HiFICL: High-Fidelity In-Context Learning for Multimodal Tasks

Supplementary Material

6. Implementation Details

6.1. Prompts

We use unified prompt templates for VQAv2, OK-VQA, and COCO to isolate the effect of the adaptation method from prompt engineering. For VQA tasks we adopt an instruction-style prefix that asks the model to answer a question given an image; for COCO we use a caption-style prefix.

For all ICL settings (0-shot, 8-shot, LoRA, LIVE, MimIC, and HiFiICL), the same templates are used for every in-context demonstration; only the number of demonstrations and their ordering differ. Tab. 4 lists the exact prefixes, ICL patterns, and decoding stop words for each task, matching the configurations used in the main paper.

6.2. Hyperparameters

Our hyperparameter settings for all trainable methods are designed for a fair and rigorous comparison, strictly following the configurations from their respective original papers where applicable. A summary of the main hyperparameters is provided in Tab. 5.

For our method, **HiFICL**, and for **MimIC** [19], we adopt the same core training configuration. We use an AdamW optimizer [31] with a learning rate of 5e-3, coupled with a cosine annealing scheduler with a 10% warmup phase.

For **LIVE** [36], the base learning rate is also 5e-3. However, in accordance with the original paper, the separate learning rate for its shift magnitude parameter is set to 1e-2.

For **LoRA** [17], we set the rank to $r = 16$. Given its substantial parameter count, we use a lower learning rate of 5e-4 to ensure stable training.

For all methods, when the training set size is 1000, we perform training for 5 epochs. For the smaller datasets in our data efficiency analysis, this is increased to 10 epochs. The task-specific ranks for HiFICL are detailed in Tab. 6.

7. Backbone Models and Qualitative Analysis

7.1. Why Idefics2 and LLaVA-Interleave?

We selected two open-source LMMs for our experiments:

- **Idefics2-8b-base:** An official open-source model from Hugging Face, serving as a standard and reproducible testbed for multimodal research. Its architecture is a clean, fully autoregressive design.
- **LLaVA-Interleave-7b:** A model from the LLaVA-NeXT series, representing one of the most mainstream and widely-used families of open-source LMMs. It is also one of the few LLaVA models that natively supports the interleaved image-text inputs required for ICL studies.

The choice of these two models allows for a robust evaluation of HiFICL's generalization. Idefics2 is a pre-trained backbone, ideal for analyzing fundamental ICL behavior. LLaVA-Interleave is an instruction-tuned model, representing a different and highly optimized training paradigm. Our results also show they have different performance profiles: Idefics2 performs better on VQAv2 and COCO, while LLaVA-Interleave excels on OK-VQA. Demonstrating strong performance on both validates HiFICL's broad applicability.

Practically, both 7-8B scale models fit on a single 80GB A100 GPU for all experimental conditions, including ICL, LoRA, and all approximation methods. This ensures a controlled and fair comparison environment by eliminating the complexities of multi-GPU setups.

7.2. Why Not Other Backbone Models?

To further justify our model selection, we also evaluated HiFICL on three alternative backbones. The findings, summarized in Tab. 7, reinforce our choice of Idefics2 and LLaVA-Interleave for the main experiments.

On the **Idefics1-9B**, which utilizes a cross-attention architecture, HiFICL does not show a clear advantage over MimIC. This is consistent with our theoretical framework, as our method's design is grounded in the mathematical properties of fully autoregressive self-attention mechanisms.

The case of **LLaVA-v1.6-Mistral-7B** is particularly insightful. This model is not natively designed for ICL, and standard 8-shot ICL degrades its performance. However, both HiFICL and MimIC successfully adapt the model and significantly improve over the zero-shot baseline, demonstrating the robustness of ICL approximation methods in this scenario.

Finally, while **LLaVA-OneVision-Qwen2-7B** shows better performance than LLaVA-Interleave, its design for long video sequences results in a doubled GPU memory requirement due to a larger hidden state size. Given that both models are based on the same Qwen2 language model and their performance on our image-based tasks is largely comparable, we selected LLaVA-Interleave for our main experiments. This choice prioritizes a more favorable balance between performance and computational cost, ensuring better reproducibility for the research community.

Table 4. Prompt templates used in our experiments. Curly brackets {} indicate fields filled with instance-specific content.

Task	Prefix prompt	ICL 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 5. Main hyperparameters for all trainable methods. Specific learning rates for LoRA and LIVE's magnitude parameter are noted in the text.

Hyperparameter	Value
Optimizer	AdamW
Base Learning Rate	5e-3
LR Schedule	Cosine w/ 10% Warmup
Weight Decay	0.05
Precision	16-mixed
Batch Size (per GPU)	2
Gradient Accumulation	2
Total Training Epochs	5 (for 1k) / 10 (<1k)

Table 6. The optimal rank r used in HiFICL for each dataset and backbone model.

Dataset	LLaVA-Interleave	Idefics2
VQAv2	8	8
OK-VQA	4	16
COCO	8	4

7.3. Additional Qualitative Examples

We provide qualitative examples on VQAv2 and COCO in Fig. 8 to complement our quantitative results. The examples, generated using the LLaVA-Interleave-7b model, visually demonstrate HiFICL's ability to produce more faithful responses. For instance, HiFICL often corrects factual errors made by other methods (e.g., identifying a "fire engine" instead of a "car") and reduces object hallucination (e.g., avoiding the erroneous "parking meters"), showcasing its effectiveness in capturing nuanced visual details.

8. LoRA Variants and Efficiency Analysis

8.1. Overview of Recent LoRA Variants

LoRA-style PEFT has evolved rapidly. As shown in Figure 9, variants like MoE-based LoRA or FlyLoRA innovate the low-rank update matrix (ΔW). However, they all operate in the static **weight space** via additive updates.

Table 7. Performance comparison across different backbone models. Best results in each group are **bolded**, second best are underlined.

Backbone	Method	VQAv2	OK-VQA	COCO
Idefics1-9B	Zero-shot	29.25	30.54	63.06
	32-shot ICL	56.18	48.48	105.89
	MimIC	59.64	**52.05**	114.89
	HiFICL	**59.71**	51.93	**115.21**
LLaVA-v1.6 (Mistral-7B)	Zero-shot	70.00	63.00	0.7157
	8-shot ICL	68.00	56.00	0.6678
	MimIC	71.24	64.62	1.2857
	HiFICL	**74.71**	**66.88**	**1.3192**
LLaVA-OneVision (Qwen2-7B)	Zero-shot	71.75	48.19	1.2091
	8-shot ICL	78.70	66.59	1.3457
	MimIC	81.22	69.43	1.4312
	HiFICL	**82.39**	**73.12**	**1.4784**

HiFICL shifts this paradigm: instead of modifying the weight space, it operates in the dynamic **activation space** to directly model the ICL-induced shift. Because HiFICL also employs low-rank virtual matrices, existing LoRA structural innovations are potentially complementary. For instance, a future "MoE-HiFICL" could dynamically route to multiple virtual key-value pairs, opening new avenues for context-aware PEFT.

8.2. Efficiency Comparison: HiFICL vs. LoRA

While both HiFICL and LoRA are parameter-efficient, their distinct mechanisms lead to fundamental differences in training efficiency. LoRA's updates are applied to the core weight matrices (W_q, W_k, etc.). To compute gradients for its adapter matrices (A, B), it requires a full backpropagation pass through the entire frozen backbone, making the process computationally intensive. In contrast, HiFICL injects its parameters into the activation space. This architectural choice enables a more localized and efficient gradient computation, as the backpropagation path for the trainable virtual key-value pairs does not involve the main backbone. As empirically verified in Figure 10, this principled design

Figure 8. Qualitative examples on VQAv2 (left) and COCO Captioning (right) using the LLaVA-Interleave-7b model.

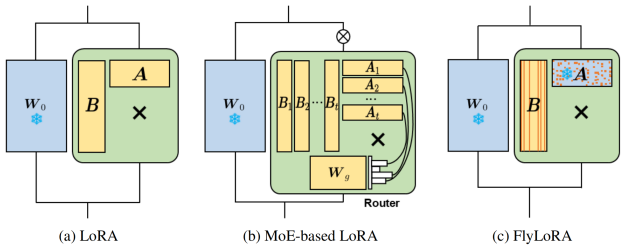

Figure 9. Schematic illustrations of different LoRA variants. (a) Standard LoRA applies a dense low-rank update using matrices A and B. (b) MoE-based LoRA decomposes the update into multiple smaller "expert" pairs $\{A_i, B_i\}$ and uses a router to select a sparse combination. (c) FlyLoRA utilizes a frozen random matrix for A and only trains a sparse matrix B, activating a small subset of its columns for each input.

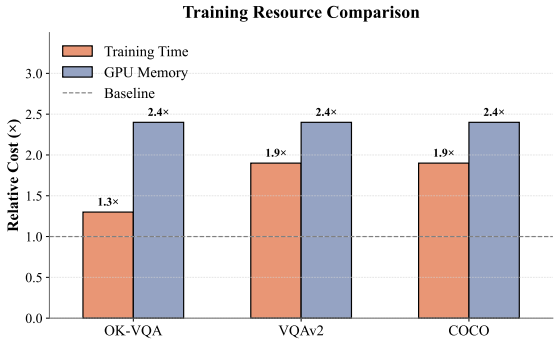

Figure 10. Training efficiency comparison between LoRA and HiFICL on the Idefics2 model. Costs (GPU Memory, Training Time, FLOPs) are relative to HiFICL (Baseline=1.0x).

consistently results in lower training time and GPU memory consumption compared to LoRA across all evaluated tasks, establishing HiFICL as a more computationally efficient PEFT solution for training.

8.3. Performance at Optimal Rank ($r = 8$)

As discussed in the main text, the default rank setting ($r = 16$) inherited from prior ICL approximation baselines can lead to overfitting for LoRA on certain datasets. To provide a generic and fair PEFT comparison, we evaluate both LoRA and HiFICL using an optimized rank of $r = 8$, which empirically serves as a robust "sweet spot" for both methods.

As shown in Tab. 8, when standardizing the rank to $r = 8$, the performance degradation of LoRA vanishes, confirming that the earlier drop was indeed due to capacity-induced overfitting. In this optimized setting, HiFICL achieves performance parity with LoRA across both LLaVA and Idefics2 models. Crucially, HiFICL achieves this highly competitive performance while utilizing approximately 4× fewer trainable parameters (2.2M vs. 8.8M/9.8M). This demonstrates that our context-aware parameterization approach is not only mathematically principled but also highly parameter-efficient compared to standard weight-space adaptation.

Table 8. Performance comparison between LoRA and HiFICL with an optimal rank of $r = 8$. Best results for each model are **bolded**.

Model	Method	# Params (M)	VQAv2	OK-VQA	COCO
LLaVA	LoRA	9.8 (×4.5)	**75.75**	**54.28**	1.3307
	HiFICL	2.2 (×1.0)	74.66	53.19	**1.3315**
Idefics2	LoRA	8.8 (×4.0)	69.58	**60.18**	**1.3448**
	HiFICL	2.2 (×1.0)	**72.08**	58.96	1.2851