Mimic In-Context Learning for Multimodal Tasks

Supplementary Material

7. Implementation Details

7.1. Prompts

Following [18], we use the same prompts for both Idefics1 and Idefics2, as shown in Tab. 6.

7.2. Hyperparameters

For all datasets, the hyperparameters for all trainable methods are as outlined in Tab. 7. When the training set size is less than 1000, we perform training for 10 epochs, and when the training set size is 1000, we perform training for 5 epochs. For LIVE, we follow the recommendations from the original paper and train for 10 epochs when the training set size is 1000 [35]. For both LIVE and MimIC, we use the same learning rate of 5×10^{-3}, and the learning rate for the shift magnitude in LIVE is set to 1×10^{-2}, in accordance with the original paper.

For LoRA, we set the rank $r = 16$ and modified $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, $\boldsymbol{W}_v$, and $\boldsymbol{W}_o$ in all attention layers of both the vision and language models. Given the substantial number of parameters in LoRA, we set the learning rate to 5×10^{-4} to ensure stable training. In Sec. 9.3, we introduce an alternative parameter setting for LoRA, denoted as LoRA†. Specifically, LoRA† modifies only $\boldsymbol{W}_o$ in the language model, with a rank of $r = 1$. This configuration is the most similar to MimIC, not only in terms of the number of parameters but also in the modification of the input to the feed-forward network (FFN) layer. For both configurations, the dropout rate is set to 0.05, and the LoRA scaling factor α is set to $2r$.

8. Exploratory Experiments

8.1. Where to Align?

In Sec. 3.2, we show that MimIC uses the output of the FFN layer in each decoder layer of both the original LMM and the MimIC LMM to compute $\mathcal{L}_{\text{align}}$, aiming to align zero-shot and ICL. However, using the self-attention output to compute $\mathcal{L}_{\text{align}}$ is also reasonable, as Eq. (2) only requires that the shift vector should be added after the self-attention. Therefore, on Idefics1, we evaluate two settings: (1) **After SA**: using the hidden states from the self-attention output; (2) **After FFN**: using the hidden states from the FFN output, which is adopted by MimIC. The results are presented in Fig. 7. We find that while "After SA" converges faster, its performance is inferior to "After FFN", especially when the training set size is small. This may be because the FFN amplifies the errors in the attention output, making it easier for the MimIC attention head to overfit and leading to poorer generalization performance.

8.2. Implementations of the Shift Vector

In Sec. 3.1, we decompose the single-head self-attention (SA) for each query in ICL into the following components: standard attention $SA(\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V})$, shift magnitude μ, and the attention difference term $\text{SA}(\boldsymbol{q}, \boldsymbol{K}_D, \boldsymbol{V}_D) - \text{SA}(\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V})$. Moreover, Eq. (2) can be rewritten as:

$$
\begin{aligned}
&\text{SA}\left(\boldsymbol{q}, \begin{bmatrix}\boldsymbol{K}_D \\ \boldsymbol{K}\end{bmatrix}, \begin{bmatrix}\boldsymbol{V}_D \\ \boldsymbol{V}\end{bmatrix}\right) \\
&= \text{SA}(\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V}) + \mu \left(\text{SA}(\boldsymbol{q}, \boldsymbol{K}_D, \boldsymbol{V}_D) - \text{SA}(\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V})\right) \\
&= (1-\mu) \underbrace{\text{SA}(\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V})}_{\text{standard attention}} + \mu \underbrace{\text{SA}(\boldsymbol{q}, \boldsymbol{K}_D, \boldsymbol{V}_D)}_{\text{attention over ICDs}}
\end{aligned}
\tag{6}
$$

Note that the attention over ICDs in the second term depends solely on ICDs and is independent of other query tokens. Therefore, we can approximate $\text{SA}(\boldsymbol{q}, \boldsymbol{K}_D, \boldsymbol{V}_D)$ using a network, *i.e.*, $h(q) := \text{SA}(\boldsymbol{q}, \boldsymbol{K}_D, \boldsymbol{V}_D)$. For simplicity, we train a linear layer $h : \mathbb{R}^{d_h} \to \mathbb{R}^{d_h}$. The results, shown in Tab. 8, indicate that using a linear layer to implement the shift vector performs significantly worse than MimIC. This may be due to the query-dependent shift being more sensitive to noise from different ICD configurations, making it less robust than using a query-independent learnable vector. To verify this, we replace the linear layer h with the learnable vector, and observe a substantial improvement in performance.

9. Additional Results

9.1. Training with Scaling Data

In Sec. 4.2, we evaluated the performance of MimIC on 1000 data samples and also examined the effect of reducing the number of training samples. However, we did not investigate how MimIC performs when scaling up the dataset. A limited number of samples may not be sufficient for MimIC to reach optimal performance. Therefore, we further validated its performance on 8000 samples.

As shown in Tab. 9, MimIC consistently outperforms LoRA in most cases and remains ahead of LIVE. While its performance improves with a larger dataset, the performance gap between MimIC and LoRA narrows and, in some cases, is even reversed. This phenomenon occurs because MimIC is trained by simulating in-context learning (ICL), which imposes inherent constraints on its upper performance limit. MimIC was designed for the low-data regime; however, if this limitation can be mitigated, MimIC has the potential to become a highly efficient method for

Task	Prefix prompt	ICD prompt	Stop words
VQAv2 OK-VQA	Instruction: provide an answer to the question. Use the image to answer.	Image:{image} Question: {question} Answer: {answer}\n	"Question", "Answer","Image"
COCO	✗	Image: {image} Caption: {caption}\n	"Caption", "Image"
COCO ICL	Instruction: provide a short caption of the input image.\n		

Table 6. The prompt templates on different tasks evaluated in our paper. The data to be replaced is between curly brackets.

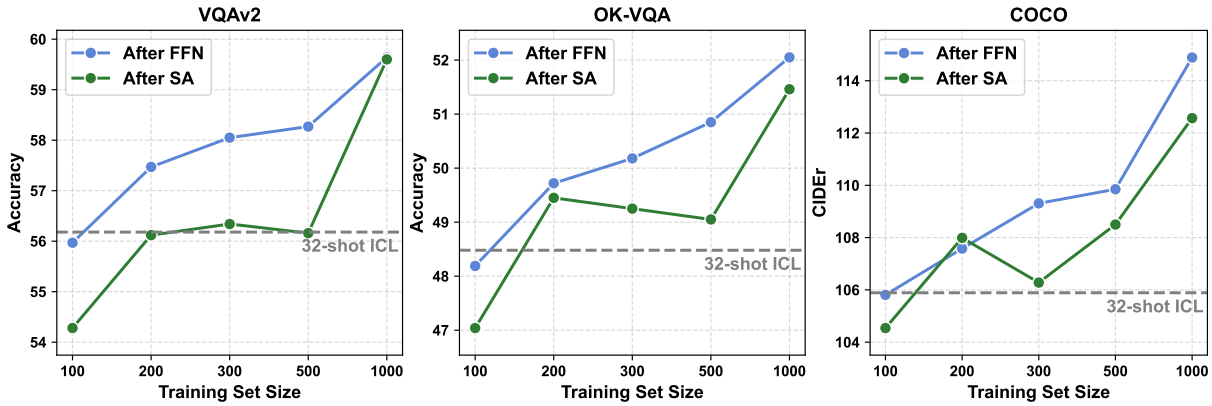

Figure 7. Performance of MimIC trained with different alignment strategy.

Hyperparameters	Value
optimizer	AdamW
warmup step ratio	0.1
precision	float16
weight decay	1e-3
batch size	2
accumulate gradient batches	2

Table 7. The common hyperparameters used in training for all trainable methods over all datasets on both Idefics1 and Idefics2.

Method	VQAv2	OK-VQA	COCO
Linear layer	47.68	42.61	112.88
Learnable vector	58.84	51.13	113.07
MimIC	**59.64**	**52.05**	**114.89**

Table 8. Performance comparison among different implementations of the shift vector.

Model	Method	VQAv2	OK-VQA	COCO
Idefics-9b	Zero-shot	29.25	30.54	63.06
	32-shot ICL	56.18	48.48	105.89
	LIVE	58.54*	50.08*	<u>117.38</u>*
	LoRA	<u>59.04</u>	<u>53.15</u>	110.37
	MimIC	**60.2**	**53.84**	**118.07**
Idefics2-8b-base	Zero-shot	55.39	43.08	40.00
	8-shot ICL	66.20	57.68	122.51
	LIVE	70.30*	58.52*	-
	LoRA	**75.24**	**64.26**	133.8
	MimIC	<u>72.85</u>	<u>61.76</u>	**133.98**

Table 9. The results of VQAv2, OK-VQA, and COCO on Idefics-9b and Idefics2-8b-base trained on 8000 samples. The weight of alignment loss is set to 0.7. Numbers marked with an asterisk (*), in **bold**, or <u>underlined</u> represent results reported in the original paper, the best results, and the second-best results, respectively.

parameter-efficient fine-tuning.

9.2. Generalize to more Tasks

We evaluated MimIC on four new tasks: 1) **Flickr30k** [56]: A large image-caption dataset consisting of 31,000 images, each paired with five descriptive captions. 2) **MME** [8]: A comprehensive benchmark designed to evaluate the performance of LMMs across 14 subtasks, assessing both perceptual and cognitive abilities. 3) **SEED-bench** [20]: A novel benchmark comprising 24,000 multiple-choice questions with precise human annotations, covering 27 evaluation

dimensions. 4) **MMMU-Pro** [57]: An advanced benchmark for evaluating large language models across multiple disciplines, featuring over 12,000 complex multiple-choice questions with ten options each, spanning 14 subjects such as mathematics, physics, and law. The results are presented in Tab. 10, which indicate that despite the increased difficulty of these benchmarks, our method consistently outperforms both LoRA and many-shot ICL in most cases.

Notably, due to computational resource constraints, we could only apply 2-shot ICL on MME, SEED-bench, and MMMU-Pro and were unable to train LoRA. In contrast, thanks to MimIC's lightweight nature, it can still be successfully trained. Additionally, since training does not require storing the KV cache, we can even train MimIC using more-shot ICL. For instance, we can train MimIC with 16-shot on VQAv2, OK-VQA, and COCO, whereas ICL inference is limited to a maximum of 8-shot. This unique advantage also enables applications in scenarios where 1-shot ICL is not feasible.

Model	Method	Flickr30k	MME	SEED	MMMU-Pro
Idefics-9b	Zero-shot	49.17	55.36	27.56	26.10
	ICL	63.41	52.11	28.30	28.14[16]
	LoRA	72.79	60.53	26.95	27.74
	MimIC	**74.03**	**63.06**	**29.89**	**31.38**[16]
Idefics2-8b-base	Zero-shot	53.04	74.80	12.91	28.92
	ICL	84.57	71.10[2]	**47.9**[2]	**32.60**[2]
	LoRA	73.03	-	-	-
	MimIC	**91.77**	**80.83**[2]	47.00[2]	31.73[2]

Table 10. Results evaluated on more tasks. The actual number of shots used is indicated in superscript. For cases without annotations, it is consistent with the description of Sec. 4.2.

9.3. Effectiveness of MimIC in Alignment Effect

Our quantitative analysis in Sec. 4.3 demonstrates that the MimIC attention head and $\mathcal{L}_{\text{align}}$ outperform the shift vector and KL divergence used in LIVE, but no comparison has been made with LoRA. It's natural to ask: whether LoRA can mimic in-context learning in the MimIC framework, as both LoRA and MimIC add a small number of trainable parameters to LMMs.

To address this, similar to Sec. 4.3, we compute the average L2 distance of the latent representations of the first answer token at each layer, with or without $\mathcal{L}_{\text{align}}$ on LoRA, compared to the 32-shot ICL. For a more intuitive comparison, we also evaluate a modified LoRA setting, denoted as LoRA†, which only modifies the output matrix W_o of self attention layers in the language model with a rank $r = 1$. This setting ensures that the number of parameters in LoRA† matches that of MimIC.

	Method	VQAv2	OK-VQA
	Zero-shot	42.97	41.21
without $\mathcal{L}_{\text{align}}$	LoRA†	54.67	48.19
	LoRA	61.18	47.18
with $\mathcal{L}_{\text{align}}$	LoRA†	37.32	36.77
	LoRA	31.89	30.18
	MimIC	**30.17**	**28.24**

Table 11. Comparison of L2 distances between different LoRA settings and 32-shot ICL, and between MimIC and 32-shot ICL. LoRA† is a modified LoRA setting, which only modifies the output matrix W_o of self attention layers in the language model with a rank $r = 1$.

The results, presented in Tab. 11, show that, regardless of whether LoRA has more parameters or the same number of parameters as MimIC, the distance from 32-shot ICL remains greater after training with $\mathcal{L}_{\text{align}}$. This indicates that the efficient design of the MimIC attention head allows it to more effectively mimic ICL.

9.4. Marrying MimIC with LoRA

Although MimIC is highly efficient, its limited number of parameters may restrict its capacity to learn more complex patterns. Fortunately, it is compatible with certain parameter efficient fine-tuning (PEFT) methods, making it possible to combine MimIC with various PEFT techniques. Here, we investigate the performance of integrating MimIC with LoRA. The results, shown in Tab. 12, indicate that adding LoRA leads to performance improvements for MimIC across three datasets, requiring only one epoch of training. This not only highlights the exceptional learning efficiency of MimIC but also suggests that using ICL as a guiding mechanism can enhance the adaptability of fine-tuning methods in few data scenarios. Furthermore, this combination may help bridge the performance gap between ICL and fine-tuning, allowing the two approaches to overcome the potential limitations of each [33].

Method	VQAv2	OK-VQA	COCO
LoRA	55.60	47.06	97.75
MimIC	59.64	52.05	114.89
MimIC + LoRA	**61.04**	**53.84**	**117.08**

Table 12. Performance of MimIC integrated with LoRA.