Querying as Prompt: Parameter-Efficient Learning for Multimodal Language Model

Supplementary Material

1. Supplementary

1.1. Implementation Details

Answer embedding module. Following [4], we use the task-specific answer classification head and embed the proposal answers to frozen embeddings. In the classification head, the mask token is mapped to an actual answer prediction in the set of possible answers. For Music-AVQA dataset [2], following [4], we use the tokenizer and language model's word embeddings to embed the proposed 42 answers anb store them in the model's parameters. For VideoQA dataset (How2QA [3] and TVQA [1]), we concat the proposal answer with the corresponding question as the text input and set the answer embedding with the embeddings of "Yes" and "No". For CMU-MOSEI dataset [5], which requires an float number output, we just adopt a regression head to predict the answer without answer embedding.

Input prompt engineering. Following [4], we set the text prompt for downstream tasks. Specifically, for VideoQA task, we design the following prompt:

``[CLS] Question: <Question>? Answer: [MASK]. Subtitles: <Subtitles> [SEP]''

for AVQA task, we design the following prompt:

``[CLS] Question: <Question>? Answer: [MASK]. [SEP]''

for MSA task, we design the following prompt:

``[CLS] Uttrance: text Sentiment score [-3, 3]. [SEP]''

Due to the utilization of a regression head in the MSA task, wherein emotion polarity is directly regressed from the [CLS] token, we have deliberately refrained from incorporating the [MASK] token within the prompt.

Evaluation metrics. For Music-AVQA, How2QA and TVQA, we follow previous works [2, 4] and adopt the prediction accuracy as the metric:

$$\text{Accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i) \tag{1}$$

where $\hat{y}_i$ is the predicted value of the i-th sample and y_i is the corresponding true value.

For CMU-MOSEI dataset, we adopt the mean absolute error (MAE), Pearson correlation (Corr), binary classification accuracy (ACC-2) and F1 score as evaluation metrics.

The MAE is defined as:

$$\text{MAE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i| \tag{2}$$

The Corr is defined as:

$$\text{Corr}(y, \hat{y}) = \frac{cov(y, \hat{y})}{std(y) * std(\hat{y})} \tag{3}$$

where $\hat{y}$ is the predicted value of all samples, and y is the corresponding true value. cov represents the covariance function and the std represents the standard deviation function.

The F1 score is defined as:

$$\text{F1}(y, \hat{y}) = 2\frac{P(y, \hat{y})R(y, \hat{y})}{P(y, \hat{y}) + R(y, \hat{y})} \tag{4}$$

where $P(y, \hat{y})$ represents the precision function and $R(y, \hat{y})$ represents the recall function.

Because the MOSEI dataset requires the prediction of sentiment polarity, we treat it as a binary classification task with a zero bound when computing the classification-related metric. Futher more, the classification evaluation metrics for the dataset are divided into two components, namely Non-negative/Negative (Left) and Positive/Negative (Right). Non-negative/Negative (Left) refers to categorizing sentences with sentiment intensities of 0 or positive values as non-negative class, and sentences with negative sentiment intensities as negative class, followed by the computation of classification metrics. Positive/Negative (Right) involves categorizing sentences with positive sentiment intensities as positive class, and sentences with sentiment intensities of 0 or negative values as negative class, followed by the computation of classification metrics.

1.2. Computation and parameter efficiency.

Parameter efficieney. In our study, we compared the parameter counts of the Text Conditioned Resampler (TCR) when employing only a single linear layer with that of utilizing a conventional Multi-Head Attention (MHA) mechanism. As shown in Table 1, initially, we calculated the parameter count when solely using FFN adapter during finetuning. Subsequently, we computed the parameter count when employing a regular Attention layer in the TCR, as indicated in the second row of Table 1. Consequently, we obtained the parameter counts for the TCR utilizing only one linear layer for each modality, revealing a parameter reduction of 90%.

Method	**Trainable Params↓**	**Extra Params↓**
only FFN Adapter	18.7M	-
Normal Attention TCR	245M	226.3M
One Linear TCR (Ours)	40M	**21.3M**

Table 1. Comparison of the trained parameters.

Method	**MACs(G)↓**	**Extra MACs(G)↓**
only Text	200.01	-
Concat Input	319.47	119.46
Querying as Prompt (Ours)	216.93	**16.92**

Table 2. Comparison of the computational cost.

Input Modality	**Accuracy↑**
only Text	55.24%
Audio+Text	68.99%
Visual+Text	76.93%
Audio+Visual+Text	**78.41%**

Table 3. Comparison of the input modalities.

Computation efficiency. We conducted a comparison between our approach and the computational cost incurred by directly concatenating Visual, Audio, and text inputs. Assuming a text token count of 23, and both Visual and Audio token counts at 10 each, the direct concatenation results in a language model input token count of 43. Conversely, employing our Querying as Prompt method leads to an input token count of 25, as illustrated in the table 2. Initially, we computed the computational cost when only text is input, yielding a value of 200.01 GMAC. Subsequently, we separately calculated the computational costs for Concatenation input and Querying as Prompt, subtracting the cost of text-only input to determine the additional computational load introduced by the other modalities. Notably, our approach incurs only 14% of the additional computational cost compared to the direct concatenation input method.

1.3. Extra experiments

Input modality ablation. We conducted a comparative analysis on the AVQA dataset, examining the outcomes derived from different modalities of input. As shown in Table 3, it is evident that the performance is notably inferior when considering only the text modality. However, with the inclusion of each additional input modality, there is a corresponding improvement in performance, indicating the utility of information from each modality. Notably, the Visual modality contributes the most substantial enhancement in accuracy.

Input Frames	10	20	30
Accuracy	78.41%	78.55%	78.75%

Table 4. The results of our method on AVQA datasets with different numbers of input video frames.

Method	Flamingo	Q-Former	Ours
Acc.↑/ Params.↓	68.5%/ 1.7B	53.93%/ 335M	**78.41%/ 40M**

Table 5. The results of reimplemented Flamingo and Q-Former on AVQA dataset.

Input video frame ablation. Table 4 shows the results of our method with different input lengths of video frames and is observed that increasing input length can even slightly boost the performance, which reveals that our method can effectively handle the long inputs of video/audio modality. For a fair comparison with previous works, we reported the result with 10 frames' input in the main text.

Comparison with Flamingo and Q-Transformer. We supplement the following results of fine-tuning Flamingo and Q-Former on the AVQA dataset in Table 5. Both models are difficult to converge due to the significantly increased scale of training parameters or inputs, leading to the poor performances on the AVQA task.

References

[1] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*, 2018. 1

[2] Guangyao Li, Yake Wei, Yapeng Tian, Chenliang Xu, Ji-Rong Wen, and Di Hu. Learning to answer questions in dynamic audio-visual scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19108–19118, 2022. 1

[3] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. *arXiv preprint arXiv:2005.00200*, 2020. 1

[4] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *Advances in Neural Information Processing Systems*, 35:124–141, 2022. 1

[5] AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2236–2246, 2018. 1