VisLingInstruct: Elevating Zero-Shot Learning in Multi-Modal Language Models with Autonomous Instruction Optimization

Anonymous ACL submission

Abstract

This paper presents VisLingInstruct, a novel approach to advancing Multi-Modal Language Models (MMLMs) in zero-shot learning. Current MMLMs show impressive zero-shot abilities in multi-modal tasks, but their performance depends heavily on the quality of instructions. VisLingInstruct tackles this by autonomously evaluating and optimizing instructional texts through In-Context Learning, improving the synergy between visual perception and linguistic expression in MMLMs. Alongside this instructional advancement, we have also optimized the visual feature extraction modules in MMLMs, further augmenting their responsiveness to textual cues. Our comprehensive experiments on MMLMs, based on FlanT5 and Vicuna, show that VisLingInstruct significantly improves zero-shot performance in visual multi-modal tasks. Notably, it achieves a 15.9% increase in accuracy over the prior state-of-the-art on the ScienceQA dataset.

However, the effectiveness of MMLMs is highly constrained by the quality of textual instructions. Current instruction-tuned models (Ouyang et al., 2022; Zheng et al., 2023b), effective as they may be, while potentially effective, introduces significant challenges, particularly for the users lacking expertise in crafting optimal instructions. The limitation leads to inconsistent and sometimes sub-optimal outputs, thus impeding the practical utility of MMLMs in the real world scenarios. To mitigate this issue, we propose a novel autonomous optimization method for textual instruction, named Visual, Linguistic, Instruction optimization (Vis-LingInstruct). VisLingInstruct introduces an innovative method through In-Context Learning (ICL) (Min et al., 2022) based on the comparison between instruction cases, using MMLMs’ linguistic capabilities to autonomously enhance and evaluate textual instruction. This method can guide the model towards generating more effective and contextually appropriate instructions.

Complementing our instructional optimization strategy, we present an architectural innovation aimed at enhancing the alignment between visual and textual modules within MMLMs. Inspired by recent advancements in models such as MiniGPT4 (Zhu et al., 2023), LLaVA (Liu et al., 2023b), mPLUG-Owl (Ye et al., 2023), and BLIVA (Hu et al., 2023), our architecture enhances the integration of textual and visual data. This integrative approach enables MMLMs to more effectively process and interpret complex tasks that require an understanding of both textual and visual elements, thereby improving accuracy and contextual understanding. Figure 1 offers a visual comparison of the alignment modules in different MMLMs, highlighting the distinctive features and benefits of our proposed method. Through this architectural enhancement, we aim to bridge the existing gaps in multi-modal data processing, creating a more cohesive and efficient model capable of tackling the
nuanced demands of multi-modal interactions.
In summary, our contributions are as follows:

• We introduce substantial architectural improvements for better integration of multi-modal data within MMLMs during training (Section 3.1).

• We present an autonomous method for optimizing instruction quality, tailored to improve the effectiveness of textual instruction during inference (Section 3.2).

• We conduct comprehensive experiments and ablation studies to demonstrate the effectiveness of VisLingInstruct and the success of each component. Notably, VisLingInstruct has improved the performance by a significant margin of 15.9% on the ScienceQA dataset.

2 Related Work

2.1 Instruction Tuning in MMLMs

Instruction tuning has emerged as a cost-effective alternative to the expensive pre-training of large models, focusing on fine-tuning a few foundational models for downstream tasks. In this context, models like InstructGPT (Ouyang et al., 2022), Flan-T5 (Chung et al., 2022), and Vicuna (Zheng et al., 2023b) represent significant strides in conversational models obtained through instruction tuning based on LLMs. These models have showcased exceptional question-answering capabilities, underscoring the importance of instruction-based approaches in language generation. In the multi-modal domain, advancements such as Mini-GPT4 (Zhu et al., 2023), LLaVA (Liu et al., 2023b), mPLUG-Owl (Ye et al., 2023), InstructBLIP (Dai et al., 2023), and BLIVA (Hu et al., 2023) have focused on instruction fine-tuning. These methods typically involve aligning images and text by introducing transitional layers, like Q-Former and fully connected layers, between visual encoders and LLMs. Our work builds upon these foundations, aiming to further optimize the instruction tuning process for enhanced performance in MMLMs.

2.2 Optimizing Instructions for Large Models

Historically, models akin to BERT (Kenton and Toutanova, 2019) have utilized prompt crafting techniques (Brown et al., 2020) to boost performance, with subsequent research exploring methods to discover higher-quality prompts (Gao et al., 2021; Lu et al., 2023). In generative models, this concept has evolved into optimizing 'instructions', leading to a series of works focused on prompt and instruction optimization (Wei et al., 2022; Min et al., 2022). Notably, UPRISE (Cheng et al., 2023) trained a prompt retriever for acquiring superior instructions, while OPRO (Yang et al., 2023) conceptualized LLMs as optimizers, formulating optimization tasks in natural language. Zheng et al. introduced STEP-BACK prompting, enabling LLMs to derive higher-level concepts from detailed instances. To the best of our knowledge, we spearhead the manual-free optimization of textual instruction in zero-shot manner for a wide range of multi-modal tasks.

3 Methods

Our approach comprises two components: First, we refine the architecture of existing multi-modal models and their fine-tuning mechanisms to augment their perceptivity of instruction, that is, the Enhanced Multi-modal Alignment (EMA). Second, subsequent to the model’s fine-tuning, we concentrate on the autonomous optimization of instructions during the inference, referred to as the Aut...
3.1 Enhancing Multi-modal Alignment

In the quest to refine MMLMs, our focus shifts to bridging the gap between the realms of visual perception and linguistic expression. This section delves into our pioneering approach to enhancing the alignment between visual and textual modules within MMLMs, introducing a series of architectural innovations and training optimizations designed to synergize these two distinct modalities seamlessly.

**Integrative Processing of Text and Image:** At the core of our architectural enhancements is the integrative processing of textual and visual data. This process involves constructing a unified representation by merging detailed textual embeddings with rich visual information. We introduce the Cross-Modal Alignment Attention (CMAA) algorithm to achieve this integration, specifically designed to harmonize these disparate data modalities. This algorithm leverages attention mechanisms and cross-modal feature fusion, to ensure that the resulting multi-modal representation encapsulates both the intricacies of language and the finer details of visual content:

\[
U_{mm} = \sum_{i=1}^{N} \text{softmax}(\text{emb}_\text{que} \cdot \text{emb}_\text{text}^T) \cdot \text{emb}_\text{text}(i) \quad (1)
\]

where \(\text{emb}_\text{text}(i)\) and \(\text{emb}_\text{vis}(i)\) represent the embedding of the textual instruction and Queries for the \(i\)-th element respectively. Simultaneously, \(\text{emb}_\text{text}(i)\) serves as both the key (K) and value (V) functions as the query (Q). The textual instruction, after undergoing CMAA, transforms into \(U_{mm}\). Subsequently, \(U_{mm}\) concatenate onto the output of Queries in the form of Figure 1, culminating in the final integration of visual and textual elements.

**Optimized Model Training and Performance:** In developing this new architecture, our approach extends beyond mere technical integration to encompass strategic training and performance optimization. We employ selective weight freezing methods, where specific layers of the pre-trained model are kept static to preserve learned features, and targeted fine-tuning, where newly introduced components or layers are specifically trained to adapt to the task at hand. This targeted approach allows us to fine-tune the model’s performance without the need for extensive retraining, thereby enhancing the learning efficiency and ensuring the robustness and scalability of the model. The loss function used for training takes the following form:

\[
p(Y_{text}|X_{img}) = \prod_{i=1}^{L} p(y_i|X_{img}, Y_{text}^{[1:i-1]}) \quad (2)
\]

where \(\theta\) is the trainable parameters, \(X_{img}\) and \(Y_{text}\) respectively denote the input image and the output text, \(Y_{text}^{[1:i-1]}\) represents the input instruction and the text already generated up to the \(i-1\) step.
<table>
<thead>
<tr>
<th>Initial Instruction</th>
<th>Rewritten Instruction</th>
<th>Optimized Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>I have some texts along with their corresponding scores. The text is arranged in descending order according to the score, where a lower score indicates better quality.</td>
<td>Please write a better text based on these text-score pairs.</td>
<td></td>
</tr>
<tr>
<td>Text: What is remarkable about this particular image? Score: 2.4442</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text: What is unpalatable about this soup? Score: 2.4442</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text: Does this scene look similar to the others? Score: 2.4442</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text: What stands out or distinctive in this stage? Score: 2.4442</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 3:** The examples of IAS ranking in different domains. On the left side is the image input provided to the MMLMs. On the right side, within the blue box, lies the initial instruction, while the rewritten instruction is contained within the green box. The ‘score’ indicates the quality of corresponding instructions with respect to the model, while the lower score (i.e., high quality) instruction always ranks later in the ICL demonstration. By utilizing the paradigm of ICL, MMLMs learn the relationship between the scores of the two cases to generate higher-quality new instructions that lie in the yellow box.

### 3.2 Autonomous Instruction Optimization

During inference, the textual instruction has a significant impact on the generation results of MMLMs. Therefore, we propose an approach that leverages the inherent text processing capabilities of LLMs to optimize textual instructions, thereby aligning the results generated by MMLMs more closely with user requirements. This method comprises two stages: Rewriting Textual Instructions and Instruction Alignment Optimization.

**Rewriting Textual Instructions:** Rewriting is the first stage in our methodology. LLMs exhibit strong text rewriting capabilities, maintaining semantic information and changing the content of the text. Therefore, our goal is to use the ability of LLM to rewrite the initial textual instructions, aiming to obtain a pair of instruction with roughly similar semantics to lay the foundation for the second stage. Furthermore, the rewritten instruction resulting from this process is not necessarily required to be of higher quality than the initial instruction. As long as there is a difference between the two, it is enough to meet the requirements of subsequent processes. This reduces the complexity of the rewrite task, making the barrier to implementation relatively low.

Specifically, we have intricately designed a prompt tailored for LLMs to rewrite textual instruction. The prompt should explicitly instruct LLMs on how to rewrite the textual instruction while ensuring minimal semantic changes between the initial and rewritten version. The template of the prompt used in this stage can be referred to in the Appendix B. An important consideration to note is that since this stage solely involves rewriting instructions, the entire MMLMs aren’t required. Involving only LLMs could slightly reduce the time consumption caused by the rewriting process.

**Instruction Comparison Optimization:** At this stage, we design a method that enables MMLMs to identify which instruction is better through comparative analysis and strive to generate instruction with better quality. As illustrated in Figure 2, we innovatively rank the cases in ICL so that the model can learn the quality of instructions only through the comparison between initial instruction and rewritten instruction (Ren and Liu, 2023).

Given that the ultimate goal of the instruction is to facilitate inference by MMLMs, we believe that the quality of these instructions should be evaluated by the MMLMs themselves. Specifically, we allow MMLMs to score the instruction solely by the itself without the help of external discriminator. Therefore, we proposed the Instruction Alignment Score (IAS), which is formulated to quantify the divergence between the model’s predicted output and the expected output for a given instruction. We score the instruction by using a prompt

$$\text{Score} = \text{ptask} - \text{pquery}$$

where \( \text{ptask} \) is the prediction of the instruction task and \( \text{pquery} \) is the prediction of the question task.
to guide MMLMs. Please see the Appendix C for the prompt template. Defined as the expectation of negative log-likelihood, IAS is calculated as follows:

\[
IAS = E[-\log P(t_i | X_{\text{img}}, X_{\text{prompt}}, t_{i-1}; \theta)]
\]

(3)

Here, \(X_{\text{img}}\) is the input image, \(X_{\text{prompt}}\) denotes textual prompt employed to guide the model in its computational processes, \(\theta\) represents our MMLMs model and \(t_i\) represents the textual tokens that need to be generated, which is the instruction whose quality MMLMs is tasked to evaluate. A lower IAS indicates a higher alignment of the instruction with the model’s understanding, enabling MMLMs to perform better. After calculating IAS, as shown in Figure 3, we rank the instruction-IAS pairs in descending order, and combine them into a prompt in the form of ICL to input to MMLMs to generate an optimized instruction. The optimized instruction will have better inference performance compared to the initial and rewritten instructions.

4 Experiments

4.1 Datasets

The datasets in this paper primarily consists of a training dataset and the zero-shot evaluation benchmarks. The training data is sourced from LLaVA, which is also a subset of the InstructBLIP training datasets. The data was collected by the authors of LLaVA using ChatGPT/GPT-4 (OpenAI, 2023a,b), following a multi-modal instruction format. We believe that using the same dataset as previous work enables a fairer comparison in the experiments. For an image \(X_v\), there is an associated question-answer pair \(<X_q, X_a>\) related to \(X_v\). In some cases, there are multi-turn dialogues represented as \(<X^1_q, X^1_a>,...<X^m_q, X^m_a>\). During training, for single-turn dialogue data, \(X_q\) serves as the initial instruction, while \(X_a\) corresponds to the ground truth. Likewise, for multi-turn dialogue data, it is essential to concatenate the historical dialogues (excluding the last turn) and append them along with \(X^q_a\) as the initial instruction. Meanwhile, \(X^a_m\) serves as the ground truth.

For zero-shot evaluation benchmarks, to ensure alignment for comparison, we also follow InstructBLIP. The evaluation domains include: Image captioning: Flickr30K (Young et al., 2014), NoCaps (Agrawal et al., 2019). Visual Reasoning: VSR (Liu et al., 2023a), GQA (Hudson and Manning, 2019), IconQA (Lu et al., 2021). Image QA: VizWiz (Gurari et al., 2018), TextVQA (Mishra et al., 2019). Comprehensive VQA: Visual Dialog (Das et al., 2017), ScienceQA (Lu et al., 2022), HatefulMemes (Kiela et al., 2020). It’s important to note that for ScienceQA, we only evaluate the set with image context. The utilization of the overall evaluation benchmarks can be referenced in Appendix D. The evaluation metrics vary across benchmarks: NoCaps and Flickr30K employ CIDEr scores (Vedantam et al., 2015), HatefulMemes utilizes AUC scores, and Visual Dialog employs Mean Reciprocal Rank (MRR). For all remaining datasets, top-1 accuracy is used as the metric. All evaluation benchmarks have no data overlap with the training set, ensuring the authenticity of zero-shot. In the Appendix F, we provide the initial instructions used for all benchmarks in zero-shot learning.

4.2 Implementation details

In terms of the model architecture, we opted for the ViT-G/14 from EVA-CLIP (Fang et al., 2023) as the visual encoder, removing the final layer of the ViT and utilizing the output features from the penultimate layer. In line with InstructBLIP, we employed two distinct LLMs: FlanT5 and Vicuna. FlanT5, derived from the instruction-tuning of the encoder-decoder Transformer T5 (Raffel et al., 2020), encompasses two sizes: FlanT5-XL and FlanT5-XXL. Vicuna, on the other hand, is refined from the instruction-tuning of the decoder-only Transformer LLaMA (Touvron et al., 2023), and also includes two sizes: Vicuna-7B and Vicuna-13B. The weights of both Q-Former and the fully connected layers are sourced from InstructBLIP and need to correspond to different LLMs. Our entire model framework requires freezing the weights of the visual encoder, Q-Former, and LLMs, allowing only the fully connected layers to be unfrozen. Further details regarding training hyperparameters can be found in Appendix E.

4.3 Zero-shot Evaluation

During the evaluation process, we employed two different generation methods tailored to different benchmarks. For the domain of benchmarks such as Image Captioning, results were directly generated from instructions. These results were then compared against ground truth to calculate metrics. On the other hand, for classification-based VQA
As shown in Figure 4, we have enumerated three cases for comprehensive analysis. The process incorporates real-world images and instructions. Beyond the benchmarks-driven experimental analyses, we diversified our qualitative evaluation by analyzing Image Captioning Visual Reasoning Image QA Comprehensive VQA
Flickr30K Nocaps VSR GQA IconQA VizWiz TextVQA Visdial SciQA HM

### Table 1: Zero-shot results on general VQA benchmarks
Here, Visdial, SciQA, and HM respectively refer to Visual Dialog, ScienceQA, and HatefulMemes. The results for MiniGPT-4 and LLaVA are sourced from BLIVA (Hu et al., 2023), while the remaining results originate from their respective papers (Li et al., 2023; Dai et al., 2023).

<table>
<thead>
<tr>
<th>Model</th>
<th>Flickr30K</th>
<th>Nocaps</th>
<th>VSR</th>
<th>GQA</th>
<th>IconQA</th>
<th>VizWiz</th>
<th>TextVQA</th>
<th>Visdial</th>
<th>SciQA</th>
<th>HM</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLIP-2 (FlanT5_{XXL})</td>
<td>73.7</td>
<td>104.5</td>
<td>68.2</td>
<td>44.6</td>
<td>45.4</td>
<td>29.4</td>
<td>44.1</td>
<td>46.9</td>
<td>64.5</td>
<td>52.0</td>
</tr>
<tr>
<td>BLIP-2 (Vicuna_{13B})</td>
<td>74.9</td>
<td>107.5</td>
<td>50.9</td>
<td>41.0</td>
<td>40.6</td>
<td>19.6</td>
<td>42.5</td>
<td>45.1</td>
<td>61.0</td>
<td>53.7</td>
</tr>
<tr>
<td>MiniGPT-4 (Vicuna_{13B})</td>
<td>/</td>
<td>/</td>
<td>50.7</td>
<td>30.8</td>
<td>37.6</td>
<td>34.8</td>
<td>18.7</td>
<td>/</td>
<td>/</td>
<td>29.0</td>
</tr>
<tr>
<td>LLaVA (Vicuna_{13B})</td>
<td>/</td>
<td>/</td>
<td>56.3</td>
<td>41.3</td>
<td>43.0</td>
<td>37.7</td>
<td>28.3</td>
<td>/</td>
<td>/</td>
<td>9.2</td>
</tr>
<tr>
<td>InstructBLIP (FlanT5_{XXL})</td>
<td>84.5</td>
<td>119.9</td>
<td>64.8</td>
<td>48.4</td>
<td>50.0</td>
<td>32.7</td>
<td>46.6</td>
<td>46.6</td>
<td>70.4</td>
<td>56.6</td>
</tr>
<tr>
<td>InstructBLIP (FlanT5_{XXL})</td>
<td>83.5</td>
<td>120.0</td>
<td>65.6</td>
<td>47.9</td>
<td>51.2</td>
<td>30.9</td>
<td>46.6</td>
<td>48.5</td>
<td>70.6</td>
<td>54.1</td>
</tr>
<tr>
<td>InstructBLIP (Vicuna_{7B})</td>
<td>82.4</td>
<td>123.1</td>
<td>54.3</td>
<td>49.2</td>
<td>43.1</td>
<td>34.5</td>
<td>50.1</td>
<td>45.2</td>
<td>60.5</td>
<td>59.6</td>
</tr>
<tr>
<td>InstructBLIP (Vicuna_{13B})</td>
<td>82.8</td>
<td>121.9</td>
<td>52.1</td>
<td>49.5</td>
<td>44.8</td>
<td>33.4</td>
<td>50.7</td>
<td>45.4</td>
<td>63.1</td>
<td>57.5</td>
</tr>
<tr>
<td>BLIVA (Vicuna_{13B})</td>
<td>87.1</td>
<td></td>
<td>62.2</td>
<td>44.9</td>
<td></td>
<td>42.9</td>
<td>58.0</td>
<td>45.6</td>
<td></td>
<td>55.6</td>
</tr>
<tr>
<td>BLIVA (FlanT5_{XXL})</td>
<td>87.7</td>
<td></td>
<td>68.8</td>
<td>52.4</td>
<td></td>
<td>44.0</td>
<td>57.2</td>
<td>36.2</td>
<td></td>
<td>50.0</td>
</tr>
<tr>
<td>Ours(FlanT5_{13B})</td>
<td>85.3</td>
<td>119.5</td>
<td>64.1</td>
<td>47.9</td>
<td>50.4</td>
<td>33.0</td>
<td>48.7</td>
<td>47.0</td>
<td>71.0</td>
<td>60.0</td>
</tr>
<tr>
<td>Ours(FlanT5_{XXL})</td>
<td>88.5</td>
<td>120.4</td>
<td>66.9</td>
<td>48.1</td>
<td>51.2</td>
<td>31.3</td>
<td>48.8</td>
<td>49.2</td>
<td>81.8</td>
<td>55.7</td>
</tr>
<tr>
<td>Ours(Vicuna_{7B})</td>
<td>87.9</td>
<td>124.2</td>
<td>60.1</td>
<td>52.0</td>
<td>44.2</td>
<td>42.7</td>
<td>60.6</td>
<td>45.7</td>
<td>74.6</td>
<td>62.7</td>
</tr>
<tr>
<td>Ours(Vicuna_{13B})</td>
<td>84.0</td>
<td>119.8</td>
<td>56.2</td>
<td>52.9</td>
<td>50.3</td>
<td>45.0</td>
<td>65.6</td>
<td>45.7</td>
<td>71.0</td>
<td>58.9</td>
</tr>
</tbody>
</table>

### 4.4 Ablation study
To investigate the impact of Enhancing Multi-modal Alignment (EMA, Section 3.1) and Autonomous Instruction Optimization (AIO, Section 3.2) on the final results, we conducted ablation studies by individually removing them during evaluation.

- **EMA** demonstrates that the instruction mechanism on the vanilla baseline, the overall performance of all models is significantly enhanced.
- This indicates that our EMÀ method indeed enhances the alignment between images and text.
- Moreover, if AIO continues to be integrated on the basis of EMA, the evaluation results can be further improved. This adequately shows that the two mechanisms can strengthen each other.

### 4.5 Qualitative evaluation
Beyond the benchmarks-driven experimental analyses, we diversified our qualitative evaluation by incorporating real-world images and instructions. As shown in Figure 4, we have enumerated three cases for comprehensive analysis. The process...
How should I cook this? What is needed? What is the appropriate method to prepare this dish? Could you specify the essential ingredients?

To cook the spaghetti and meat sauce, you will need the following ingredients: spaghetti, ground beef, onions, garlic, canned tomatoes, tomato sauce, salt, pepper, and olive oil. You can start by browning the ground beef in a pan, then add the onions and garlic, followed by the canned tomatoes, tomato sauce, salt, pepper, and olive oil. Once the sauce is heated through, you can add the cooked spaghetti to the pan and toss everything together until the spaghetti is fully coated with the sauce. Finally, you can serve the spaghetti and meat sauce hot on a plate, garnished with a sprig of parsley or basil if desired.

Vanilla Response:
Introducing our new line of cat-themed mugs! Perfect for any cat lover, these mugs come in a variety of colors and designs. Whether you’re drinking your morning coffee or enjoying a cup of tea, our cat mugs are sure to bring a smile to your face. Order now and show off your love for cats with every sip.

EMA Response:
Introducing our new line of cat-themed mugs! Perfect for any cat lover, these mugs come in a variety of colors and designs to match your personal style. Whether you’re drinking your morning coffee or enjoying a cup of tea, our cat mugs will add a touch of whimsy and charm to your daily routine. Each mug is made with high-quality ceramic materials, ensuring durability and long-lasting use. Get yours today and show off your love for cats!

EMA & AIO Response:
Introducing our newly designed mug featuring a cute cat design, perfect for any coffee or tea lover. The mug is made of high-quality ceramic material, ensuring that it is durable and long-lasting. The unique design of the cat on the mug makes it an eye-catching addition to any kitchen or coffee table. Whether you’re drinking your morning coffee or enjoying a cup of tea in the evening, this mug will add a touch of whimsy and charm to your daily routine. So why settle for a boring mug when you can have one that’s both functional and adorable? Get yours today and start enjoying your favorite beverages in style!

Figure 4: The one on the left is a case written for a product advertisement, the one in the middle is a recipe description, and the one on the right is a poetry creation. Qualitative comparison of three responses from different ablations: initial instruction with vanilla model (blue), initial instruction with EMA model (purple), and optimized instruction with EMA model (green).
Table 2: Results of ablation studies for Enhancing Multi-modal Alignment (EMA) and Autonomous Instruction Optimization (AIO) in different LLMs models. Among them, EMA is split into Rewriting Textual Instructions (Rewriting) and Instruction Comparison Optimization (Comparison) for discussion respectively. Vanilla represents the baseline model without any of our proposed modules and ✓ indicates that the module has been integrated.

<table>
<thead>
<tr>
<th>Vanilla</th>
<th>EMA Rewriting</th>
<th>AIO Comparison</th>
<th>Image Captioning</th>
<th>Visual Reasoning</th>
<th>Image QA</th>
<th>Comprehensive VQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlaxT5-XL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84.5</td>
<td>119.9</td>
<td>64.8</td>
<td>48.4</td>
<td>50.0</td>
<td>32.7</td>
<td>46.6</td>
</tr>
<tr>
<td>FlaxT5-XXL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85.1</td>
<td>119.7</td>
<td>63.5</td>
<td>48.6</td>
<td>50.0</td>
<td>32.8</td>
<td>48.5</td>
</tr>
<tr>
<td>Vicuna-7B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84.7</td>
<td>118.1</td>
<td>66.8</td>
<td>48.5</td>
<td>49.0</td>
<td>31.8</td>
<td>47.5</td>
</tr>
<tr>
<td>Vicuna-13B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85.3</td>
<td>119.5</td>
<td>64.1</td>
<td>47.9</td>
<td>50.4</td>
<td>33.0</td>
<td>48.7</td>
</tr>
</tbody>
</table>

commences with the input of an image, subsequent questions and answers revolve around this visual context. This is followed by the presentation of instructions, encompassing both the initial instructions and the optimized by the AIO module. Conclusively, model response is delineated. The output section for evaluation includes: the results obtained by inputting the initial instructions into the vanilla model (Vanilla Response); the results obtained by inputting the initial instructions into the integrated EMA module model (EMA Response); and the results from inputting the optimized instructions into the integrated EMA module model (EMA & AIO Response), which is VisLingInstruct.

The outcome as observed in the figure suggests that the EMA Response demonstrates an improvement over the Vanilla Response, both in terms of content accuracy and richness of detail. For instance, within the case of poetry creation, the erroneously presented ‘3 huts’ is accurately identified as ‘a small house’. In the case of recipe description, the narrative about spaghetti is much more detailed in the EMA Response. Furthermore, the EMA & AIO response also surpasses the EMA response alone, evident in the former’s answers possessing a superior logical organization and better fulfillment of user intent. This is well illustrated in all three cases presented in the figure. And for more on the performance in multi-turn dialogues, we have provided a demonstration and discussion in the Appendix G.

5 Conclusion

This paper proposes VisLingInstruct, a novel autonomous instruction optimization framework for visual-linguistic multi-modal models. We conducted a comprehensive study on multi-modal models and demonstrated the powerful autonomous instruction optimization capabilities of the VisLingInstruct model, demonstrating strong zero-shot learning capabilities in a series of benchmarks. At the end of the experiment, qualitative examples were used to demonstrate the specific situation of VisLingInstruct in autonomous instruction optimization, such as knowledge-based image description, image-based text creation and multi-turn dialogue. We hope that VisLingInstruct can inspire more new research on autonomous optimization of multi-modal instruction.
Limitations

Limitations of the current work are discussed below. Firstly, while the proposed multimodal autonomous instruction optimization framework performs well in terms of effectiveness, its structure is relatively intricate. Streamlining the process while maintaining effectiveness would greatly facilitate the practical application of this technology. Secondly, the experiments in this paper are focused on image and text modalities, and further validation is needed to determine the effectiveness of our framework in other modalities, such as video and audio.

References


A Algorithm

A.1 Algorithm Overview

The algorithmic core of our approach in VisLingInstruct is structured around two main processes: Cross-Modal Alignment Attention and Autonomous Instruction Optimization. The former process harmonizes the integration of text and image, while the latter refines the textual instructions for MMLMs.

A.2 Cross-Modal Alignment Attention

The Cross-Modal Alignment Attention (CMAA) algorithm focuses on the integration of textual and visual embeddings, creating a unified text representation.

**Algorithm 1 Cross-Modal Alignment Attention**

Require: Textual embeddings $E_{text}$, Queries embeddings $E_{que}$
Ensure: Unified multi-modal representation $U_{mm}$
1: Initialize cross-modal alignment mechanism
2: for each element $i$ in $E_{text}$ do
3: Compute attention between $E_{text}(i)$ and $E_{que}$
4: Assign attention weight on $E_{text}(i)$
5: end for
6: $U_{mm} \leftarrow$ Aggregate of aligned and weighted $E_{text}$
return $U_{mm}$

A.3 Autonomous Instruction Optimization

The Autonomous Instruction Optimization (AIO) is designed to transform initial instruction into an optimized format.

**Algorithm 2 Autonomous Instruction Optimization**

Require: Initial instructions $I_i$
Ensure: optimized instruction $I_{opt}$
1: Initialize autonomous instruction optimization
2: Rewriting the initial instruction $I_i$ to obtain $I_j$
3: Calculating the IAS for $I_i$ and $I_j$
4: Ranking the instruction-IAS pairs
5: $I_{refined} \leftarrow$ Constructing the prompt input for Instruction Comparison in MMLMs

B Instruction rewriting templates

Here is the template used for Instruction rewriting in this paper, where ‘{}’ signifies the instruction that requires modification:

There is the text {}. Please modify the text to make it better while retaining the sentence structure and keywords.

C IAS templates

In the following prompt template, {} is used to place instructions requiring MPG calculation.

<Image>Based on the image given, the most appropriate instruction should be: {}

D Zero-shot evaluation datasets details

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>Part</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flickr30K</td>
<td>test</td>
<td>1000</td>
</tr>
<tr>
<td>NoCaps</td>
<td>val</td>
<td>4500</td>
</tr>
<tr>
<td>VSR</td>
<td>test</td>
<td>1222</td>
</tr>
<tr>
<td>GQA</td>
<td>test-dev</td>
<td>12578</td>
</tr>
<tr>
<td>IconQA</td>
<td>test</td>
<td>6316</td>
</tr>
<tr>
<td>VizWiz</td>
<td>test-dev</td>
<td>4319</td>
</tr>
<tr>
<td>TextVQA</td>
<td>val</td>
<td>5000</td>
</tr>
<tr>
<td>Visual Dialog</td>
<td>val</td>
<td>2064</td>
</tr>
<tr>
<td>ScienceQA</td>
<td>test</td>
<td>2017</td>
</tr>
<tr>
<td>HatefulMemes</td>
<td>val</td>
<td>1040</td>
</tr>
</tbody>
</table>

Table 3: The selected part in all zero-shot evaluation benchmarks, and accompanied by specific data count.

E Training details

We implement VisLingInstruct by LAVIS library (Li et al., 2022). We fine-tuned the fully connected layers for 3 epochs, employing different hyperparameters across distinct LLMs. We employ a batch size of 32, 128 and 256 for the Vicuna-7B/13B, FlanT5-XL and FlanT5-XXL, respectively. For each model, we conduct validation every 1K steps. A consistent aspect across our training procedures was the utilization of the AdamW (Loshchilov and Hutter, 2018) optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$, and a weight decay of 0.05. Furthermore, we implemented a linear warm-up of the learning rate over the initial 1K steps, escalating from $10^{-8}$ to $10^{-5}$, followed by cosine decay towards a minimum learning rate of 0.

All our model’s trainable parameter counts are maintained within the range of a few million. Under the conditions of 8 A100 40G, the training durations for FlanT5, Vicuna 7B, and Vicuna 13B
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Initial instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flickr30K/NoCaps</td>
<td>&lt;Image&gt;A short image description:</td>
</tr>
<tr>
<td>VSR</td>
<td>&lt;Image&gt;Based on the image, is this statement true or false? {}</td>
</tr>
<tr>
<td>GQA/Visdial</td>
<td>&lt;Image&gt;Question: {} \ Short answer:</td>
</tr>
<tr>
<td>IconQA</td>
<td>&lt;Image&gt;Question: {} Options: {} \n Answer:</td>
</tr>
<tr>
<td>VizWiz</td>
<td>&lt;Image&gt;Answer the question based on the image. Reply in one phrase/word or say ‘unanswerable’. Question: {} \n Short answer:</td>
</tr>
<tr>
<td>TextVQA</td>
<td>&lt;Image&gt;OCR tokens: {} Question: {} \n Short answer:</td>
</tr>
<tr>
<td>SciQA</td>
<td>&lt;Image&gt;Given the image, choose the correct option for the following question. Question: {} \n Options: {}</td>
</tr>
<tr>
<td>HM</td>
<td>&lt;Image&gt;This is an image with: {} written on it. Is it hateful?</td>
</tr>
</tbody>
</table>

Table 4: Presentation of initial instructions for each benchmark.

are 105 minutes, 135 minutes, and 210 minutes, respectively.

F Instructions in benchmarks

Table 4 illustrates the initial instructions for all benchmarks. The initial instructions were predominantly sourced from InstructBLIP. ‘{}’ contains entities such as questions from each individual case. For instructions with options, we separate the choices alphabetically, for instance: (a) apple (b) banana (c) pineapple.

G Multi-turn dialogues case studies

As illustrated in Figures 5 and 6, we present additional cases regarding multi-turn dialogue to showcase the strengths and distinctive features of our approach. The multi-turn generation results on the right originate from the models incorporating the EMA and AIO proposed in this paper (VisLingInstruct), while those on the left exclude this aspect, representing the results from Vanilla Response.
Can you give me a short introduction about the movie?

Titanic was released in 1997 and became one of the highest-grossing movies of all time, grossing over $2 billion worldwide.

Can you recommend me some other similar movies?


Figure 5: This multi-turn dialogue example demonstrates the effectiveness of VisLingInstruct in movie introduction and recommendation. Our generated results (right) exhibit significantly improved accuracy in identifying instructional content and richness in answers compared to the vanilla model’s results (left).
How should I make this food?

To make this meal, you would need the following ingredients: steak, broccoli, tomatoes, and dipping sauce. First, cook the steak to your desired level of doneness. Then, chop the broccoli and tomatoes into bite-sized pieces. Finally, serve the steak with the vegetables and dipping sauce on a plate.

Is there anything I need to pay attention to?

Yes, it's important to pay attention to the cooking time and temperature of the steak to ensure that it is cooked to your desired level of doneness. Overcooking or undercooking the steak can affect its texture, taste, and overall quality. To avoid this, use a meat thermometer to check the internal temperature of the steak and adjust the cooking time accordingly.

Are there any foods that go well with this dish?

Yes, there are several foods that go well with this steak and vegetable dish. Here are a few suggestions:

1. Rice or pasta: Adding a side of rice or pasta to the meal can provide additional carbohydrates and complement the flavors of the steak and vegetables.
2. Potatoes: Roasted or mashed potatoes can be a great addition to the meal, especially if you prefer a heartier dish.
3. Salad: Adding a salad made with greens, vegetables, and a light vinaigrette dressing can provide a refreshing contrast to the rich flavors of the steak and dipping sauce.
4. Bread or rolls: Serving the steak and vegetables with a side of bread or rolls can make for a satisfying sandwich-style meal.

The best approach to cooking this dish, which consists of a steak, broccoli, and tomatoes, would be to ensure that the ingredients are cooked evenly and thoroughly. This can be achieved by using a variety of cooking techniques, such as grilling, pan-frying, or oven-roasting the steak. For the broccoli and tomatoes, they can be sautéed or steamed to retain their texture and flavor. Additionally, it's important to season the dish with appropriate spices or sauces to enhance its taste and make it more appealing to the diner.

When considering key factors, what should I be particularly aware of?

When considering key factors for cooking the dish consisting of a steak, broccoli, and tomatoes, there are several aspects to keep in mind:

1. Timing: Ensure that each ingredient is cooked at the appropriate time to avoid overcooking or undercooking any of them. The steak should be cooked to the desired level of doneness, while the broccoli and tomatoes should be cooked just enough to retain their texture and flavor.
2. Temperature: Maintaining a consistent temperature throughout the cooking process is crucial to ensure that the ingredients are cooked evenly and thoroughly. This can be achieved by using a thermometer to monitor the temperature of the grill, pan, or oven during cooking.
3. Seasoning: Properly seasoning the dish with appropriate spices or sauces can enhance its taste and make it more appealing to the diner. Experiment with different combinations of herbs, spices, and sauces to find the perfect blend that complements the flavors of the steak, broccoli, and tomatoes.
4. Food pairings: Some ideal food pairings for the dish consisting of a steak, broccoli, and tomatoes include:
   1. Red wine: A full-bodied red wine, such as Cabernet Sauvignon or Merlot, can complement the rich flavors of the steak and enhance the overall taste of the dish.
   2. Potatoes: Roasted or mashed potatoes can be a great side dish to balance the flavors and textures of the steak, broccoli, and tomatoes.
   3. Green salad: Adding a simple green salad with a light vinaigrette dressing can provide a refreshing contrast to the rich flavors of the steak and vegetables.
   4. Bread: A crusty bread or roll can be used to soak up the juices from the steak and vegetables, adding an extra layer of flavor and texture to the dish.

Figure 6: This is a multi-turn dialogue about cooking topics. The two generated answers exhibit similar accuracy in content recognition. However, in terms of richness and the final food pairing, our response (right) surpasses the vanilla model’s response (left) significantly.