

Modular and Parameter-Efficient Multimodal Fusion with Prompting

Sheng Liang, Mengjie Zhao, Hinrich Schütze
Center for Information and Language Processing (CIS)
LMU Munich, Germany
{shengliang, mzhao}@cis.lmu.de

Abstract

Recent research has made impressive progress in large-scale multimodal pre-training. In the context of the rapid growth of model size, it is necessary to seek efficient and flexible methods other than finetuning. In this paper, we propose to use prompt vectors to align the modalities. Our method achieves comparable performance to several other multimodal fusion methods in low-resource settings. We further show that our method is modular and parameter-efficient for processing tasks involving two or more data modalities.

1 Introduction

The success of large-scale pretrained language models (PLMs; Devlin et al. (2019); Yang et al. (2019); Brown et al. (2020); Raffel et al. (2020)) and image encoders (Dosovitskiy et al., 2021; Liu et al., 2021b) has stimulated a surge of pretrained *multimodal models* (Lu et al., 2019; Tan and Bansal, 2019; Radford et al., 2021; Lin et al., 2021) that align text with data in other modalities.

The fast-growing number of parameters in the pretrained models encourages researchers to create more data- and parameter-efficient methods than finetuning (Houlsby et al., 2019; Zhao et al., 2020; Zaken et al., 2021; Li and Liang, 2021; He et al., 2022). Recently, prompting – concatenating manually designed prompt phrases (Schick and Schütze, 2021; Tam et al., 2021; Le Scao and Rush, 2021; Zhao and Schütze, 2021) or trained embedding vectors (Li and Liang, 2021; Lester et al., 2021) to the text input of PLMs – has become an important research direction.

Following this trend, Tsimpoukelli et al. (2021) introduce *Frozen*, successfully extending PLMs into few-shot learners (i.e., models that perform well with only a handful of data) for multimodal tasks, by pretraining a vision encoder whose outputs are prompts fed to the PLM. Frozen performs

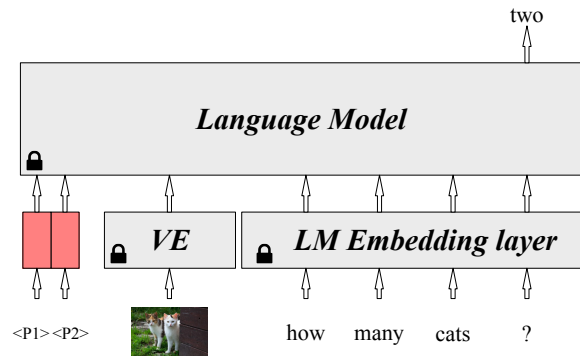


Figure 1: Model architecture. We disentangle VE’s functionality by introducing prompt vectors. The only work of VE is to extract image representations. PLM and VE are fixed (grey) during training; two prompt vectors are the only trainable parameters (red).

strongly on low-resource visual question answering through GPT3-style (Brown et al., 2020) priming (in-context learning). Frozen consists of two components: A vision encoder (VE) (in their case, NF-ResNet-50 (Brock et al., 2021)) and an off-the-shelf PLM like GPT3. When pretraining Frozen, the PLM takes the image representations extracted by VE as prompts, to generate captions describing the input image. PLM parameters are *fixed* and VE is pretrained from scratch. The success of Frozen shows the potential of prompting-based systems for solving multimodal tasks (Zhou et al., 2021; Yang et al., 2021; Salaberria et al., 2021).

One inherent discrepancy between Frozen and prompting for NLP tasks (Li and Liang, 2021; Lester et al., 2021) is that the prompt vectors in Frozen represent part of the input, the image: They are image features extracted by VE. In contrast, prompt vectors in NLP are agnostic to the input texts: They are trainable parameters of the PLM embedding layer to be optimized during training. Recall that the PLM in Frozen is fixed when pretraining VE. This implies that VE’s trainable parameters serve two quite distinct purposes: (i) ex-

tract high quality image representations; (ii) align the image and text representation spaces.

We investigate the efficacy of *disentangling* the functionality of VE. Concretely, we fix the parameters of PLM and VE, and allocate extra free parameters for learning the alignment between spaces of different modalities when conducting a multimodal task; this is achieved by introducing additional prompt vectors. As a result, VE can dedicate itself to extract high quality image representations. We hypothesize that disentanglement has two benefits. First, *higher modularity* is achieved compared to Frozen because VE is freed from the objective of aligning modalities. Higher modularity brings higher flexibility, which is not applicable in systems like Frozen: We can easily change the type of VE, e.g., replacing a CNN with a Transformer; adding extra modalities like speech data is made possible as well. Our architecture meets the desideratum stated by [Srivastava et al. \(2014\)](#): It should be possible to modularly add modalities to an existing multimodal system. Second, higher *parameter efficiency* is achieved by fixing the encoders of different modalities during training; the prompt vectors are the only module to be trained for aligning the representation spaces.

We present **PromptFuse**, a prompting-based approach extending PLMs to multimodal tasks in a modular and efficient manner. Our contributions: (i) We show that the prompting paradigm of utilizing PLMs ([Liu et al., 2021a](#)) effectively strengthens PLMs with the ability of processing data in modalities besides text. With only $\approx 15\text{K}$ trainable parameters, PromptFuse performs comparably to several multimodal fusion methods in low-resource regimes. (ii) We further propose **BlindPrompt**, which enforces that the prompt vectors solely focus on task-specific information and is therefore less prone to overfitting.

2 Related Work

Prompting is a more data- and parameter-efficient method of using pretrained language models (PLMs; [Devlin et al. \(2019\)](#); [Yang et al. \(2019\)](#); [Brown et al. \(2020\)](#); [Raffel et al. \(2020\)](#)) than finetuning ([Devlin et al., 2019](#)). Concretely, [Brown et al. \(2020\)](#), [Schick and Schütze \(2021\)](#), [Tam et al. \(2021\)](#), [Le Scao and Rush \(2021\)](#), and [Gao et al. \(2021\)](#) show that prompting outperforms finetuning in many NLP tasks when annotations are limited, i.e., in *few-shot learning*. [Li and Liang](#)

([2021](#)) introduce prefix-tuning, only updating the prompt vectors, keeping the PLM fixed. [Lester et al. \(2021\)](#) introduce prompt-tuning – a simple form of prefix-tuning – achieving performance comparable to finetuning when scaling up the number of parameters in PLMs. As large PLMs remain unchanged during prefix- and prompt-tuning, high parameter-efficiency is achieved.

Multimodal pretraining. The success of PLMs and pretrained image encoders ([Dosovitskiy et al., 2021](#); [Liu et al., 2021b](#)) encourage fast developments of multimodal pretraining, e.g., large-scale neural networks that align texts with data in other modalities like image ([Tan and Bansal, 2019](#); [Su et al., 2019](#); [Cho et al., 2021](#); [Wang et al., 2021](#); [Kim et al., 2021](#)), video ([Sun et al., 2019](#)) and speech ([Bapna et al., 2021](#)).

Prompting methods for multimodal models were recently devised. [Zhou et al. \(2021\)](#) learn continuous prompt vectors rather than natural language descriptions to model visual concepts. [Yao et al. \(2021\)](#) mark image regions as prompts, adapting pretrained vision-language models to downstream tasks. In Frozen, for a fixed PLM, [Tsimpoukelli et al. \(2021\)](#) pretrain a VE with image captioning where image representations from the VE are used as prompt vectors. The VE in Frozen needs to achieve two objectives: Extracting high quality image representations and properly aligning image/text spaces. In this work, we show that disentangling the two functionalities – instead of pretraining a VE like Frozen, we utilize pretrained VE as feature extractor and train prompt vectors to fuse the modalities – results in a more modular and efficient multimodal system.

3 Prompting as Multimodal Fusing

We propose to decompose the functionality of VE in Frozen into: (i) providing high quality image representations to the PLM; (ii) aligning the image and text spaces for a multimodal task. Achieving (i) is straightforward – we leverage off-the-shelf pretrained image encoders, e.g., Vision Transformer (ViT; [Dosovitskiy et al. \(2021\)](#)). We align the two representation spaces by prompt-tuning ([Li and Liang, 2021](#); [Lester et al., 2021](#)), i.e., by introducing prompt vectors. Concretely, we randomly initialize N trainable vectors in the embedding layer of PLM. When processing downstream multimodal tasks, we *finetune the prompt vectors but fix PLM and VE*. Figure 1 illustrates our model. We call

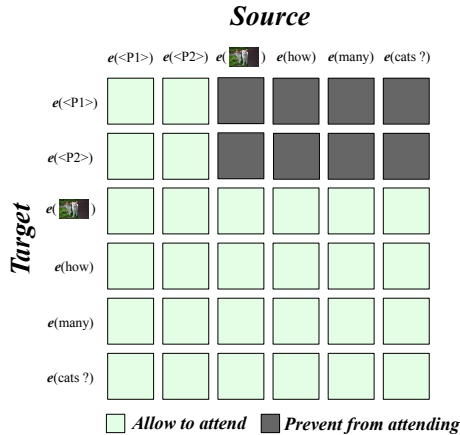


Figure 2: BlindPrompt attention mask in PLM encoder. Prompt vectors cannot attend to the input content, so their parameters solely serve to align the modalities.

our method **PromptFuse**. Having very few trainable parameters, PromptFuse is well suited for low-resource regimes.

We design a special attention mask for the PLM encoder, shown in Figure 2. While the attention of input data remains fully visible, we enforce prompt vectors to only access each other but be blind to the input data. We refer to this variant of PromptFuse as **BlindPrompt**. BlindPrompt fuses data in all modalities using the prompt vectors in self-attention layers. This further emphasizes that prompt vectors should be focusing on the *alignment* between modalities rather than on *specifics* of the content of a modality. As a result, BlindPrompt is more robust to spurious statistical cues (Niven and Kao, 2019). For example, given a picture that dogs run after a man, overfitting systems tend to answer “poodles” in response to the question “What do dogs chase?”.

4 Experiments: Two Modalities

4.1 Setup

Our model is designed to be modular, maximizing the utility of widely used pretrained vision and language models: ViT (Dosovitskiy et al., 2021) as our VE and BART (Lewis et al., 2020) as our PLM. For both models we use the pretrained *base* checkpoints from HuggingFace (Wolf et al., 2020). We use the embedding v of [CLS] as the image representation unless otherwise noted; we use cross-entropy loss during training and use greedy search when decoding.

We experiment with visual question answering (VQAv2; Goyal et al. (2017)), for which un-

derstanding both image and language is necessary when answering a question about an image. VQAv2 consists of 443,757 samples, categorized into three types: *Number*, *Yes/No*, and *Other*.

We simulate low-resource regimes by sampling 128 and 512 shots of training data. We show that PromptFuse and BlindPrompt are less prone to overfitting in low-resource scenarios than baseline methods, in which the model tends to place extra emphasis on samples of the majority answer type *Yes/No* but pays less attention to *Other*. This is because the two answering words of *Yes/No* have much higher frequency in the text corpus than the answers of the open-ended questions, i.e., *Other*.

We train the models for two epochs on the full dataset and 100 epochs on the sampled low-resource datasets. For prompting, we set the prompt length N to 20, and Appendix §A shows an ablation study. Similar to Lester et al. (2021), we empirically found that a large learning rate leads to better prompting performance. So we use learning rate $5e-1$ for prompting; learning rate $5e-4$ is used in all other experiments. Batch size is 32 and the Adam optimizer (Kingma and Ba, 2015) is used.

4.2 Baseline

We consider four baselines of fusing the modalities:

Finetune. As the baseline *Frozen*_{finetuned} in Tsimpoukelli et al. (2021), we finetune *all parameters of VE*, such that the visual embedding space is expected to be aligned with PLM’s language embedding space.

Linear. We fix VE, but train a linear layer to project its output, i.e., the visual embedding, while retaining its dimensionality.

JointProj. We concatenate the visual embedding v to the embedding vector w_i of each (sub)word in the sentence. Next, we train a linear layer to project the concatenated vectors to the PLM hidden dimension. The resulting vectors are input to the PLM encoder layers.

BlackImage. To verify that the prompt vectors use visual information from VE (as opposed to simply conditioning on spurious features of the text, as in the above “poodle” example), we train the prompt vectors with black images.

Table 1 shows the number of trained parameters of the methods. Finetune requires the largest number of trainable parameters, followed by JointProj and Linear; PromptFuse and BlindPrompt are much more parameter-efficient.

Finetune	Linear	JointProj	PromptFuse	BlindPrompt
86M	0.5M	1M	15K	15K

Table 1: Number of trainable parameters of different fusion methods in million (M) and thousand (K).

Full dataset	Other	Yes/No	Number	Overall
Finetune	20.3±0.5	69.3±0.3	29.5±0.2	40.1±0.3
Linear	8.5±0.6	63.9±0.2	23.3±0.3	30.1±0.3
JointProj	19.2±0.4	67.7±0.2	28.9±0.4	38.9±0.1
BlackImage	8.3±0.7	60.4±0.5	15.3±0.4	23.7±0.5
PromptFuse	12.2±0.6	64.9±0.4	27.1±0.2	34.1±0.4
BlindPrompt	13.3±0.9	64.5±0.4	27.4±0.1	34.8±0.8
128 shots				
Finetune	6.6±0.3	57.9±0.9	14.7±0.3	26.8±0.5
Linear	2.3±0.1	46.4±0.7	16.2±0.4	18.2±0.4
JointProj	3.9±0.5	63.3±0.1	19.4±0.6	28.4±0.3
BlackImage	0.9±0.1	38.9±0.8	6.2±0.4	14.4±0.5
PromptFuse	4.9±0.6	63.7±0.3	16.9±0.2	28.3±0.6
BlindPrompt	8.0±1.1	62.1±0.2	19.8±0.3	28.0±0.9
512 shots				
Finetune	7.3±0.3	61.1±0.2	20.2±0.4	29.2±0.3
Linear	4.3±0.4	62.2±0.5	19.2±0.4	26.6±0.4
JointProj	3.8±0.1	63.8±0.3	23.8±0.4	28.7±0.3
BlackImage	3.5±0.6	48.2±0.6	10.3±0.5	18.8±0.5
PromptFuse	6.3±0.5	63.9±0.1	21.5±0.3	29.4±0.5
BlindPrompt	8.4±0.9	63.1±0.2	22.6±0.3	29.7±0.6

Table 2: Results (accuracy) on VQAv2 validation set. We report Overall and separate performance of the three types of questions: Other, Yes/No, Number.

4.3 Results

Table 2 compares the performance of baselines and our prompting methods. We report mean and standard deviation over three runs with different random seeds.

PromptFuse outperforms the BlackImage and Linear baselines on all experiments, showing that prompting successfully utilizes visual information and fuses the two modalities.

For 128 and 512 shots, PromptFuse achieves accuracy comparable with baselines Finetune and JointProj. However, *PromptFuse and BlindPrompt are more parameter-efficient* as shown in Table 1. Prompting methods perform worse than Finetune and JointProj on full data.¹ We conjecture that this is due to having much fewer parameters, i.e., 15K, which is even smaller than the training set size 443,757. Thus we argue that PromptFuse better suits low-resource scenarios.

In low-resource experiments, PromptFuse and BlindPrompt achieve higher accuracy on *Other* and *Number*; the performance drops on *Yes/No* compared with Finetune and JointProj. This also happens between PromptFuse and BlindPrompt. For example, on 128 shots, we find that BlindPrompt

¹Finetune (40.1) performs worse than Frozen_{VQA} (48.4). We hypothesize this is because Frozen uses a much larger PLM (7 billion) than ours (139 million).

outperforms PromptFuse with 3% on *Number* and 3% on *Other*. The results indicate that our prompting methods, especially BlindPrompt, can better utilize the generalization capability of PLM to handle open-ended questions and are less prone to falling into *Yes/No* samples.

4.4 Qualitative Example

To understand how prompting helps in fusing different modalities, we compare PromptFuse and BlindPrompt to a **NoPrompt** baseline. NoPrompt directly concatenates the visual outputs from VE to the text input of the PLM without any training.

Concretely, we apply the **Integrated Gradients** method (Sundararajan et al., 2017), which measures the attribution of features to the neural network outputs. Traditional approaches define feature importance by the gradient of model outputs to input features. Integrated gradients extend this measure as the path integral of the gradient from a baseline – reflecting the absence of signal – to the actual input. In practice, we use the Captum package (Kokhlikyan et al., 2020) in our implementation.

Table 3 illustrates a qualitative example when applying NoPrompt, PromptFuse, and BlindPrompt on VQAv2. For NoPrompt, because no training is involved, visual embeddings from VE confuse the PLM, leading to a wrong prediction (“</s>”). The system is not able to correctly understand the image and question. In contrast, PromptFuse and BlindPrompt guide the PLM to pay attention to the image and identify the regions of “giraffe” and then correctly respond “Yes”.

Interestingly, the attribution scores of the question from BlindPrompt are small, compared to PromptFuse. We conjecture the reason is that, understanding the question – which has a straightforward syntactic/semantic structures – is relatively simple for the PLM because it has been pretrained on a large volume of text. BlindPrompt thus enforces that the multimodal system focus more on the visual embeddings (i.e., the encoded image), which is a new source of information for answering the question.

5 Experiments: Three Modalities

Disentangling functionality of the modality data encoder, e.g., VE, makes PromptFuse and BlindPrompt more modular than Frozen. Applying our methods to tasks involving more than two modali-


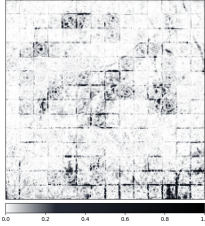
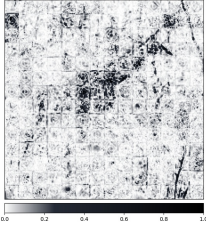
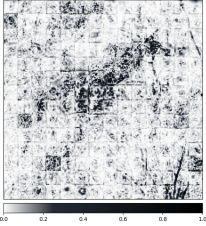
	NoPrompt	PromptFuse	BlindPrompt	
				
Question	Do you see a giraffe in the picture?	Do you see a giraffe in the picture?	Do you see a giraffe in the picture?	
Prediction	</s>	Yes	Yes	

Table 3: Attribution score magnitude heat map for image and text inputs. Black/white image pixels indicate positive/negative influence on predicting “Yes”, and the same goes for red/blue tokens. Integrated gradients are calculated only on the first prediction after decoder input “</s><s>” in an auto-regressive manner.

ties is straightforward. In contrast, Frozen incurs the high cost of pretraining encoders for new modalities. We experiment on the sarcasm detection dataset MUsTARD (Castro et al., 2019) with video, audio, and text data.²

Setup. To process video, we first use OpenFace (Baltrusaitis et al., 2018) to sample important frames containing human faces. Next, ViT is leveraged to extract visual representations from each frame. We then average visual representations of all frames to represent the video. To process audio, we use librosa (McFee et al., 2015) to remove background noise and convert audio to waveform with a sampling rate of 16,000 Hz. We then use pre-trained wav2vec2 (Baevski et al., 2020) to encode the waveform and apply the same averaging strategy as for video. BART is used as our PLM. We use a verbalizer of *True/False* in this experiment.

We adopt the speaker-dependent setup in MUsTARD: 334 training and 356 testing samples. We compare PromptFuse, BlindPrompt, and Finetune for 8, 32, and 64 shots. Note that Finetune uses 180M trainable parameters in the vision and audio encoders. We also conduct an experiment training on the full dataset for 5 epochs. The remaining setup is the same as §4.1.

Results. Table 4 reports performance over ten runs. PromptFuse and BlindPrompt outperform Finetune in 8- and 64-shot experiments. Prompting methods perform comparably to Finetune in other experiments, *while they are clearly more parameter-efficient*. Overall, the three-modality

	Precision	Recall	F-Score
Full dataset			
Finetune	65.6±0.2	73.9±2.7	68.4±0.5
PromptFuse	64.2±0.4	72.1±3.6	66.2±0.7
BlindPrompt	63.8±0.5	71.9±3.1	66.5±0.8
8 shots			
Finetune	42.8±4.3	69.5±9.9	52.7±5.5
PromptFuse	41.1±4.8	71.0±13.1	53.1±5.8
BlindPrompt	44.2±4.5	71.8±12.8	54.0±6.1
32 shots			
Finetune	53.9±4.1	70.6±9.1	59.1±5.2
PromptFuse	53.8±4.7	71.1±10.8	58.5±5.4
BlindPrompt	54.6±4.1	69.7±10.3	58.7±5.5
64 shots			
Finetune	59.5±2.3	70.4±7.7	61.4±2.8
PromptFuse	59.2±2.7	70.2±7.4	62.0±3.3
BlindPrompt	60.1±2.4	70.9±7.8	61.7±3.1

Table 4: Results on MUsTARD test set.

experiment provides observations in line with §4.3. More importantly, it highlights two strengths of prompting: High modularity and parameter-efficiency.

6 Conclusion

We propose PromptFuse and BlindPrompt as methods for aligning different modalities in a modular and parameter-efficient manner. We show that prompting, which requires only a few trainable parameters, performs comparably to several multimodal fusion methods in low-resource scenarios. The high modularity property of prompting supports – by avoiding the need to finetune large pre-trained models – flexible addition of modalities at low cost.

Acknowledgements

This work was supported by the European Research Council (# 740516). We thank the anonymous reviewers for valuable comments.

²To highlight modularity, we utilize pretrained encoders rather than the data preprocessing pipelines in Castro et al. (2019). For example, we use pretrained wav2vec2 (Baevski et al., 2020) rather than Mel-Frequency Cepstral Coefficients (Davis and Mermelstein, 1980) when processing audio data.

References

- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proceedings of the 34th Conference on Neural Information Processing Systems*, volume 33.
- Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. 2018. **OpenFace 2.0: Facial behavior analysis toolkit**. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 59–66.
- Ankur Bapna, Yu-an Chung, Nan Wu, Anmol Gulati, Ye Jia, Jonathan H Clark, Melvin Johnson, Jason Riesa, Alexis Conneau, and Yu Zhang. 2021. SLAM: A unified encoder for speech and language modeling via speech-text joint pre-training. *arXiv preprint arXiv:2110.10329*.
- Andrew Brock, Soham De, Samuel L Smith, and Karen Simonyan. 2021. High-performance large-scale image recognition without normalization. *arXiv preprint arXiv:2102.06171*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. **Language models are few-shot learners**. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards multimodal sarcasm detection (an _obviously_ perfect paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Florence, Italy. Association for Computational Linguistics.
- Jaemin Cho, Jie Lei, Haochen Tan, and M. Bansal. 2021. Unifying vision-and-language tasks via text generation. In *ICML*.
- S. Davis and P. Mermelstein. 1980. **Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences**. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. **An image is worth 16x16 words: Transformers for image recognition at scale**. In *International Conference on Learning Representations*.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. **Making pre-trained language models better few-shot learners**. 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 3816–3830, Online. Association for Computational Linguistics.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. **Towards a unified view of parameter-efficient transfer learning**. In *International Conference on Learning Representations*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. **Long short-term memory**. *Neural Comput.*, 9(8):1735–1780.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.
- Wonjae Kim, Bokyoung Son, and Ildoo Kim. 2021. **Vilt: Vision-and-language transformer without convolution or region supervision**. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5583–5594. PMLR.
- Diederik P. Kingma and Jimmy Ba. 2015. **Adam: A method for stochastic optimization**. In *ICLR*.
- Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. **Captum: A unified and generic model interpretability library for pytorch**.
- Teven Le Scao and Alexander Rush. 2021. **How many data points is a prompt worth?** In *Proceedings of*

- the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2627–2636, Online. Association for Computational Linguistics.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). 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 4582–4597, Online. Association for Computational Linguistics.
- Junyang Lin, Rui Men, An Yang, Chang Zhou, Ming Ding, Yichang Zhang, Peng Wang, Ang Wang, Le Jiang, Xianyan Jia, et al. 2021. [M6: A chinese multimodal pretrainer](#). *arXiv preprint arXiv:2103.00823*.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. [Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing](#). *arXiv preprint arXiv:2107.13586*.
- Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021b. [Swin Transformer: Hierarchical vision transformer using shifted windows](#). *arXiv preprint arXiv:2103.14030*.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. [Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks](#). In *NeurIPS*.
- Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. [librosa: Audio and music signal analysis in python](#). In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25. Citeseer.
- Timothy Niven and Hung-Yu Kao. 2019. [Probing neural network comprehension of natural language arguments](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. [Learning transferable visual models from natural language supervision](#). *arXiv preprint arXiv:2103.00020*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Ander Salaberria, Gorka Azkune, Oier Lopez de Lacalle, Aitor Soroa, and Eneko Agirre. 2021. [Image captioning for effective use of language models in knowledge-based visual question answering](#). *arXiv preprint arXiv:2109.08029*.
- Timo Schick and Hinrich Schütze. 2021. [It’s not just size that matters: Small language models are also few-shot learners](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.
- Nitish Srivastava, Ruslan Salakhutdinov, et al. 2014. [Multimodal learning with deep boltzmann machines](#). *J. Mach. Learn. Res.*, 15(1):2949–2980.
- Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. [VL-BERT: Pre-training of generic visual-linguistic representations](#). *arXiv preprint arXiv:1908.08530*.
- Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. [VideoBERT: A joint model for video and language representation learning](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. [Axiomatic attribution for deep networks](#). In *International Conference on Machine Learning*, pages 3319–3328. PMLR.
- Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. [Improving and simplifying pattern exploiting training](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Hao Tan and Mohit Bansal. 2019. [LXMERT: Learning cross-modality encoder representations from transformers](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

- Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. Es-lami, Oriol Vinyals, and Felix Hill. 2021. Multi-modal few-shot learning with frozen language models. *ArXiv*, abs/2106.13884.
- Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021. SimVLM: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. **Transformers: State-of-the-art natural language processing**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. 2021. An empirical study of GPT-3 for few-shot knowledge-based VQA. *arXiv preprint arXiv:2109.05014*.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. **XLNet: Generalized autoregressive pretraining for language understanding**. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2021. CPT: Colorful prompt tuning for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*.
- Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. **BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models**. *CoRR*, abs/2106.10199.
- Mengjie Zhao, Tao Lin, Fei Mi, Martin Jaggi, and Hinrich Schütze. 2020. **Masking as an efficient alternative to finetuning for pretrained language models**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2226–2241, Online. Association for Computational Linguistics.
- Mengjie Zhao and Hinrich Schütze. 2021. **Discrete and soft prompting for multilingual models**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8547–8555, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2021. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*.

A Ablation Analysis

As an ablation analysis, we test variants of PromptFuse and BlindPrompt with full data on VQAv2 dataset. All experiment setup follows §4.1.

Prompt length. PromptFuse and BlindPrompt have an extremely limited number of trainable parameters, making it challenging to achieve performance as finetuning in high-resource scenarios. Intuitively, we would like to inject more prompt vectors to increase the number of trainable parameters. Table 5 shows that both PromptFuse and BlindPrompt obtain best accuracy when the prompt length is set to 60. Using a particularly large length (e.g., 100) harms performance. This is in line with Lester et al. (2021): They find that too much prompt information may bring negative effects. Since more prompt vectors also consume more training time, we use 20 in our experiments.

	5	10	20	40	60	80	100
PromptFuse	28.5	30.4	34.1	35.3	35.8	34.2	30.3
BlindPrompt	27.1	30.7	34.8	35.5	35.6	34.4	30.9

Table 5: Overall accuracy on VQAv2 validation set with prompt length ranging from 5 to 100. We report mean performance over three random seeds.

Prompt position. In this work we inject prompt vectors at the beginning of input fed to PLM (see Figure 1), here we test two alternative positions for injection: (i) middle, i.e., inserting between vision and (sub)word embeddings; (ii) end of the question. Results in Table 6 show that these positions yield similar performance, indicating that our approach is not largely affected by prompt positions.

Prompt encoder. Another approach to increase trainable parameters is to use an extra module to encode prompt vectors. We test two neural network modules: (i) a linear layer; (ii) an LSTM (Hochreiter and Schmidhuber, 1997). Both modules have the same hidden dimension as the PLM. However, these variants only bring small improvements, as presented in Table 6. Future work may explore more advanced methods of scaling up the number of parameters.

Visual embedding. In addition to utilizing the [CLS] embedding, there are two alternative ViT outputs can be used as the visual embeddings: (i) the entire embedded sequence; (ii) the embedding averaged over the sequence. Table 6 shows that these approaches achieve comparable results. To save computational resources, we use [CLS] for

		PromptFuse	BlindPrompt
	Baseline	34.1±0.4	34.8±0.8
Prompt Position	Middle End	33.7±0.4 34.3±0.5	34.9±0.7 34.5±0.6
Prompt Encoder	Linear LSTM	34.7±0.5 34.9±0.4	35.0±0.6 35.1±0.4
Visual Embedding	Seq Avg	34.6±0.6 33.9±0.5	34.7±0.5 34.9±0.4

Table 6: Results on VQAv2 validation set with variants of prompt position, encoder, and visual embedding.

	Other	Yes/No	Number	Overall
BART				
PromptFuse	12.2±0.6	64.9±0.4	27.1±0.2	34.1±0.4
BlindPrompt	13.3±0.9	64.5±0.4	27.4±0.1	34.8±0.8
BERT				
PromptFuse	-	67.5±0.3	28.4±0.2	-
BlindPrompt	-	67.8±0.4	28.6±0.2	-
T5				
PromptFuse	15.8±0.7	65.4±0.2	27.3±0.3	36.5±0.4
BlindPrompt	16.2±0.8	65.2±0.3	27.4±0.2	36.6±0.6

Table 7: Results with BERT and T5 on VQAv2 validation set.

images in VQAv2. For video frames and speech signals in MUSTARD, we use average due to large sequence lengths.

B Modularity

This section further demonstrates the modularity and flexibility of PromptFuse and BlindPrompt. Besides the ability of utilizing encoders of more than two modalities as shown in §5, the modular design allows PromptFuse and BlindPrompt to use PLMs other than BART. Concretely, we compare BERT/T5 to BART, by full data training on VQAv2 as §4.1. BERT is a masked language model, thus we train and evaluate only on *Number* and *Yes/No* samples, by filling the mask in pattern “Question: *input question* Answer: [MASK]”.

As reported in Table 7, BERT performs well on *Number* and *Yes/No* compared to BART, indicating that PromptFuse/BlindPrompt can also be applied to encoder-only architecture. Also, T5 outperforms BART, especially on *Other*, further indicating that PromptFuse/BlindPrompt are compatible with new PLMs, which give increasingly better task performance.

C Experiment Setup

Table 8 shows the setup used in all of our experiments. We use 8 GEFORCE GTX 1080Ti GPUs and gradient accumulation is applied during training.

Dataset	Modalities	# Train	# Test	Runs	Batch Size	Epochs	Prompt Length	LR (Prompt)	LR (Other)
VQAv2 low-resource	Image, Text	443,757	214,354	3	32	2	20	5e-1	5e-4
	Image, Text	128/512	214,354	3	32	100	20	5e-1	5e-4
MUStARD low-resource	Video, Audio, Text	334	356	10	8	5	20	5e-1	5e-4
	Video, Audio, Text	8/32/64	356	10	8	50	20	5e-1	5e-4

Table 8: Dataset statistics and hyperparameters used in the experiments.