilize the power of input features. ProGrad [46] only updates the prompts whose gradients are aligned to the "general knowledge" generated by the original prompts. KgCoOp [36] tries to align the output embeddings of the text encoder with those of the pretrained CLIP to preserve beneficial information. MaPLe [16] injects deep learnable soft prompts into both the image and text branches to enable prompt collaboration. DualPrompt [35] extends the prompts to learn different task information conditionally in continual learning. DePT [41] decouples base-specific knowledge from feature channels into an isolated feature space during prompt tuning. These works have demonstrated the effectiveness of prompt learning to adapt large-scale vision-language models for downstream tasks with minimal extra computation costs. In this paper, we introduce prompt learning into multimodal models to increase their robustness to missing-modality scenarios, via attaching different types of prompts according to various missing cases.

3 Method

3.1 Overall framework

Problem definition. We first provide a succinct overview of the missing-modality scenario addressed in this paper. Without loss of generalizability, we explore multimodal inputs with $M = 2$ modalities m_1 and m_2 (e.g., text and image), which is naturally extensible to more modalities. Specifically, with a multimodal dataset $D = \{D^c, D^{m_1}, D^{m_2}\}$, we denote $D^c = \{x^{m_1}, x^{m_2}, y\}$ as the modality-complete case, where x denotes the input and y represents the label. We let $D^{m_1} = \{x^{m_1}, y\}$ and $D^{m_2} = \{x^{m_2}, y\}$ denote the modality-incomplete cases, wherein one modality is absent (e.g., missing text and missing image). Some examples of modality-complete and modality-incomplete inputs are depicted in the lower part of Fig. 1.

Recap MMP [19]. MMP [19] first introduced missing-aware prompts to handle missing-modality cases. Specifically, it assigns $2^M - 1$ types of prompts for tasks involving M modalities (e.g., 3 prompts for vision-language tasks, including one prompt for modality-complete case, one prompt for image-only case, and one prompt for text-only case). Given an input sample, it first selects the prompts corresponding to the missing case and then prepends the prompts to the input and intermediate features of the multimodal backbone, instructing the pretrained model to perform prediction. In this procedure, only the learnable prompts are updated and the multimodal backbone keeps frozen.

3.2 Overall framework

Though MMP [19] has achieved notable progress in improving the robustness of multimodal models upon missing-modality cases by incurring minimal computational costs, it only assigns independent prompts to the input and intermediate features, which (1) fails to consider the relationships between prompts of different layers, and (2) lacks the correlations between prompts and input features, and (3) overlooks the complementarity of multimodal inputs. To better adapt the pretrained multimodal model for missing-modality scenarios, we propose to design three types of missing-aware prompts by capturing the relationships between prompts and inputs. First, we generate missing-aware prompts by leveraging the correlations of preceding prompts across multiple layers and various modalities. Second, we dynamically generate the prompts for each input sample to fit its characteristics. Third,
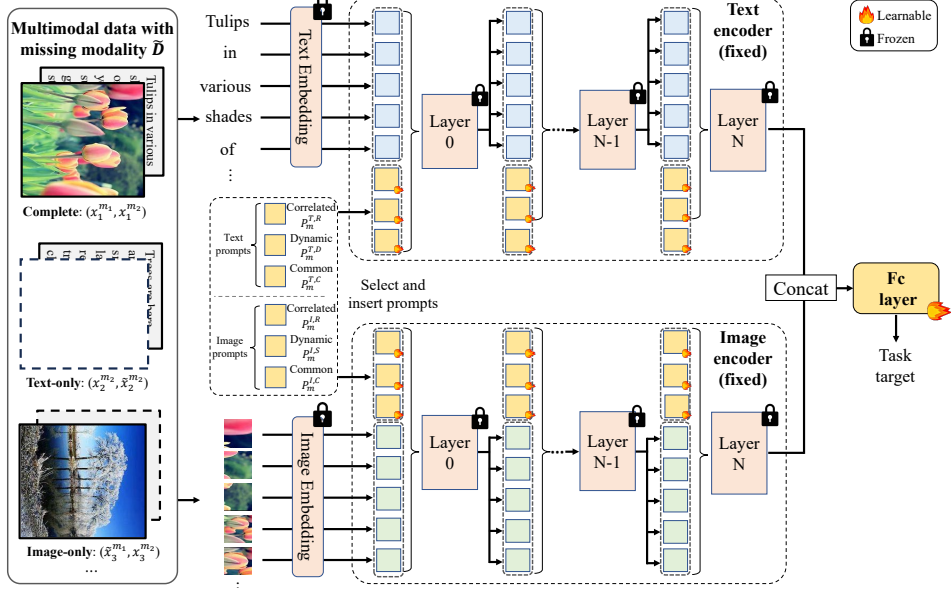
Figure 1: The overview of our proposed framework. We first select the prompt P_m^T and P_m^I with $m \in \{c, m_1, m_2\}$ for the text encoder and image encoder according to the missing case (e.g., complete, text-only, image-only) of the multimodal inputs (x^{m1}, x^{m2}). The prompt P_m^T (P_m^I) is composed of three types of missing-aware prompts including the correlated prompts $P_m^{T,R}$ ($P_m^{I,R}$), dynamic prompts $P_m^{T,D}$ ($P_m^{I,D}$) and modal-common prompts $P_m^{T,C}$ ($P_m^{I,C}$). Then we prepend the prompts to the inputs and intermediate features of both encoders to instruct the model to fit the missing case. Finally, we concatenate the task-related token of both encoders as the final representation, and pass it through a fully-connected layer for class prediction. In the whole procedure, only the fully-connected (fc) layer and deep correlated prompts are updated while others keep frozen.

we decompose multimodal prompts into modal-common parts and modal-specific parts, which fuses beneficial information from other modalities and enables each encoder to focus its unique features.

Fig. 1 shows the framework overview of our proposed method. Specifically, without loss of generalizability, we adopt the widespread two-stream multimodal method CLIP [29] as our backbone. Given the input text and image, we first employ pretrained text and image embedding layers from the pretrained CLIP [29] to convert them into token sequences. For each input sample, we select the corresponding prompt P_m^T and P_m^I for the text encoder and image encoder, respectively, given the type of missing modality with $m \in \{c, m_1, m_2\}$. The prompt for each encoder is composed of three types of missing-aware prompts including the correlated prompts $P_m^{T,R}$ ($P_m^{I,R}$), dynamic prompts $P_m^{T,D}$ ($P_m^{I,D}$) and modal-common prompts $P_m^{T,C}$ ($P_m^{I,C}$). We concatenate the missing-aware prompts and prepend them to the input tokens x^{m_1} (x^{m_2}) as a whole sequence to process. Finally, we concatenate the task-related token of both encoders as the final output representation, and pass it through a fully-connected layer for task prediction. In this procedure, only the parameters of the fully-connected layer and the newly introduced deep correlated prompts are updated in the training procedure, and the backbone (i.e., the text embedding layer, image embedding layer, text encoder and image encoder) is kept frozen. The illustration of our proposed prompts is given in Fig. 2. We next introduce our prompting designs in detail.

3.3 Deep Correlated Prompt Learning

3.3.1 Correlated Prompts

MMP [19] append independent prompts to the input and intermediate features of the multimodal backbone to guide model predictions. Though it could theoretically provide enough guidance for features in each layer, the prompts across each layer lack synergy, which fails to cooperate with the representations of each layer that various semantics. We argue that prompts across consecutive layers should be closely correlated to receive necessary semantics from preceding layers to instruct

(a) Baseline   (b) MMP

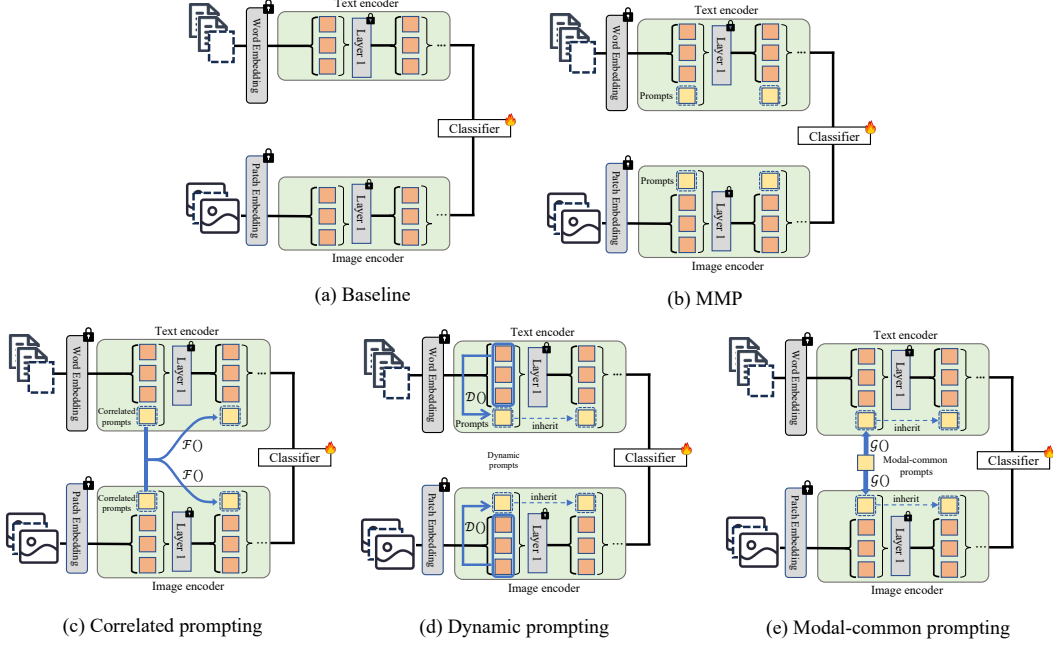(c) Correlated prompting   (d) Dynamic prompting   (e) Modal-common prompting

Figure 2: (1) Baseline, which simply uses fixed image encoder and text encoder and only finetunes the classifier to handle downstream tasks. (2) MMP, which inserts independent prompts at each layer to guide the model to handle missing-modality cases. (3) Correlated prompts, which generate the prompts of the next layer based on the prompts of both modalities in the current layer to enable cooperation of prompts from both modalities. (4) Dynamic prompts, which dynamically computes the prompts based on different input features to better guide the behavior of the model, avoiding using fixed prompts for different inputs. (5) Modal-common prompts, which store the shared information across different modalities and facilitate the model to encode modal-specific information to better handle the missing scenarios in each modality.

the current layer to handle missing modalities. Thus, we propose to generate the prompts of the current layer based on the observation of the preceding prompts. Besides, the features of different modalities usually contain complementary information. Leveraging the complementary information from various modalities could further help the model predictions. We propose to incorporate the beneficial information from multimodal inputs to help guide the outputs of each modality.

Specifically, we prepend the prompts to the input features and the intermediate features of the multimodal backbone up to a depth of J. Taking the image encoder as an example, the calculation process of the i_{th} ($i \in [0, \ldots, J-1]$) layer $\mathcal{V}_i$ could be expressed as:

$$[\_\_, X_i^I] = \mathcal{V}_i([P_{m,i-1}^{I,R}, X_{i-1}^I]). \tag{1}$$

Here, X_i^I is the features of the i_{th} layer in the image encoder and $P_{m,i-1}^{I,R}$ is the newly introduced correlated prompts of the $(i-1)_{th}$ layer. After the J_{th} layer, the correlated prompt is retained from the previous layer, and the calculation process for the i_{th} ($i \in [J, \ldots, N-1]$) layer $\mathcal{V}_i$ can be presented as:

$$[P_{m,i}^{I,R}, X_i] = \mathcal{V}_i([P_{m,i-1}^{I,R}, X_{i-1}^I]) \tag{2}$$

where N denotes the length of all layers.

To leverage the correlations of prompts between different layers, we generate the prompts of the i_{th} ($i \in [1, \ldots, J-1]$) layer $\mathcal{V}_i$ by observing prompts of its preceding layer $\mathcal{V}_{i-1}$ to inject beneficial information, as:

$$P_{m,i}^{I,R} = \mathcal{F}_{i-1}^I(P_{m,i-1}^{I,R}) \tag{3}$$

where $\mathcal{F}_{i-1}^{I,R}(\cdot)$ denotes the prompt generation function for the $(i-1)_{th}$ layer in the image encoder. To decrease the required parameters and add non-linearity to $\mathcal{F}(\cdot)$, we adopt $\mathcal{F}(\cdot)$ as a bottleneck

MLP with the GELU activation [10] in between, followed by a LayerNorm (LN) function [3] as:

$$\mathcal{F}(\cdot) = \mathrm{LN}(\mathrm{Fc}(\mathrm{GELU}(\mathrm{Fc}(\cdot)))). \tag{4}$$

The intermediate feature dimension of $\mathcal{F}(\cdot)$ is r (usually $r = \frac{1}{16}$) times of its input channel dimension. For the correlated prompts at the input level, we leave it randomly initialized without the generation procedure.

To leverage complementary information from different modalities, we incorporate prompts from both encoders to fuse their distinct semantics for instructions. Taking the image encoder as an example, we generate the prompts of the i_{th} ($i \in [1, \ldots, J-1]$) layer $\mathcal{V}_i$ based on the prompts of the layer $\mathcal{V}_{i-1}$ in the image encoder and the prompts of layer $\mathcal{T}_{i-1}$ in the text encoder as:

$$P_{m,i}^I = \mathcal{F}_{i-1}^I(\mathrm{Concat}(P_{m,i-1}^I, P_{m,i-1}^T)). \tag{5}$$

Here, Concat denotes the concatenation operation. Instead of leaving prompts uncorrelated across different modalities, our design improves mutual synergy between different modalities to help adapt to various missing scenarios.

3.3.2 Dynamic prompts

Different input samples usually contain information of various semantics. Using fixed prompts for different inputs may not well fit their distinct features, and thus can't offer enough guidance for the model to fit different missing-modality scenarios. Thus, we propose to dynamically generate the prompts based on the input features to adjust the instructions to fit the characteristics of different inputs.

Specifically, taking the image encoder as an example, we generate the dynamic prompts $P_m^{I,D}$ based on the input features X_0^I as:

$$P_m^{I,D} = \mathcal{D}^I(X_0^I) \tag{6}$$

where $\mathcal{D}^I$ denotes the dynamic prompt generation function for the image encoder. To deal with the varying length of tokens for different inputs, we instantiate $\mathcal{D}^I$ as a self-attention layer [31], whose architecture can be expressed as :

$$\mathcal{D}(\cdot) = \mathrm{LN}(\mathrm{MLP}(\mathrm{LN}(\mathrm{MHA}(\cdot)))). \tag{7}$$

Here, MHA is the multi-head attention mechanism in transformers [31]. We set the number of heads as 1 for simplicity. The dynamic prompts are only inserted at the input level. For the subsequent layers, the prompts are retained from the previous layers and updated together with intermediate features following Eq. 2.

3.3.3 Modal-common prompts

Multimodal inputs usually contain both modal-specific information and modal-common information across different modalities. By disentangling the modal-common features from the modal-specific features, each modality could aggregate beneficial information from the shared features across various input modalities and build more powerful representations based on its unique characteristics. We decompose the multimodal missing-aware prompts into modal-common parts and modal-specific parts to store common characteristics across different modalities and model-specific features for each modality, respectively.

Specifically, we introduce a modal-common prompt P_m^C for multimodal inputs to embody the mutual features across different modalities, and accordingly encourage the proposed correlated prompts and dynamic prompts to embed modal-specific instructions. To cooperate with the features of different modalities, we project the modal-common prompt P_m^C to the image and text space to obtain $P_m^{T,C}$ and $P_m^{I,C}$ via projection layers, respectively, as:

$$\begin{aligned} P_m^{T,C} &= \mathcal{G}^T(P_m^C) \\ P_m^{I,C} &= \mathcal{G}^I(P_m^C). \end{aligned} \tag{8}$$

Here, $\mathcal{G}^T$ and $\mathcal{G}^I$ are the projection layers for the text modality and the image modality, respectively, which are both instantiated as a MLP with the intermediate feature reduction factor r=16. We only insert modal-common prompts at the input level and leave the prompt in the intermediate layers reserved and updated following Eq. 2.

6

4 Experiments

4.1 Experimental Setup

Datasets. We follow the previous works [19, 27] to evaluate our methods:

MM-IMDb [2] is currently the largest publicly available multimodal dataset for genre prediction on movie genre classification. It is notated with both image and text modalities for 25959 movies. As a movie might have several genres, the genre prediction task is thus a multi-label classification task.

UPMC Food-101 [33] is a large multimedia dataset consisting of noisy image-text pairs collected from Google Image Search with 101 food categories. It has identical categories to the largest publicly available ETHZ Food-101 dataset [5], used to classify the categories of different foods.

Hateful Memes [17] is a multimodal dataset for hateful meme detection (image + text) that contains 10,000+ new multimodal examples created by Facebook AI. To prevent the model from relying on a single modality, it is constructed to make unimodal models more likely to fail.

Implementation details. We use CLIP [29] as our multimodal backbone, with ViT-B/16 [7] as the image encoder. For the image input, we follow CLIP [29] to resize input images into 224×224. The patch size is set as 16 for the multimodal transformer. For the text modality, we use the tokenizer from pretrained CLIP to tokenize the text input. The maximum length of text inputs is 77. We freeze all the parameters of both the image encoder and text encoder, and only tune the parameters of deep correlated prompts and the fc layer (for task target). We set the length L_p of learnable prompts as 36 and prepend them to the features of $M = 6$ layers. We use Adam optimizer with initial learning rate of 1e-2 and weight decay 2e-2. The learning rate is warmed up for 10% of the total training steps and then is decayed linearly to zero. We perform our experiments with batch size of 4 on a 3090 GPU. For the missing modality, we stop feeding the inputs into the corresponding encoder and use a zero-filled tensor as the output instead.

Metrics. We use corresponding proper metrics for each dataset to evaluate our method. For MM-IMDb [2], we adopt F1-Macro to measure the multi-label classification performance. For UPMC Food-101 [33], we employ the top-1 classification accuracy to evaluate the recognition performance. For Hateful Memes [17], we use Area Underthe Receiver Operating Characteristic Curve (AUROC).

Setting of Missing Modality. In this paper, we focus on the general realistic scenarios in real life, where any modality may appear in both training and testing phases. To stimulate this condition, we follow MMP [19] to define the missing rate η as the proportion of modality-incomplete data to the entire dataset. For vision-language tasks, there exist three types of missing cases: missing-both (text and image), missing-text and missing-image. For the missing-both case, the training and testing data are composed of $\frac{\eta}{2}$ text-only data, $\frac{\eta}{2}$ image-only data and (1-η) modality-complete data. For the missing-text and missing-image cases, the training and testing data are composed of η image-only (text-only) data and (1-η) modality-complete data. This definition could be naturally extended to data with more modalities by using ($\frac{\eta}{M^2-2}$) modality-incomplete data for each missing case and (1-η) complete data. In our experiments, we use η=70% by default.

4.2 Experimental Results

Effectiveness. We first verify the effectiveness of our proposed components across different missing cases including missing-image, missing-text and missing-both. We conduct the experiments upon the MM-IMDb [2] dataset with missing rates ranging from 0% to 100%. The methods used for comparison include (1) baseline, which directly sets the features as zeros when a modality is missing; (2) Ours (A), which only equips the correlated prompts; (3) Ours (B), which equips both the correlated prompts and the dynamic prompts; (4) Ours, which uses all the three proposed prompts. The only difference between our method and the baseline is inserting learnable prompts at each layer which only bring few extra parameters. The results are shown in Fig. 3.

It's first noticed that compared to the baseline, all our three variants could notably promote the performance across all missing rates for various missing-modality cases, demonstrating strong robustness to different missing-modality scenarios. Using the correlated prompts boosts the performance most, and equipping the dynamic prompts or modal-common prompts offers a similar performance boost. Using all three proposed prompts achieves the best performance, which verifies the effectiveness of
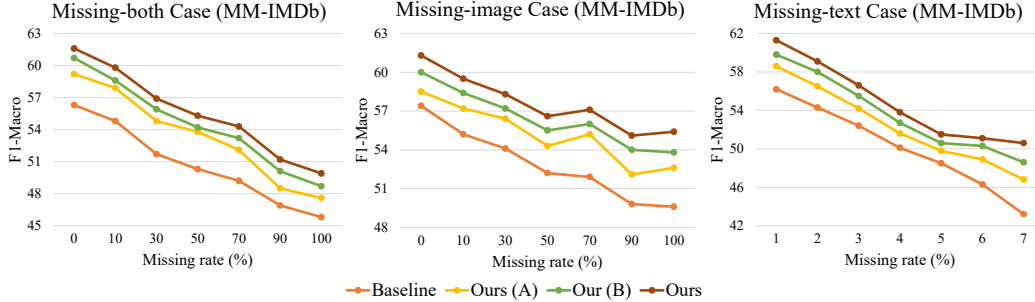
Figure 3: Comparison of our final model (Ours) with (1) baseline, which directly drops the features when a modality is missing; (2) Ours (A), which only equips the correlated prompts; (3) Ours (B), which equips both the correlated prompts and the dynamic prompts. The experiments are conducted on the val set of MM-IMDb dataset [2] across different missing rates (0–100%) upon three different missing-modality scenarios (missing-both, missing-image and missing-text).

capturing correlations between prompts and input features. An interesting observation is that the performance degradation is much smaller when input images are missing compared to missing-both and missing-text, which shows that text is more important for the task target on this dataset.

Ablation study. We test the configurations for our proposed three prompts on the val set of MM-IMDb dataset in Tab. 1, Tab. 2 and Tab. 3, respectively. In the upper part of Tab. 1, we first explore how to leverage the correlations between prompts of different layers. "No projection" denotes using independent prompts for each layer and "Fc" denotes using a fully connected layer for projection. It's observed that compared to using independent prompts, either using a fc or a MLP offers a notable performance boost. The MLP gives better performance for its non-linearity and we use it by default. We then explore the prompt depth J (how many layers to insert prompts) in the middle of Tab. 1. We notice that the performance reaches a peak when J=6, and either decreasing it or increasing it would degrade the performance. Finally, we verify the efficacy of incorporating multimodal prompts for instruction in the bottom part of Tab. 1. It's observed that introducing bi-modal prompts to generate the prompts of the current layer gives better performance, which shows that different modalities can offer complementary information.

We test the configurations for the dynamic prompts in Tab. 2. We first explore the prompt depth for the dynamic prompts. It's observed that the performance reaches a peak when the prompt depth equals 1, and continues to decrease when the prompt depth increases. We thus set the prompt depth as 1. We then explore the generation functions for the dynamic prompts. Besides the default attention mechanism, we test other alternatives which use a maximum, minimum or average pooling layer followed by a fc layer to generate the prompts. It's observed that all approaches could give a notable performance boost, and the attention function achieves the best performance.

Table 1: Ablations for the correlated prompts.

Configurations	F1-Macro(%)
No projection	52.54
Fc	53.76
MLP (r=16)	**54.24**
Depth = 3	53.74
Depth = 6	**54.24**
Depth = 12	53.86
uni-modal	53.52
bi-modal	**54.24**

Table 2: Ablations for the configurations of dynamic prompts.

Configurations	F1-Macro(%)
Depth = 1	**54.24**
Depth = 2	54.08
Depth = 3	53.76
Depth = 6	53.52
Attention	**54.24**
Max & Projection	53.12
Min & Projection	53.35
Avg & Projection	53.42

Table 3: Ablations for the modal-common prompts.

Configurations	F1-Macro(%)
Depth = 1	54.24
Depth = 2	54.16
Depth = 3	54.42
Depth = 6	**54.52**
Fc	53.76
MLP (r=4)	**54.36**
MLP (r=8)	54.12
MLP (r=16)	54.24

We explore the configurations for the modal-common prompts in Tab. 3. We first examine the prompt depth for the modal-common prompts. It's observed that as the prompt depth ranges from 1 to 6, our model achieves similar results and reaches a peak when the prompt depth equals 6. Considering the accuracy-computation trade-off, we set the prompt depth as 1 by default. We then explore the projection functions for the modal-common prompts. Generally, we observe that using a MLP for

| (a) Test with missing-both | (b) Test with missing-image | (c) Test with missing-text |

Figure 4: Ablations on the generalizability to different testing scenarios across various missing rates on the val set of MM-IMDb dataset [2]. (a) All models are trained on *missing-both* cases, and evaluated on *missing-both* cases with different missing rates. (b) Models are trained on *missing-both* or *missing-image* cases, and evaluated on *missing-image* cases with different missing rates. (c) Models are trained on *missing-both* or *missing-text* cases, and evaluated on *missing-text* cases with different missing rates.

projection performs better than using a fc for projection. Using different channel reduction factors r including 4, 8 and 16 achieves similar performance. We thus use a MLP with r=16 by default.

Generalizability. We conduct experiments to verify the generalizability of our method when trained and tested upon different missing cases. We evaluate models trained on *missing-both* cases or *missing a specific modality*, and test them on the missing-both, missing-image and missing text cases in Fig. 4(a), (b) and (c), respectively. It's first noticed that all our variants outperform the baseline by a large margin across various missing cases and missing rates. Models trained with a certain modality missing (e.g., missing-text, missing-image), usually perform slightly better than models trained upon missing-both cases when tested with a certain modality missing. Models trained with missing both modalities perform robustly to cases with any missing type (e.g. missing-both, missing-text and missing-image). We further observe the model trained with higher missing rates (e.g., 70%), is more robust to high-rate missing cases (70%–100%) during testing than models trained with lower missing rates (e.g., 30% and 50%). It is concluded that models trained on missing-both cases are robust to various missing cases, which demonstrate strong robustness compared to training with one modality.

Comparison with other methods. We compare our method with recent approaches upon three commonly-used datasets, i.e., MM-IMDb [2], UPMC Food-101 [33] and Hateful Memes [17], to verify its effectiveness upon different missing-modality scenarios. We include the following methods for comparison (1) baseline, which directly drops the features when a modality is missing; (2) CoOp [44], which only prepends prompts at the input level; (3) MMP [19], which inserts independent prompts for the input and intermediate features of the multimodal backbone; (4) MaPLe [16], which generates the prompts in the image encoder based on those of the text encoder; (5) DePT [41], which decouples base-specific knowledge from feature channels into an isolated feature space during prompt tuning. MMP [19] is originally based on the ViLT backbone and we reimplement it following the same protocol as ours. We compare these methods across three different missing rates including η=50%, η=70% and η=90% upon three various missing-modality cases including missing-image, missing-text and missing-both in Table. 4. Our method largely outperforms other methods on three datasets across different missing rates, which verifies its robustness to different missing scenarios. Compared to MMP [19], our method exceeds it over all missing cases by a large margin, which demonstrates the superiority of leveraging the correlations between prompts and input features.

An interesting observation is that on most datasets (e.g., MM-IMDb [2] and Food101 [33]), missing text input usually has higher effects on the performance than missing image input. We figure that texts are more crucial for the tasks by providing more detailed explanations and precise captions than images on these two datasets. Instead, on the Hateful Memes dataset [17], it's observed that missing image input influences the results more than missing text or missing both. This may reflect the different focus when constructing various datasets.

Efficiency. Compared with only finetuning the last fc layer to fit downstream tasks, our deep correlated prompts only introduce extra 4.0M parameters, which just consume 2.4% of the entire model (151M), but could notably improve the performance by a large margin over different missing cases and missing rates.

9

Table 4: Comparison with CoOp [44], MMP [19], MaPLe [16] and DePT [41] on the MM-IMDb [2], UPMC Food-101 [33], and Hateful Memes [17] datasets under various missing-modality cases with different missing rates. The bold number indicates the best performance.

Datasets	Missing rate η	Train/Test Image	Train/Test Text	Validation set CoOp	MMP	MaPLe	DePT	Ours	Testing set CoOp	MMP	MaPLe	DePT	Ours
MM-IMDb (F1-Macro)	50%	100%	50%	51.23	52.07	52.76	53.87	**55.23**	48.06	48.88	49.58	50.64	**52.13**
		50%	100%	53.04	54.52	55.26	56.04	**57.32**	49.89	51.46	52.32	52.78	**54.32**
		75%	75%	51.46	52.12	52.87	54.02	**55.45**	48.37	49.32	49.56	50.87	**52.32**
	70%	100%	30%	47.26	48.23	48.75	49.87	**51.35**	44.13	45.64	45.52	46.38	**48.52**
		30%	100%	52.32	53.21	53.98	55.04	**56.21**	48.82	50.52	50.64	52.13	**53.14**
		65%	65%	50.22	51.34	52.31	53.17	**54.24**	46.84	48.12	49.16	50.32	**51.42**
	90%	100%	10%	47.86	48.84	50.12	50.98	**52.36**	44.76	45.32	46.84	47.56	**49.26**
		10%	100%	51.65	52.36	53.14	54.12	**55.42**	48.32	49.12	50.13	50.88	**52.22**
		55%	55%	47.44	48.04	48.82	49.98	**51.26**	44.12	44.87	45.12	46.54	**48.04**
Food101 (Accuracy)	50%	100%	50%	77.36	78.24	79.87	80.24	**82.33**	77.45	77.89	79.64	80.16	**82.11**
		50%	100%	86.98	87.12	87.48	87.85	**89.23**	87.02	87.16	87.35	82.14	**89.12**
		75%	75%	81.76	81.98	82.58	83.26	**85.25**	81.24	81.72	82.34	83.12	**85.24**
	70%	100%	30%	76.65	76.74	76.87	76.87	**79.18**	76.34	76.52	77.02	77.34	**78.87**
		30%	100%	85.21	86.12	86.36	86.52	**87.53**	84.78	85.64	85.89	86.12	**87.32**
		65%	65%	79.14	79.56	80.06	81.85	**82.38**	78.87	79.12	79.84	81.46	**81.87**
	90%	100%	10%	72.65	73.74	73.25	74.22	**75.54**	71.87	73.14	73.46	74.12	**75.26**
		10%	100%	82.16	82.78	83.42	84.02	**86.26**	81.67	82.14	83.12	83.56	**85.78**
		55%	55%	77.36	77.78	78.26	78.66	**80.39**	76.46	76.58	77.85	78.12	**79.87**
Hateful Memes (AUROC)	50%	100%	50%	58.32	58.56	58.78	59.31	**60.24**	60.56	60.31	60.87	61.87	**62.32**
		50%	100%	60.34	61.12	61.34	61.78	**62.34**	62.41	62.35	63.13	63.88	**64.46**
		75%	75%	62.34	62.87	63.14	63.24	**63.78**	64.87	65.84	65.46	65.86	**66.02**
	70%	100%	30%	58.54	59.02	59.36	60.02	**60.56**	60.74	61.12	61.26	61.56	**62.82**
		30%	100%	60.12	60.78	61.32	61.54	**62.32**	62.74	63.24	63.14	63.48	**64.12**
		65%	65%	62.34	62.56	63.12	63.32	**63.78**	64.82	65.04	65.23	65.48	**66.08**
	90%	100%	10%	58.02	57.34	58.32	59.02	**60.34**	60.03	57.21	60.74	61.14	**62.08**
		10%	100%	59.02	59.32	60.21	60.56	**61.34**	61.46	61.52	61.87	62.42	**63.87**
		55%	55%	62.32	62.56	63.24	63.78	**64.34**	64.32	63.34	64.85	65.37	**66.78**

Performance v.s. MMP when owning comparable parameters. We compare our method with MMP by allowing them to own comparable parameters in Tab. 5. Specially, either when we decrease the required parameters of our method to 0.2M (0.2% of the entire model) by reducing the channel dimension, or we simultaneously increase the required parameters of both methods, our method consistently outperforms MMP, which verifies its effectiveness.

Table 5: Performance v.s. MMP when owning comparable parameters on the val set of MM-IMDb [2].

Parameters	0.2M	1.6M	2.8M	4.0M	5.2M	6.4M
MMP(%)	51.34	50.74	50.82	50.72	50.64	50.46
Ours(%)	53.14	53.56	53.88	**54.24**	54.12	53.98

Limitations and broader impacts. The limitations include (1) we only test the effectiveness upon two commonly-used two-stream multimodal models, and don't apply our method to other popular multimodal models. (2) we only include two modalities in the experiments, and will incorporate more modalities in the future. The broader impacts include (1) The missing-modality cases happen at times in real life. This paper proposes a novel method by adapting large multimodal models towards missing-modality scenarios, which increases the robustness of large multimodal models. (2) Our methods can notably decrease the required computations compared to previous methods upon missing-modality learning, which achieves a better accuracy-computation trade-off in real life.

5   Conclusion

In this paper, we tackle two main challenges in multimodal learning, including (1) any modality may be missing in any learning phase (e.g., training, testing or both), and (2) how to efficiently adapt large-scale multimodal transformers to fit missing-modality cases. We propose deep correlated prompting by leveraging the correlations between prompts of different layers and the relationships between prompts and input features. Results on three diverse datasets across various missing types and missing ratios verify the effectiveness of our proposed method.

Acknowledgments and Disclosure of Funding

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

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

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv preprint arXiv:2203.17274*, 3:11–12, 2022.

[5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[7] 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.

[8] Maksim Dzabraev, Maksim Kalashnikov, Stepan Komkov, and Aleksandr Petiushko. Mdmmt: Multidomain multimodal transformer for video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3363, 2021.

[9] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 214–229. Springer, 2020.

[10] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[11] Lianyu Hu, Liqing Gao, Zekang Liu, and Wei Feng. Continuous sign language recognition with correlation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2529–2539, 2023.

[12] Lianyu Hu, Liqing Gao, Zekang Liu, and Wei Feng. Scalable frame resolution for efficient continuous sign language recognition. *Pattern Recognition*, 145:109903, 2024.

[13] 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.

[14] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

[15] Mengmeng Jing, Jingjing Li, Lei Zhu, Ke Lu, Yang Yang, and Zi Huang. Incomplete cross-modal retrieval with dual-aligned variational autoencoders. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3283–3291, 2020.

[16] 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.

[17] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33:2611–2624, 2020.

[18] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021.

[19] Yi-Lun Lee, Yi-Hsuan Tsai, Wei-Chen Chiu, and Chen-Yu Lee. Multimodal prompting with missing modalities for visual recognition. In *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, pages 14943–14952, 2023.

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

[21] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, and Jianfeng Gao. Multimodal foundation models: From specialists to general-purpose assistants. *arXiv preprint arXiv:2309.10020*, 1, 2023.

[22] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[23] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

[24] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022.

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

[26] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

[27] Mengmeng Ma, Jian Ren, Long Zhao, Davide Testuggine, and Xi Peng. Are multimodal transformers robust to missing modality? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18177–18186, 2022.

[28] Mengmeng Ma, Jian Ren, Long Zhao, Sergey Tulyakov, Cathy Wu, and Xi Peng. Smil: Multimodal learning with severely missing modality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2302–2310, 2021.

[29] 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.

[30] Teven Le Scao, Thomas Wang, Daniel Hesslow, Lucile Saulnier, Stas Bekman, M Saiful Bari, Stella Biderman, Hady Elsahar, Niklas Muennighoff, Jason Phang, et al. What language model to train if you have one million gpu hours? *arXiv preprint arXiv:2210.15424*, 2022.

[31] 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.

[32] Hu Wang, Yuanhong Chen, Congbo Ma, Jodie Avery, Louise Hull, and Gustavo Carneiro. Multi-modal learning with missing modality via shared-specific feature modelling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15878–15887, 2023.

[33] 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.

[34] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

[35] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648. Springer, 2022.

[36] Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6757–6767, 2023.

[37] 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.

[38] Qiyue Yin, Shu Wu, and Liang Wang. Unified subspace learning for incomplete and unlabeled multi-view data. *Pattern Recognition*, 67:313–327, 2017.

[39] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Trans. Mach. Learn. Res.*,

2022, 2022.

[40] Jiandian Zeng, Tianyi Liu, and Jiantao Zhou. Tag-assisted multimodal sentiment analysis under uncertain missing modalities. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1545–1554, 2022.

[41] Ji Zhang, Shihan Wu, Lianli Gao, Hengtao Shen, and Jingkuan Song. Dept: Decoupled prompt tuning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[42] Jinming Zhao, Ruichen Li, and Qin Jin. Missing modality imagination network for emotion recognition with uncertain missing modalities. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2608–2618, 2021.

[43] 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.

[44] 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.

[45] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv preprint arXiv:2310.01852*, 2023.

[46] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15659–15669, 2023.

A   Appendix / supplemental material

Compatibility with complete data. In this paper, we have previously assumed that there is no guarantee modality-complete data can be entirely collected due to various factors in real-world scenarios. However, these still exist publicly-available modality-complete datasets which are carefully collected and annotated for model training. Thus, we perform additional experiments to compare our method with the baseline and input-level prompting by training on modality-complete data. To stimulate the missing-modality case during testing, we randomly select data with different modality-missing cases for each data pair (i.e., text-only, image-only, or complete) during training. Note that, different from other experimental settings introduced in the paper, here one data pair can be in various missing-modality cases at different training epochs. We plot the results in Fig. 5. It is observed that our method consistently outperforms the baseline and input-level prompting with better performance across different missing ratios.
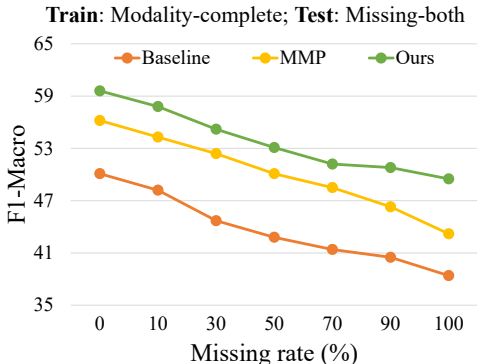

Figure 5: All models are trained with modality-complete data, where each data pair can be randomly assigned with different missing modalities (i.e., text-only, image-only, and modality-complete) at different training epochs to mimic the possible missing modalities during testing. Evaluation is on missing-both cases with different missing rates.

Deployment on other backbones. We deploy our proposed method upon other commonly-used vision-language models like CoCa [39] and ViLT [18], to verify its flexibility. CoCa is trained jointly with contrastive loss and captioning loss, which subsumes model capabilities from both contrastive approaches like CLIP [29] and generative methods like SimVLM [34]. ViLT is a single-stream multimodal backbone which directly concatenates the input text features and image features and feed them into a common transformer for processing. We adopt ViT-B/32 [7] as the image encoder for CoCa, and list the results in Table. 6 and Table 7, respectively. It is observed that our method can bring notable performance boost across three datasets compared to the baseline, which verifies its flexibility across various large-scale vision-language models.

Table 6: Quantitative results with CoCa [39] as our backbone on the MM-IMDB, UPMC Food-101, and Hateful Memes datasets with missing rate $\eta = 70\%$ under various modality-missing scenarios. The bold number indicates the best performance.

Datasets	Image	Text	Baseline	Ours
MM-IMDb (F1-Macro)	100%	30%	38.96	**49.34**
	30%	100%	45.78	**55.12**
	65%	65%	42.35	**51.24**
Food101 (Accuracy)	100%	30%	73.41	**77.56**
	30%%	100%	82.34	**86.26**
	65%	65%	77.24	**80.46**
Hateful Memes (AUROC)	100%	30%	54.87	**58.36**
	30%	100%	56.46	**60.34**
	65%	65%	57.82	**61.83**

14

Table 7: Quantitative results with ViLT [18] as our backbone on the MM-IMDB, UPMC Food-101, and Hateful Memes datasets with missing rate $\eta = 70\%$ under various modality-missing scenarios. The bold number indicates the best performance.

Datasets	Image	Text	MMP	Ours
MM-IMDb (F1-Macro)	100%	30%	39.22	**45.26**
	30%	100%	46.30	**51.24**
	65%	65%	42.66	**48.45**
Food101 (Accuracy)	100%	30%	74.53	**78.85**
	30%	100%	86.18	**86.76**
	65%	65%	79.08	**80.85**
Hateful Memes (AUROC)	100%	30%	59.11	**61.24**
	30%	100%	63.06	**64.12**
	65%	65%	66.07	**66.68**

Table 8: Ablations for the prompt length on the MM-IMDB dataset with missing rate $\eta = 70\%$. The bold number indicates the best performance.

Prompt length	12	24	36	48	60
F1-Macro(%)	53.56	53.88	**54.24**	54.12	53.98

Ablations for the prompt length. We ablate the prompt length for our DCP on the MM-IMDB dataset with missing rate $\eta = 70\%$ in Tab. 8. It's observed that the performance continues to increase when the prompt depth ranges from 12 to 36, and reaches a peak when the prompt depth equals 36. Further increasing thr prompt depth would degrade the performance. We thus set the prompt depth as 36.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: [Yes]

   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: They are discussed at the end of the paper.

   Guidelines:
   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.

- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: They are given in the implementation details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is released.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: They are given in the implementation details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Error bars are not reported due to the high computational costs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: They are included in the implementation details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: [Yes]

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: They are discussed at the end of the paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: [Yes]

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.