MMICT: Boosting Multi-Modal Fine-Tuning with In-Context Examples

TAO CHEN*, Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, China
ENWEI ZHANG*, YUTING GAO, KE LI, and XING SUN, Tencent Youtu Lab, China
YAN ZHANG, HUI LI†, and RONGRONG JI, Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, China

Although In-Context Learning (ICL) brings remarkable performance gains to Large Language Models (LLMs), the improvements remain lower than fine-tuning on downstream tasks. This paper introduces Multi-Modal In-Context Tuning (MMICT), a novel multi-modal fine-tuning paradigm that boosts multi-modal fine-tuning by fully leveraging the promising ICL capability of multi-modal LLMs (MM-LLMs). We propose the Multi-Modal Hub (M-Hub), a unified module that captures various multi-modal features according to different inputs and objectives. Based on M-Hub, MMICT enables MM-LLMs to learn from in-context visual-guided textual features and subsequently generate outputs conditioned on the textual-guided visual features. Moreover, leveraging the flexibility of M-Hub, we design a variety of in-context demonstrations. Extensive experiments on a diverse range of downstream multi-modal tasks demonstrate that MMICT significantly outperforms traditional fine-tuning strategy and the vanilla ICT method that directly takes the concatenation of all information from different modalities as input. Our implementation is available at: https://github.com/KDEGroup/MMICT.

CCS Concepts: • **Computing methodologies** → **Natural language generation**; **Computer vision tasks**.

Additional Key Words and Phrases: Multi-Modal Alignment, Text Generation, In-Context Tuning

ACM Reference Format:
Tao Chen, Enwei Zhang, Yuting Gao, Ke Li, Xing Sun, Yan Zhang, Hui Li, and Rongrong Ji. 2024. MMICT: Boosting Multi-Modal Fine-Tuning with In-Context Examples. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1 (January 2024), 17 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recently, a great number of works on large-scale language models (LLMs) [44] have sprung up, propelling the evolution of human-like artificial intelligence. By escalating the model size, for instance, from 1 billion (GPT 1) to 175 billion parameters (GPT 3) or more, LLMs can demonstrate extraordinary proficiency in comprehending human language. Many researchers attempt to further augment the text-based LLMs by incorporating additional modalities (e.g., image, and video), leading

*Both authors contributed equally.
†Corresponding Author.

Authors' Contact Information: Tao Chen, taochen@stu.xmu.edu.cn, Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, China; Enwei Zhang, miyozhang@tencent.com; Yuting Gao, yutinggao@tencent.com; Ke Li, tristanli.sh@gmail.com; Xing Sun, winfred.sun@gmail.com, Tencent Youtu Lab, China; Yan Zhang, bzhy986@gmail.com; Hui Li, hui@xmu.edu.cn; Rongrong Ji, rrji@xmu.edu.cn, Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, China.

ACM 1551-6865/2024/1-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

to the creation of multi-modal LLMs (MM-LLMs). Representative works include but not limited to KOSMOS-1 [15], Flamingo [1] and GPT-4 [26].

With the prosperous development, LLMs have shown a capacity for in-context learning (ICL) [9], which involves learning and prediction solely based on a few examples in the context and does not update any parameters of LLMs. For instance, in the field of MM-LLMs, Flamingo capitalizes on the interleaved multi-modal data to enhance its multi-modal in-context learning capabilities. Building upon OpenFlamingo [3], the open-source version of Flamingo, Otter [17] is capable of executing new instructions with a few in-context learning examples using multi-modal in-context instruction tuning. Additionally, some studies [25, 45] verify that various demonstration factors (e.g., demonstration format and demonstration order) heavily affect the performance of ICL.

Although ICL can bring remarkable performance gains to MM-LLMs, the improvements still lag behind fine-tuning on training data for downstream tasks [1, 6]. For example, on the VQAv2 task, Flamingo achieves 63.1% accuracy with 4 demonstration examples while the accuracy after fine-tuning is 82.0%. The observation inspires us: Can we combine the two learning paradigms by leveraging ICL to further enhance the fine-tuning performance on downstream multi-modal tasks?

To this end, in this paper, we propose Multi-Modal In-Context Tuning (MMICT), a novel multi-modal fine-tuning paradigm that harnesses ICL to improve multi-modal fine-tuning. MMICT enables MM-LLMs to learn from visual-guided textual features of demonstration examples in fine-tuning. Furthermore, based on the in-context information and the textual-guided visual features extracted from visual inputs and textual instructions, MMICT predicts the textual label paired with the visual inputs.

MMICT is built based on BLIP-2 [18]. BLIP-2 adopts a traditional fine-tuning strategy with only query inputs, and exclusively utilizes the cross-modal pre-trained Qformer to extract visual features. To better capture the multi-modal features within a unified model architecture, we design the Multi-Modal Hub (M-Hub) used in MMICT. Different from Qformer, M-Hub can produce either uni-modal features or multi-modal features that fuse information from different modalities. Considering that different demonstration factors (e.g., feature extraction strategy, sampling number for demonstrations and sampling strategies for demonstrations) may heavily affect the performance, we design various variants of in-context demonstrations by leveraging the flexibility of M-Hub.

In summary, the contributions of this work are:

- **Innovative Paradigm.** We introduce MMICT, a novel fine-tuning paradigm that can further augment the performance of MM-LLMs on a variety of downstream multi-modal tasks by harnessing its promising ICL capability. Furthermore, our proposed MMICT exhibits robustness against varying demonstration surfaces.
- **Thoughtful Design.** Based on the unique model architecture and representation learning strategy of Q-former used in the pre-training stage, we transcend Q-former's conventional use as a uni-modal feature extraction module by advancing it to M-Hub that is capable of capturing both uni-modal representations and visual-language representations within a unified architecture.
- **Insightful Discoveries.** Through the exploration of various demonstration formats, we unveil several intriguing and pivotal findings. These insights illuminate potential explanations and pave the way for future research in this domain.

The remaining parts of this paper are organized as follows: Sec. 2 introduces the related work of this study. Sec. 3 describes the details of MMICT. Sec. 4 provides the results and analysis of experiments. Sec. 7 concludes this work.

2 RELATED WORK

2.1 Multi-Modal Large Language Models

In recent years, the trend of using LLMs to integrate information from multiple modalities has gained significant attention, resulting in the so-called MM-LLMs.

Pioneering studies like VisualGPT [5] and Frozen [31] have demonstrated the benefits of employing a pre-trained language model as a vision-language model decoder. Flamingo [1] is proposed to align a pre-trained vision encoder and language model using the gated cross-attention mechanism. It is trained on billions of image-text pairs, showcasing impressive in-context few-shot learning capabilities. BLIP-2 [18] introduces a Q-Former to efficiently align visual features with the language model. GPT-4 [26] shows more powerful visual understanding and reasoning abilities after pre-training on a vast collection of aligned image-text data. To empower LLMs with the ability of video understanding, a multi-branch cross-modal pre-training framework Video-LLaMA [42] is proposed to achieve both vision-language alignment and audio-language alignment by connecting the LLM to off-the-shelf uni-modal pre-trained models. MoE-LLaVA [21] constructs a MoE-based sparse LVLM architecture with an outrageous number of parameters but a constant computational cost. Gemini family [2] exhibits remarkable capabilities across image, audio, video, and text understanding even on memory-constrained devices.

In summary, there are four mainstream methods that combine visual encoder and LLM into MM-LLM: 1) The addition of extra modules in the LLM that enable deep interaction and fusion, exemplified by Otter [17] and CogVLM [34], which incurs high computational costs. 2) The incorporation of learnable query tokens to extract information in a query-based manner, facilitating the conversion of features to enhance the comprehension of LLM, as demonstrated in the BLIP-2 [18], X-LLM [4] and MiniGPT4 [46], which compress visual tokens into a smaller number of representation vectors. 3) The inclusion of supplementary learnable parameters in the LLM, as seen in the LLaMA-Adapter [12], yields a faster training speed albeit with marginally lower performance in comparison to alternative approaches. 4) The simple usage of a MLP-based interface to bridge the modality gap. For example, LLaVA series adopts one/two linear MLP [23, 24] to project visual tokens and align the feature dimension with word embeddings. Despite their simplicity, the large amount of prefix visual tokens increases the context length in MM-LLM. Considering the pros and cons of the aforementioned methods, we opt for BLIP-2 as the base of MMICT.

2.2 In-Context Learning

In-context learning involves learning based on only a few examples in the form of demonstration. Essentially, it estimates the likelihood of the potential answer conditioned on the demonstration using a well-trained language model. For multi-modal tasks, Flamingo [1] capitalizes on the interleaved multi-modal data to enhance its few-shot ICL capabilities. Moreover, the paradigm of generating query text conditioned on in-context examples ensures its ICL capacity during the inference phase. Building upon OpenFlamingo, Otter [17] introduces the in-context instruction tuning paradigm for multi-modal models. However, they do not capture cross-modality information and only use uni-modal features from different modalities as demonstrations. The multi-modal features are modeled in the cross-attention module of the LLM. Differently, MMICT models cross-modality information to guide the construction of the multi-modal demonstrations. More detailed comparisons are provided in Sec. 6.

As demonstrations play a vital role in ICL, many works study demonstration designing strategies. For instance, in natural language processing, several works aim to select good examples for ICL through unsupervised methods based on pre-defined metrics [19] or supervised methods [35]. For tasks requiring complex reasoning (e.g., math word problems and commonsense reasoning), some

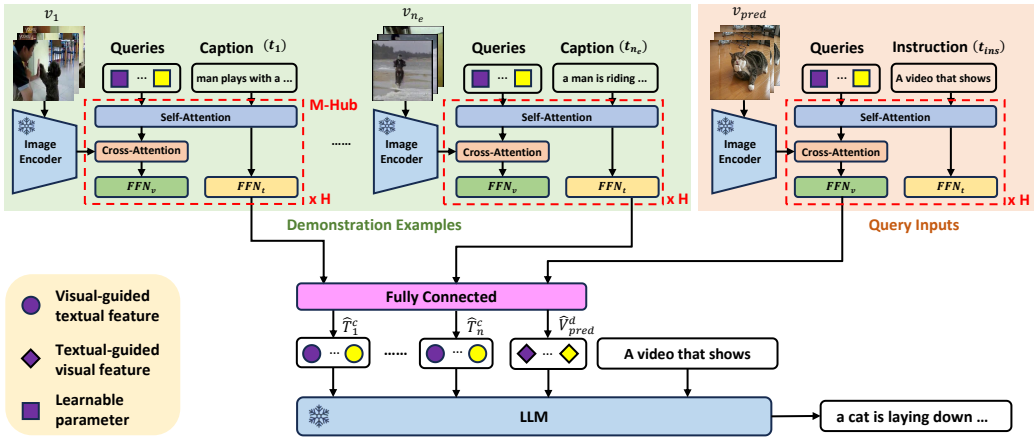

Fig. 1. Overview of MMICT. M-Hub can output both visual-guided textual features (upper left green part) and instruction-guided visual features (upper right orange part). MMICT learns from visually-guided textual features derived from demonstration examples and generates outputs based on instruction-guided visual features obtained from input queries.

works design better demonstrations for ICL by describing tasks with the instruction [36] and adding intermediate reasoning steps [37]. Differently, MMICT proposes a better feature extraction strategy for demonstrations to avoid information redundancy and exhibits robustness against different demonstration surfaces.

2.3 Multi-modal Alignment

Multi-modal tasks necessitate a model that adeptly translates visual scenes into language descriptions. Currently popular approaches [10, 20, 29] leverage an encoder-decoder architecture and achieve promising results. An optimal multi-modal model embodies the trifecta of visual comprehension, visual-text alignment, and textual generation. Initially, a diverse array of feature extractors is employed to procure visual representations, fundamental for basic visual understanding. Subsequently, these visual representations are transformed into a hidden vector h, which shares a semantic space with the language representations' hidden vector, thus bridging the gaps between vision and language. In essence, this alignment occurs in a shared semantic space, uniting the visual modality (e.g., video or image) with the textual modality. Ultimately, the language representations undergo decoding to produce the definitive descriptions. The pivotal step in this multi-modal model is the visual-text alignment, mapping visual representations to the language domain. This alignment is crucial because the language network (such as LSTM or Transformer) within the decoder can unleash its full text generation prowess when the input representation resonates with the language domain. However, existing techniques tend to concentrate either on video comprehension [28, 30] or image understanding [16, 33]. Thanks to the flexibility of our proposed M-Hub, MMICT is capable of executing text generation tasks, whether based on images or videos, within a unified framework.

3 OUR METHOD

In this section, we illustrate the details of MMICT. Fig. 1 provides an overview of MMICT. Firstly, we demonstrate how MMICT learns from in-context visual-guided textual features and generates outputs according to textual-guided visual features in Sec. 3.1. Then, in Sec. 3.2, we illustrate the Multi-Modal Hub (M-Hub) that encodes the multi-modal fused features in MMICT within a unified

architecture. In the following, we use lower-case fonts to indicate raw data, bold lower-case fonts to represent vectors, and bold upper-case fonts to denote matrices.

3.1 Multi-Modal In-Context Tuning

For multi-modal tasks, MMICT introduces a novel fine-tuning paradigm that fully leverages the remarkable ICL capacity of MM-LLMs. If only the visual or textual data is selected and fed into MM-LLMs, it could lead to a performance decline due to the absence of information from the other modality. To encapsulate the visual-textual information present in multi-modal tasks, a straightforward method could be directly concatenating the visual and textual data from the demonstration examples together, and then feeding them into MM-LLMs. This simplistic approach, however, is suboptimal as it incorporates a significant amount of redundant information from the visual and textual modalities. Instead, we argue that fusing multi-modal information as demonstrations is a more effective strategy, as it not only integrates information from different modalities but also circumvents information redundancy.

In this study, we feed paired features from different modalities into the Multi-modal Hub (M-Hub, see Sec. 3.2) to obtain multi-modal fused features. Moreover, considering the modality gap between vision and text, we retain the visual-guided textual features as demonstrations. To elucidate the detailed formulation of MMICT, let us take video captioning as an example. As depicted in Fig. 1, given a video-instruction pair $\{v_{pred}, t_{ins}\}$ (upper right orange part in Fig. 1) accompanied with other pairs $\{v_1, t_1, ..., v_{n_e}, t_{n_e}\}$ (upper left green part in Fig. 1) that are randomly selected as demonstration examples, where $\{v_*, t_*\}$ denotes a video clip and the corresponding text, and n_e is the number of demonstration examples. The model needs to predict the label y (e.g., the caption text "a cat is laying down washing his face") paired with v_{pred} according to the in-context information. The frozen image encoder (e.g. EVA [11]) takes as input the video clips $\{v_1, ..., v_{n_e}, v_{pred}\}$ and outputs the corresponding encoded visual features $\{\mathbf{Z}_1^v, ..., \mathbf{Z}_{n_e}^v, \mathbf{Z}_{pred}^v\}$. We first feed these features and their paired text into M-Hub, and then pass them to a share-weight fully-connected network to extract visual-guided textual demonstration features and textual-guided visual features:

$$\hat{\mathbf{T}}_k^c = \mathrm{FC}\big(\mathcal{G}(\mathbf{Z}_k^v, t_k)\big) \tag{1}$$

$$\hat{\mathbf{V}}_{pred}^d = \mathrm{FC}\big(\mathcal{G}(\mathbf{Z}_{pred}^v, t_{ins})\big) \tag{2}$$

where $k \in \{1, ..., n_e\}$. FC and t_{ins} denote single-layer fully-connected network and textual instruction, respectively. $\mathcal{G}$ indicates M-Hub. Finally, we concatenate multi-modal fused features with t_{ins} and feed them into the frozen LLM:

$$\begin{aligned} \mathrm{C} &= \left[\hat{\mathbf{T}}_1^c, \langle \mathrm{EOC} \rangle, ..., \hat{\mathbf{T}}_{n_e}^c, \langle \mathrm{EOC} \rangle, \hat{\mathbf{V}}_{pred}^d\right] \\ \hat{y} &= \mathrm{LLM}\big([\mathrm{C}, \mathcal{T}(t_{ins})]\big) \end{aligned} \tag{3}$$

where $[\cdot, \cdot]$ denotes the concatenation operation. $\langle \mathrm{EOC} \rangle$ token ("end of chunk") is appended to the end of each demonstration example for separating them explicitly, and $\mathcal{T}$ is the textual tokenizer from the LLM. $\hat{y}$ denotes the outputs of the model conditioned on the in-context information, and it gradually gets closer to the ground truth y during training.

3.2 Multi-Modal Hub

Owing to the huge model size of LLMs and vision foundation models, training their parameters for multi-modal tasks proves to be challenging. As a result, many researchers endeavor to incorporate a comparatively lightweight and trainable Visual Prompt Generator (VPG) between them while maintaining their fully frozen state [41]. Among these efforts, the BLIP-style multi-modal pre-training approach effectively connects LLMs to vision foundation models via the Q-former. However,

previous works [18, 46] solely utilize this module to extract uni-modal visual features for LLMs after pre-training, thereby overlooking the benefits of multi-modal pre-training.

To address this issue, we propose the M-Hub for capturing multi-modal features. As depicted in the upper left part of the Fig. 1, M-Hub comprises of H blocks. Specifically, each block in the M-Hub consists of a shared self-attention module, a cross-attention module that interacts with the frozen image encoder for extracting text-aligned visual features digestible for LLMs, and two modality-specific feed-forward layers. The queries are a set of learnable parameters $\mathbf{Q} = \{\mathbf{q}_i\}_{i=1}^{n_q}$, where n_q is the number of parameters. We initialize the weights of the M-Hub using the pre-trained Q-former to leverage the advantages of multi-modal pre-training.

Given a video-text pair $\{v, t\}$, we uniformly sample n_f frames from the video clip v, and subsequently extract frame features $\mathbf{Z}^v = \{\mathbf{E}_i^v\}_{i=1}^{n_f}$ by separately passing each frame through the frozen image encoder. Then, a simple approach can be acquiring frame-level visual features via individually feeding frame features into VPG and then concatenating them. Considering that there exists both inter-modality information redundancy and intra-modality information redundancy (e.g., within a video clip), the aforementioned approach is suboptimal. To handle this problem, we propose a simple yet efficient method that fully capitalizes on its robust capacity to filter redundant information. Specifically, we flatten frame features and then feed them into M-Hub to obtain the video-level visual features.

Moreover, the unified architecture of the M-Hub can enhance the representation learning of visual-textual inputs. For instance, M-Hub can output both visual-guided textual features $\hat{\mathbf{T}}^c$ and textual-guided visual features $\hat{\mathbf{V}}^d$ as follows:

$$\begin{aligned} \mathbf{P}_h, \mathbf{R}_h &= \text{Self-Attention}\big([\hat{\mathbf{V}}_{h-1}^d, \hat{\mathbf{T}}_{h-1}^c]\big) \\ \mathbf{O}_h &= \text{Cross-Attention}\big(\mathbf{P}_h, \mathbf{Z}^v\big) \\ \hat{\mathbf{V}}_h^d &= \text{FFN}_v\big(\mathbf{O}_h\big), \quad \hat{\mathbf{T}}_h^c = \text{FFN}_t\big(\mathbf{R}_h\big) \end{aligned} \tag{4}$$

where $\hat{\mathbf{V}}_0^d = \mathbf{Q}$, $\hat{\mathbf{T}}_0^c = t$, and h is the h-th block of M-Hub. In the h-th block of M-Hub, we handle features $\hat{\mathbf{V}}_{h-1}^d$ and $\hat{\mathbf{T}}_{h-1}^c$ from different modalities, which are outputs from the preceding block. To integrate multi-modal information, these features are concatenated within the self-attention layer. We partition the previous n_q tokens in the concatenated features to derive multi-modal fused visual features $\mathbf{P}_h$, and retain the remaining tokens to obtain multi-modal fused textual features $\mathbf{R}_h$. The valuable visual information $\mathbf{P}_h$ is extracted together with $\mathbf{Z}^v$ via the cross-attention layer, resulting in $\mathbf{O}_h$. Finally, $\mathbf{O}_h$ and $\mathbf{R}_h$ are passed through modality-specific feed-forward layers to yield $\hat{\mathbf{V}}_h^d$ and $\hat{\mathbf{T}}_h^c$, respectively. Note that, depending on the varying inputs and objectives, M-Hub can output other types of features. We will elaborate on its flexibility in Sec. 4.6.

3.3 Training Objective

MMICT is trained on the next-token prediction task, i.e., learn to generate the next token depending on the previous context. The training objective is to minimize the negative log-likelihood of tokens in the label y:

$$\mathcal{L} = -\sum_{y \in Y} \sum_{t=1}^{|y|} log\big(p(y_t|\mathbf{C}, t_{ins}, y_1, ..., y_{t-1})\big) \tag{5}$$

where y_t is the t-th token in the ground truth label y, $|y|$ is the number of tokens in y, and Y is the ground truth label set.

Table 1. Instruction templates used for image captioning and video captioning.

Task	Instruction Templates
Image Captioning	A short image caption: A image that shows Write a short description for the image. Briefly describe the content of the image. Use a few words to illustrate what is happening in the image. Can you briefly explain what you see in the image?
Video Captioning	A short video caption: A video that shows Write a short description for the video. Briefly describe the content of the video. Use a few words to illustrate what is happening in the video. Can you briefly explain what you see in the video?

3.4 Inference

During inference, MMICT generates predictions as follows:

$$\hat{y} = \text{LLM}\big([\hat{\mathbf{V}}^d_{pred}, \mathcal{T}(t_{ins})]\big) \tag{6}$$

For evaluation, unlike the training stage, we first generate the predicted label $\hat{y}$. Then, $\hat{y}$ is compared with the ground-truth label y for calculating different evaluation metrics.

4 EXPERIMENTS

4.1 Implementation Details

We follow BLIP-2[1] to implement MMICT. Concretely, we experiment with two LLMs: FlanT5 [8] with encoder-decoder architecture and OPT [43] with decoder-only architecture. In our approach, we utilize $\text{FlanT5}_{\text{XL}}$ for FlanT5 and $\text{OPT}_{\text{2.7B}}$ for OPT, respectively. For the frozen image encoder, we use EVA [11], a state-of-the-art pre-trained vision transformer model. We initiate the parameters of M-Hub with the pre-trained Q-former. We freeze the image encoder and the LLM, and only train the M-Hub and the fully-connected network (Eq. 1 and Eq. 2) for better evaluating the effectiveness of MMICT.

4.2 Evaluation Tasks and Datasets

We evaluate MMICT on several prevalent downstream multi-modal tasks, including image captioning, video captioning, visual question answering (VQA) and video question answering (VideoQA) across six different datasets.

For captioning tasks, we evaluate MMICT on 3 public datasets including COCO Caption [22], MSVD [38], MSR-VTT [40]. We use BLEU@4 (B@4) [27], CIDEr (C) [32] as metrics. t_{ins} is randomly sampled from the pre-defined instruction templates , which are shown in Tab. 1

For open-ended question answering tasks, we evaluate MMICT on 3 public datasets including VQAv2 [13], MSVD [38] and MSR-VTT [38]. We formulate them as generative problems. During inference, we use beam search with a beam width of 5 to generate answers from the whole vocabulary with no restrictions. Accuracy (Acc) is used as the evaluation metric. We evaluate VQAv2 on its validation set (the test label is not publicly available) and evaluate MSVD and MSR-VTT on their respective test sets. t_{ins} for these tasks is designed as "Question: {} Answer:".

[1] https://github.com/salesforce/lavis

4.3 Environment and Hyper-parameters

We follow most of the settings for fine-tuning hyper-parameters in BLIP-2, except that we freeze the image encoder and set the size of images/videos to be 224×224. We train our model for 5 epochs on 4 NVIDIA V100 GPUs. Each video clip comprises of n_f frames, we pass them through M-Hub to obtain video-level visual features. Therefore, we define the batch size in terms of the number of frames, and set the batch size on video tasks to 48. We treat an image as a single-frame video, where $n_f = 1$, and use a batch size of 15 for image tasks. n_f, n_e, and n_q are set to 16, 2, and 32, respectively. The demonstration examples for each data sample are randomly sampled from the same dataset. The AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$, and a weight decay of 0.05 is used. Additionally, we apply a linear warmup of the learning rate during the initial 1,000 steps, increasing from 10^{-8} to 10^{-5}, followed by a cosine decay with a minimum learning rate of 0.

4.4 Baselines

For simplicity, we denote the formulation of MMICT as $\{\hat{\mathbf{T}}^c_1, ..., \hat{\mathbf{T}}^c_{n_e}, \hat{\mathbf{V}}^d_{pred}\}$, which represents the inputs to the LLM. We consider two baselines in our experiments:

- **VanillaFT**: It is the traditional method for fine-tuning on downstream tasks. Its formulation can be symbolized as $\{\mathbf{V}^a_{pred}\}$.
- **VanillaICT-B$_{\mathbf{VT}}$**: VanillaICT denotes that M-Hub serves only as a uni-modal encoder. And we use 'Base' (B) to indicate that the text is directly fed into the LLM. VanillaICT-B$_{\mathrm{VT}}$ directly prompts an MM-LLM with the concatenation of all uni-modal information from demonstration examples to capture in-context information. We denote its formulation as $\{\mathbf{V}^a_1, t_1, ..., \mathbf{V}^a_{n_e}, t_{n_e}, \mathbf{V}^a_{pred}\}$.

We mainly compare MMICT with the above two baselines to demonstrate the superiority of using in-context learning to boost the fine-tuning performance of MM-LLMs. Additionally, we show the results of several state-of-the-art methods (SOTA) on each downstream task:

- **VLAB** [14]: VLAB is a video language pre-training method that transfers CLIP's learned representations to video-text tasks.
- **VAST** [7]: VAST is an omni-modality video-text foundational model that can perceive and process vision, audio, and subtitle modalities from videos.
- **mPLUG-2** [39]: mPLUG-2 is a multi-module composition network. It contains shared modules for modality collaboration and uses different modality modules to deal with modality entanglement.

4.5 Overall Performance

Tab. 2 reports the performance of MMICT compared with two baselines, i.e., VanillaFT and VanillaICT-B$_{\mathrm{VT}}$. The performance of SOTA methods is denoted in gray. From the results shown in Tab. 2, we can observe that:

(1) MMICT outperforms baselines for four downstream multi-modal tasks on six datasets, and even achieves new SOTA results on MSVD for video captioning and VideoQA tasks. The results show that in-context tuning can enhance the performance of MM-LLMs on downstream multi-modal tasks.

(2) A notable performance gap is observed between VanillaICT-B$_{\mathrm{VT}}$ with OPT and other methods across most datasets. One possible explanation could be that the superfluous information from demonstration examples may considerably impact the performance of MM-LLMs with decoder-only architectures. In these architectures, the outputs are invariably influenced by the inputs via the mask self-attention module. Conversely, MMICT utilize the in-context

Table 2. Performance of all methods. We mainly compare our MMICT with two baselines. The performance of SOTA methods is denoted in gray. The best results across different types of MM-LLMs are shown in bold.

Method	LLM	Caption						VQA	VideoQA	
		COCO		MSR-VTT		MSVD		VQAv2	MSR-VTT	MSVD
		B@4	C	B@4	C	B@4	C	Acc	Acc	Acc
VLAB [14]		-	152.5	54.6	74.9	79.3	179.8	-	49.6	61.0
VAST [7]		-	149.0	-	78.0	-	-	-	50.1	60.2
mPLUG-2 [39]		41.6	137.7	57.8	80.3	75.0	165.8	-	48.0	58.1
VanillaFT		42.4	144.5	51.3	74.7	73.0	174.3	69.6	43.4	63.9
VanillaICT-B_{VT}	FlanT5	42.4	142.5	50.6	74.4	75.4	178.5	69.8	42.8	64.3
MMICT		**43.6**	**145.7**	**51.6**	**74.8**	**76.5**	**179.1**	**72.3**	**45.6**	**66.7**
VanillaFT		43.7	**145.8**	47.6	69.9	80.3	177.1	56.9	42.6	56.9
VanillaICT-B_{VT}	OPT	36.3	116.9	37.7	57.3	43.2	82.9	62.7	40.5	64.7
MMICT		**43.9**	145.5	**52.0**	**71.4**	**80.4**	**180.4**	**73.0**	**46.0**	**66.3**

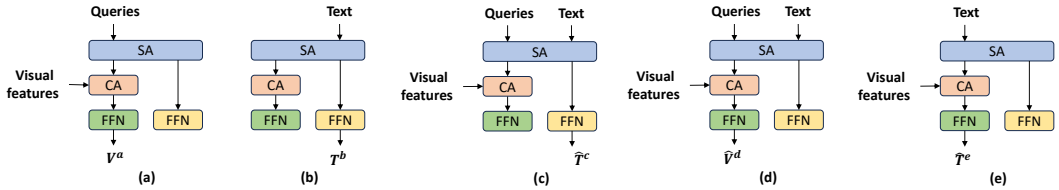

Fig. 2. Different usages of M-Hub. As demonstrated in (a) and (b), it can function as a uni-modal encoder. Moreover, it can also operate as a multi-modal fusion encoder, as shown in (c), (d), and (e).

Table 3. The results of ablation studies about ICT demonstration variants. Best results are shown in bold. For the evaluation, we train on one third data of the complate datasets that is randomly sampled.

Method	Caption				VQA	VideoQA
	COCO		MSVD		VQAv2	MSVD
	B@4	C	B@4	C	Acc	Acc
VanillaICT-B_{VT}	40.6	138.6	73.7	175.9	69.0	62.5
VanillaICT-B_{T}	41.8	140.2	74.7	176.7	68.4	61.5
VanillaICT-E_{T}	41.9	140.0	75.0	177.5	68.8	63.5
InstructICT-E_{VT}	41.4	138.9	73.8	175.2	69.0	62.1
InstructICT-E_{V}	38.7	133.4	69.7	166.3	69.4	62.9
InstructICT-E_{T}	41.5	139.8	74.2	176.8	69.8	62.3
MMICT	**42.0**	**140.4**	**76.2**	**177.7**	**69.9**	**64.3**

visual-guided textual features to incorporate multi-modal fused information and circumvent redundancy. This strategy consequently leads to a substantial enhancement in performance.

4.6 The Impacts of Demonstration Formats

As illustrated in Fig. 2, depending on the different inputs and objectives, M-Hub can function as:

(a) An uni-modal visual encoder. It can take image/video features as input, and output uni-modal visual features $\mathbf{V}^a$.
(b) An uni-modal textual encoder that can output uni-modal textual features $\mathbf{T}^b$.
(c) A multi-modal fusion encoder that can output visual-guided textual features $\hat{\mathbf{T}}^c$, where the learned queries and textual features interact with each other in the self-attention layers of the M-Hub.
(d) A multi-modal fusion encoder that can capture textual-guided visual features $\hat{\mathbf{V}}^d$.
(e) Moreover, we replace the learned queries with input text to obtain the visual-attended textual features $\hat{\mathbf{T}}^e$ for exploring the performance of multi-modal representations obtained after direct interaction between visual features and textual features.

Based on the flexibility of M-Hub mentioned above, we provide ablation studies to analyze the key factors that contribute to MMICT 's performance, with insights and qualitative results. In InstructICT, M-Hub functions as a multi-modal fusion encoder. Additionally, we use 'Encoding' (E) to signify that features from specific modalities are encoded by M-Hub. We design various in-context demonstration variants as follows:

- **VanillaICT-B$_\mathbf{T}$**: In VanillaICT-B$_\mathrm{T}$, we remove the visual information from the demonstration examples to explore the information redundancy existed in VanillaICT-B$_\mathrm{VT}$. We formulate VanillaICT-B$_\mathrm{T}$ as $\{t_1, ..., t_{n_e}, \mathbf{V}^a_{pred}\}$.
- **VanillaICT-E$_\mathbf{T}$**: As shown in Fig. 2 (b), the M-Hub can work as the uni-modal text encoder. To explore its effectiveness, we design VanillaICT-E$_\mathrm{T}$, which can be formulated as $\{\mathbf{T}^b_1, ..., \mathbf{T}^b_{n_e}, \mathbf{V}^a_{pred}\}$.
- **InstructICT-E$_\mathbf{VT}$**: We extend VanillaICT-B$_\mathrm{VT}$ to InstructICT-E$_\mathrm{VT}$, where the M-Hub works as the multi-modal fusion encoder. Its formulation can be denoted as $\{\hat{\mathbf{V}}^d_1, \hat{\mathbf{T}}^c_1, ..., \hat{\mathbf{V}}^d_{n_e}, \hat{\mathbf{T}}^c_{n_e}, \hat{\mathbf{V}}^d_{pred}\}$.
- **InstructICT-E$_\mathbf{V}$**: Compared with MMICT, InstructICT-E$_\mathrm{V}$ only retain the textual-guided visual features. Its formulation can be symbolized as $\{\hat{\mathbf{V}}^d_1, ..., \hat{\mathbf{V}}^d_{n_e}, \hat{\mathbf{V}}^d_{pred}\}$.
- **InstructICT-E$_\mathbf{T}$**: To explore the performance of adopting direct interactions between the visual features and the textual features, we replace the learned queries with the text inputs. the formulation of InstructICT-E$_\mathrm{T}$ can be symbolized as $\{\hat{\mathbf{T}}^e_1, ..., \hat{\mathbf{T}}^e_{n_e}, \hat{\mathbf{V}}^d_{pred}\}$.

To compare and analyze these in-context demonstration variants more efficiently, we randomly sample one third data of the datasets for using in the ablation studies. We show the ablation study results of these variants over four different datasets in Tab. 3. From the results, we have the following findings:

(1) When we replace the visual-guided textual features $\hat{\mathbf{T}}^c$ from demonstration examples with the uni-modal textual features $\mathbf{T}^b$, i.e., transitioning from MMICT to VanillaICT-E$_\mathrm{T}$, we observe that the performance remains almost unchanged on image captioning and video captioning tasks, while it declines on VQA and VideoQA tasks. The observation suggests that information redundancy exists across different modalities in captioning tasks, whereas multi-modal information is crucial for visual/video question answering tasks.
(2) InstructICT-E$_\mathrm{T}$ exhibits comparable performance on image captioning and VQA tasks when compared to MMICT. However, its performance is inferior to MMICT on video captioning and VideoQA tasks. The difference between InstructICT-E$_\mathrm{T}$ and MMICT lies in the fact that textual features directly interact with visual features through cross-attention in the former, while textual features and learned queries interact with each other through self-attention in the latter. A possible explanation for the performance difference could be that the information contained in images is mostly useful, whereas video clips contain redundant information. On

Table 4. Performance using different numbers of demonstration examples.

n_e	MSVD		
	Caption		VideoQA
	B@4	C	Acc
0	75.1	176.9	64.9
1	75.2	177.1	65.9
2	**76.5**	179.1	**66.7**
3	76.1	**179.5**	64.7
4	75.4	178.7	64.7

Table 5. Performance using different sample strategies for demonstration examples.

Sample strategy	Caption				VQA	VideoQA
	COCO		MSVD		VQAv2	MSVD
	B@4	C	B@4	C	Acc	Acc
Random	**43.6**	**145.7**	**76.5**	**179.1**	72.3	**66.7**
One-to-many	**43.6**	145.2	76.2	178.3	**72.6**	66.1

the other hand, learned queries demonstrate a strong ability to extract useful information, which can alleviate this issue.

(3) To illustrate the existence of a modality gap between vision and text, we compare the performance of using visual-guided textual features and textual-guided visual features as demonstrations, i.e., MMICT and InstructICT-E_V. And significant performance gaps are observed between them. Furthermore, InstructICT-E_V is almost inferior to all other variants. This observation indicates that MM-LLMs may struggle to learn from in-context visual features and could even be misled by them.

(4) VanillaICT-B_{VT} and VanillaICT-B_T directly feed raw text into the LLM. Different from them, InstructICT-E_{VT} and VanillaICT-E_T firstly employ M-Hub to encode raw text, and then input the enhanced representations into the LLM. However, no performance improvements are observed between them. We suspect that the powerful understanding ability of LLMs causes them to overlook the potential benefits of M-Hub.

4.7 The Impacts of Sampling for Demonstrations

We also investigate the impacts of different settings of sampling for demonstrations on the performance.

4.7.1 Number of Samples. Tab. 4 provides the experimental results with the sample number n_e varying from 0 to 4 on MSVD. Note that when $n_e = 0$, the model generates outputs only according to the textual-guided visual features from query inputs. From Tab. 4, it is observable that MMICT can effectively learn from in-context information when n_e is set to 1 or 2. However, as n_e continues to increase, the performance remains almost unchanged and may even decline. The observation suggests that the MM-LLMs could be negatively influenced by the in-context information when n_e becomes overly large.

4.7.2 Sampling Strategy. The performance using different sampling strategies for demonstration examples is reported in Tab. 5. One-to-many indicates that for each video-text pair in the dataset, we

Table 6. Results of using demonstrations during inference.

Method	Demonstrations	Caption				VQA	VideoQA
		COCO		MSVD		VQAv2	MSVD
		B@4	C	B@4	C	Acc	Acc
VanillaFT	✗	42.4	144.5	73.0	174.3	69.6	63.9
VanillaICT-B_{VT}	✗	42.4	142.5	75.4	178.5	69.8	64.3
MMICT	✗	**43.6**	145.7	**76.5**	179.1	72.3	**66.7**
VanillaFT	✓	33.8	111.2	51.1	119.0	67.0	47.8
VanillaICT-B_{VT}	✓	43.2	143.8	75.3	178.2	70.0	56.6
MMICT	✓	**43.6**	**145.9**	**76.5**	**180.0**	**72.8**	56.7

Table 7. Results using different levels of features in VanillaFT.

Level	Caption				VideoQA	
	MSR-VTT		MSVD		MSR-VTT	MSVD
	B@4	C	B@4	C	Acc	Acc
Frame	**51.3**	74.7	73.0	174.3	**43.4**	**63.9**
Video	51.0	**75.1**	**74.6**	**177.9**	42.9	63.3

randomly sample its demonstration examples from the same video that have different text paired with them. The results in Tab. 5 show that MMICT is robust to changes in the different sample strategies.

4.8 The Impacts of Demonstrations on Inference

Tab. 6 presents the impact of using demonstrations on the model's performance during the inference phase. The used LLM is FlanT5. Note that the results for the default setting reported in Tab. 1 of our submission use demonstrations during fine-tuning. For each data sample in the test set, we randomly sample its demonstrations from the training set. The results lead us to the following observations:

(1) The performance of MMICT gets slightly improved when demonstrations are incorporated during inference across most tasks. However, for the VideoQA task on the MSVD dataset, demonstrations appear to affect the performance of all methods negatively. Despite this, MMICT consistently surpasses baselines.
(2) VanillaICT-B_{VT} demonstrates similar performance on most tasks, regardless of whether demonstrations are used during the inference phase or not. In contrast, a significant decline in performance can be observed for VanillaFT when demonstrations are incorporated.

Above findings indicate that the use of demonstrations during inference can have varying effects on different methods and tasks. MMICT and VanillaICT-B_{VT} have fully learned from demonstrations during in-context tuning and they show comparable performance regardless of whether demonstrations are leveraged during inference. Besides, the observations underscore the importance of in-context tuning in enabling MM-LLMs to learn from demonstrations effectively.

4.9 The Impacts of Feature Levels

Tab. 7 shows the performance of using different levels of features in VanillaFT. For frame-level features, we pass each video clip through the frozen image encoder and M-Hub individually, and then concatenate them together to obtain frame-level features. The results in Tab. 7 illustrate that, despite using 16 times fewer tokens (i.e., frame number n_f), the performance using video-level features

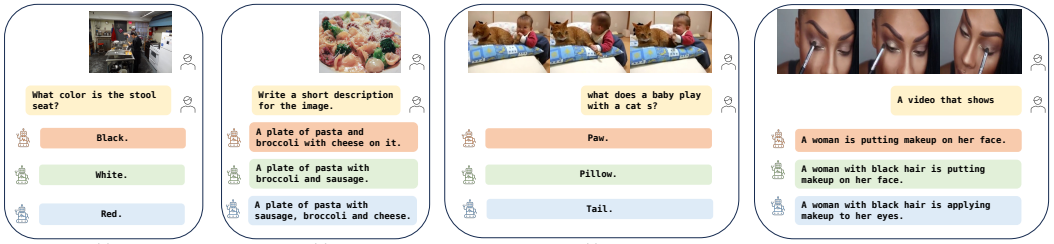

Fig. 3. Case study on (a) visual question answering, (b) image captioning, (c) video question answering, and (d) video captioning. We show the answers generated by VanillaFT, VanillaICT-B_{VT} and MMICT in orange, green and blue, respectively.

Table 8. Results on different sets of VQAv2.

Method	LLM	VQAv2	
		val	test-dev
VanillaFT	FlanT5	69.6	71.6
VanillaICT-B_{VT}	FlanT5	69.8	71.8
MMICT	FlanT5	**72.3**	**74.6**
VanillaFT	OPT	56.9	60.0
VanillaICT-B_{VT}	OPT	62.7	64.0
MMICT	OPT	**73.0**	**75.2**

is comparable to, and in some cases even surpasses, the performance achieved using frame-level features. The observation further demonstrates that video clips contain redundant information which does not significantly contribute to the performance. Note that, due to the limitations of the input length of LLMs, we do not conduct experiments for VanillaICT-B_{VT} and MMICT, which additionally take demonstrations as input.

4.10 Case Study

We randomly sample some cases covering four different multi-modal downstream tasks. The results of the sampled cases generated by MMICT, VanillaFT and VanillaICT-B_{VT} are shown in Fig. 3. From Fig. 3, we can observe that MMICT is capable of better understanding the detailed information contained in multi-modal data, while it is difficult for baselines to capture the details. For instance, in Fig. 3(d), baselines only recognize the face while MMICT can identify that the target is eyes.

5 EXPERIMENTS ON VQAV2

Note that the test labels of VQAv2 dataset are not publicly available. Tab. 8 reports the results of MMICT compared with two baselines, i.e., VanillaFT and VanillaICT-B_{VT}, on the validation set and the test set. The results demonstrate that MMICT consistently outperforms the baselines on the validation set and the test set of VQAv2.

6 THE DIFFERENCES BETWEEN MMICT AND OTTER

Otter is the most similar wrok compared with MMICT, we have briefly discussed it in Sec. 2.2. In this section, we provide detailed comparisons between them, including the experimental results.

As shown in Tab. 9, the four main differences between MMICT and Otter are as follows:

Table 9. The differences to Otter.

Method	Target Stage	Multi-modal Fusion	In-context Unrestriction	Prediction w/o Examples
Otter	Pre-training	LLM	✗	✗
MMICT	Fine-tuning	M-Hub	✓	✓

Table 10. Results compared with Otter on MSVD tasks.

Method	LLM	Caption		VideoQA
		B@4	C	Acc
Otter	LLaMa7B	78.9	**184.2**	65.2
MMICT	OPT2.7B	**80.4**	180.4	**66.3**

(1) Otter uses in-context examples during pre-training, while MMICT is a novel fine-tuning paradigm applied to downstream tasks.
(2) The interaction of visual and textual features is implemented in M-hub. Conversely, Otter incorporates additional cross-attention modules within the layers of LLM, leading to an increase in computational requirements compared to M-hub.
(3) The data format of Otter necessitates meticulous design, as depicted in Fig. 2 of their paper. In contrast, our data format is more flexible and does not mandate a specific design. Remarkably, our method can function effectively even with random sampling for demonstrations, as illustrated in Section 4.7.
(4) At inference, Otter generates predictions based on the provided in-context examples. Differently, MMICT exhibits robust performance without demonstrations.

Furthermore, as displayed in Tab. 10, we also provide a direct comparison with Otter. From the results, we can find that although the size of LLM in MMICT is much smaller than that in Otter, MMICT exists comparable and even superior performance on video captioning and VideoQA tasks when compared to Otter.

7 CONCLUSION

In this paper, we propose MMICT for boosting multi-modal fine-tuning with in-context examples. MMICT enables MM-LLMs to learn from visual-guided textual features of demonstrations, and subsequently generate outputs with the textual-guided visual features of input queries. We propose the M-Hub used in MMICT to capture the multi-modal fused features within a unified architecture. Furthermore, we design various demonstration variants by fully considering the flexibility of M-Hub. From our extensive experiments conducted across six different multi-modal datasets, we can find that MMICT exceeds traditional fine-tuning strategy and VanillaICT. Additional experiments on different demonstration factors (i.e., feature extraction strategy, sampling number for demonstrations and sampling strategies for demonstrations) further ascertain the effectiveness and robustness of MMICT. In the future, we plan to experiment MMICT with more modalities (e.g., audio) and verify its effectiveness on more multi-modal tasks.

ACKNOWLEDGMENTS

This work is supported by National Science and Technology Major Project (No. 2022ZD0118201) and National Natural Science Foundation of China (No. 62002303, 42171456).

REFERENCES

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a Visual Language Model for Few-Shot Learning. In *NeurIPS*. 23716–23736.

[2] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023. Gemini: A Family of Highly Capable Multimodal Models. *arXiv Preprint* (2023). https://arxiv.org/abs/2312.11805.

[3] Anas Awadalla, Irena Gao, Joshua Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. 2023. *OpenFlamingo*. https://doi.org/10.5281/zenodo.7733589

[4] Feilong Chen, Minglun Han, Haozhi Zhao, Qingyang Zhang, Jing Shi, Shuang Xu, and Bo Xu. 2023. X-LLM: Bootstrapping Advanced Large Language Models by Treating Multi-Modalities as Foreign Languages. *arXiv Preprint* (2023). https://arxiv.org/abs/2305.04160.

[5] Jun Chen, Han Guo, Kai Yi, Boyang Li, and Mohamed Elhoseiny. 2022. VisualGPT: Data-efficient Adaptation of Pretrained Language Models for Image Captioning. In *CVPR*. 18009–18019.

[6] Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. 2022. Improving In-Context Few-Shot Learning via Self-Supervised Training. In *NAACL-HLT*. 3558–3573.

[7] Sihan Chen, Handong Li, Qunbo Wang, Zijia Zhao, Mingzhen Sun, Xinxin Zhu, and Jing Liu. 2023. VAST: A Vision-Audio-Subtitle-Text Omni-Modality Foundation Model and Dataset. *arXiv Preprint* (2023). https://arxiv.org/abs/2305.18500.

[8] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. *arXiv Preprint* (2022). https://arxiv.org/abs/2210.11416.

[9] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A Survey on In-context Learning. *arXiv Preprint* (2023). https://arxiv.org/abs/2301.00234.

[10] Shanshan Dong, Tian-Zi Niu, Xin Luo, Wu Liu, and Xinshun Xu. 2023. Semantic Embedding Guided Attention with Explicit Visual Feature Fusion for Video Captioning. *ACM Trans. Multim. Comput. Commun. Appl.* 19, 2 (2023), 68:1–68:18.

[11] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. 2022. EVA: Exploring the Limits of Masked Visual Representation Learning at Scale. *arXiv Preprint* (2022). https://arxiv.org/abs/2211.07636.

[12] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. 2023. LLaMA-Adapter V2: Parameter-Efficient Visual Instruction Model. *arXiv Preprint* (2023). https://arxiv.org/abs/2304.15010.

[13] 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 *CVPR*. 6325–6334.

[14] Xingjian He, Sihan Chen, Fan Ma, Zhicheng Huang, Xiaojie Jin, Zikang Liu, Dongmei Fu, Yi Yang, Jing Liu, and Jiashi Feng. 2023. VLAB: Enhancing Video Language Pre-training by Feature Adapting and Blending. *arXiv Preprint* (2023). https://arxiv.org/abs/2305.13167.

[15] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, Qiang Liu, Kriti Aggarwal, Zewen Chi, Johan Bjorck, Vishrav Chaudhary, Subhojit Som, Xia Song, and Furu Wei. 2023. Language Is Not All You Need: Aligning Perception with Language Models. *arXiv Preprint* (2023). https://arxiv.org/abs/2302.14045.

[16] Weitao Jiang, Weixuan Wang, and Haifeng Hu. 2021. Bi-Directional Co-Attention Network for Image Captioning. *ACM Trans. Multim. Comput. Commun. Appl.* 17, 4 (2021), 125:1–125:20.

[17] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. 2023. Otter: A Multi-Modal Model with In-Context Instruction Tuning. *arXiv Preprint* (2023). https://arxiv.org/abs/2305.03726.

[18] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. *arXiv Preprint* (2023). https://arxiv.org/abs/2301.12597.

[19] Xiaonan Li and Xipeng Qiu. 2023. Finding Supporting Examples for In-Context Learning. *arXiv Preprint* (2023). https://arxiv.org/abs/2302.13539.

[20] Yehao Li, Jiahao Fan, Yingwei Pan, Ting Yao, Weiyao Lin, and Tao Mei. 2022. Uni-EDEN: Universal Encoder-Decoder Network by Multi-Granular Vision-Language Pre-training. *ACM Trans. Multim. Comput. Commun. Appl.* 18, 2 (2022), 48:1–48:16.

[21] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. 2024. MoE-LLaVA: Mixture of Experts for Large Vision-Language Models. *arXiv Preprint* (2024). https://arxiv.org/abs/2401.15947.

[22] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*, Vol. 8693. 740–755.

[23] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved Baselines with Visual Instruction Tuning. *arXiv Preprint* (2023). https://arxiv.org/abs/2310.03744.

[24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning. In *NeurIPS*.

[25] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. In *ACL*, Vol. 1. 8086–8098.

[26] OpenAI. 2023. GPT-4 Technical Report. *arXiv Preprint* (2023). https://arxiv.org/abs/2303.08774.

[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL*. 311–318.

[28] Min Peng, Xiaohu Shao, Yu Shi, and Xiangdong Zhou. 2024. Hierarchical Synergy-Enhanced Multimodal Relational Network for Video Question Answering. *ACM Trans. Multim. Comput. Commun. Appl.* 20, 4 (2024), 91:1–91:22.

[29] Yaya Shi, Haiyang Xu, Chunfeng Yuan, Bing Li, Weiming Hu, and Zheng-Jun Zha. 2023. Learning Video-Text Aligned Representations for Video Captioning. *ACM Trans. Multim. Comput. Commun. Appl.* 19, 2 (2023), 63:1–63:21.

[30] Pengjie Tang, Hanli Wang, and Qinyu Li. 2019. Rich Visual and Language Representation with Complementary Semantics for Video Captioning. *ACM Trans. Multim. Comput. Commun. Appl.* 15, 2 (2019), 31:1–31:23.

[31] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal Few-Shot Learning with Frozen Language Models. In *NeurIPS*. 200–212.

[32] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *CVPR*. 4566–4575.

[33] Cheng Wang, Haojin Yang, and Christoph Meinel. 2018. Image Captioning with Deep Bidirectional LSTMs and Multi-Task Learning. *ACM Trans. Multim. Comput. Commun. Appl.* 14, 2s (2018), 40:1–40:20.

[34] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. 2023. CogVLM: Visual Expert for Pretrained Language Models. *arXiv Preprint* (2023). https://arxiv.org/abs/2311.03079.

[35] Xinyi Wang, Wanrong Zhu, and William Yang Wang. 2023. Large Language Models Are Implicitly Topic Models: Explaining and Finding Good Demonstrations for In-Context Learning. *arXiv Preprint* (2023). https://arxiv.org/abs/2301.11916.

[36] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *ACL*, Vol. 1. 13484–13508.

[37] Canwen Xu, Yichong Xu, Shuohang Wang, Yang Liu, Chenguang Zhu, and Julian J. McAuley. 2023. Small Models are Valuable Plug-ins for Large Language Models. *arXiv Preprint* (2023). https://arxiv.org/abs/2305.08848.

[38] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. Video Question Answering via Gradually Refined Attention over Appearance and Motion. In *ACM Multimedia*. 1645–1653.

[39] Haiyang Xu, Qinghao Ye, Ming Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qi Qian, Wei Wang, Guohai Xu, Ji Zhang, Songfang Huang, Fei Huang, and Jingren Zhou. 2023. mPLUG-2: A Modularized Multi-modal Foundation Model Across Text, Image and Video. *arXiv Preprint* (2023). https://arxiv.org/abs/2302.00402.

[40] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *CVPR*. 5288–5296.

[41] Ao Zhang, Hao Fei, Yuan Yao, Wei Ji, Li Li, Zhiyuan Liu, and Tat-Seng Chua. 2023. Transfer Visual Prompt Generator across LLMs. *arXiv Preprint* (2023). https://arxiv.org/abs/2305.01278.

[42] Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding. *arXiv Preprint* (2023). https://arxiv.org/abs/2306.02858.

[43] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pre-trained Transformer Language Models. *arXiv Preprint* (2022). https://arxiv.org/abs/2205.01068.

[44] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. *arXiv Preprint* (2023). https://arxiv.org/abs/2303.18223.

[45] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate Before Use: Improving Few-shot Performance of Language Models. In *ICML*, Vol. 139. 12697–12706.

[46] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. *arXiv Preprint* (2023). https://arxiv.org/abs/2304.10592.