Template Matters: Understanding the Role of Instruction Templates in Multimodal Language Model Evaluation and Training

Anonymous authors
Paper under double-blind review

Abstract

Current multimodal language models (MLMs) evaluation and training approaches overlook the influence of instruction format, presenting an elephant-in-the-room problem. Previous research deals with this problem by manually crafting instructions, failing to yield significant insights due to limitations in diversity and scalability. In this work, we propose a programmatic instruction template generator capable of producing over 3.9B unique template combinations by filling randomly sampled positional synonyms into weighted sampled meta templates, enabling us to comprehensively examine the MLM's performance across diverse instruction templates. Our experiments across eight common MLMs on five benchmark datasets reveal that MLMs have high template sensitivities with at most 29% performance gaps between different templates. We further augment the instruction tuning dataset of LLaVA-1.5 with our template generator and perform instruction tuning on LLaVA-1.5-7B and LLaVA-1.5-13B. Models tuned on our augmented dataset achieve the best overall performance when compared with the same scale MLMs tuned on at most 75 times the scale of our augmented dataset, highlighting the importance of instruction templates in MLM training.

1 Introduction

Multimodal Language Models (MLMs) have revolutionized vision-language learning by performing visual instruction tuning on diverse, high-quality multimodal instruction data Liu et al. (2024c); Zhu et al. (2023); Laurençon et al. (2024); Li et al. (2024); Zhang et al. (2024b). MLMs achieve unprecedented performance on various visual tasks Lin et al. (2024); Luo et al. (2024); Ma et al. (2024); Xue et al. (2024b); Wang et al. (2024b). However, previous MLM evaluation and training methods overlook a significant *elephant-in-the-room* problem: different instruction formats will largely influence MLMs' performance. Although recent studies Zhang et al. (2024a); Xie et al. (2024); Liu et al. (2024e); Sclar et al. (2023) demonstrate that MLMs may produce distinct outputs when changing the instruction format (as shown in Figure 1), research on MLMs' sensitivity to instruction formats remains largely unexplored. Previous works designed hand-crafted instructions in a limited amount, which restricts the evaluation scale, thereby weakening their conclusions, and limiting the opportunity to finetune the MLMs with augmentation on different instruction formats.

To systematically investigate the instruction sensitivity of MLMs and their impact on faithful evaluation, we propose to evaluate MLMs on Visual Question Answering (VQA) data augmented by various instruction templates without changing the meaning of the original QA pairs. To efficiently create diverse, high-quality instruction templates in sufficient quantities, we introduce a *programmatic template generator* that leverages diverse meta templates to produce semantically equivalent instruction templates automatically and scalably. Our approach can construct diverse instruction templates by random sampling from carefully curated word and phrase spaces to populate predefined placeholders, enabling the efficient generation of semantically consistent yet diverse instruction templates at scale. Our method can produce an extensive template space comprising 15K visual question templates and 249K choice-related templates, culminating in a comprehensive VQA instruction template space of 3.9B unique combinations. To effectively manage this vast template space, we use a tree-based organizational framework based on grammatical structures complemented by an efficient diversity sampling algorithm. This programmatic approach ensures the generation of instruction

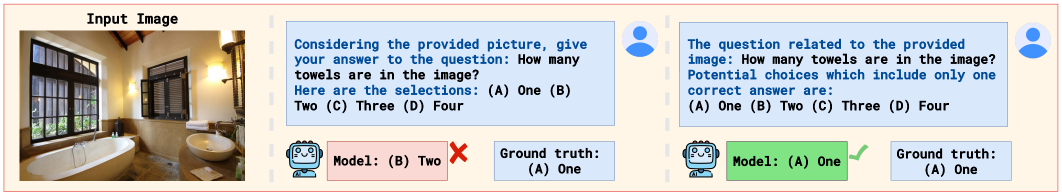
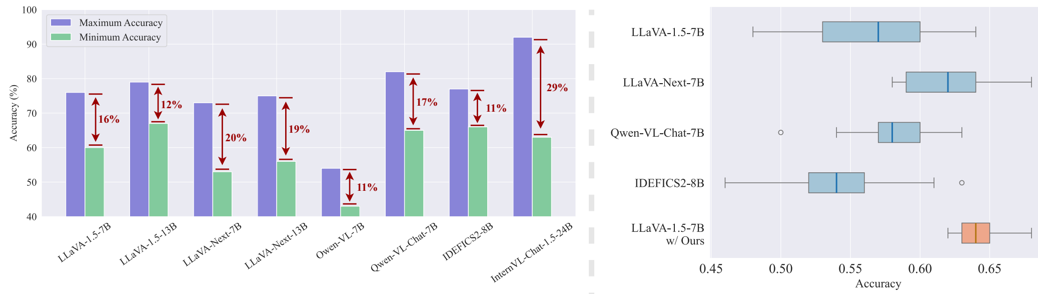


Figure 1: An example of using different instruction templates to prompt MLM without changing the original QA pairs. The instruction templates are marked in **blue**. **Prompting MLM with different instruction templates can twist the output of MLM**.



(a) Evaluation across diverse instruction templates

(b) Instruction-tuning with template augmentation

Figure 2: **The left (a) illustrates the high sensitivity of Multimodal Language Models (MLMs) to variations in instruction templates.** We compare the best and worst accuracy of eight prominent MLMs across 100 different instruction templates on the MMBench dataset. The accuracy gaps are marked in **red bold**; **The right (b) shows that visual instruction tuning with diverse instruction templates significantly improves MLM's performance and reduces the performance variance.** LLaVA-1.5-7B trained with diverse instruction templates achieves the highest average performance and the lowest performance variance among similar-scale MLMs on the SeedBench dataset, evaluated across 25 instruction templates that are not included in the training.

templates that maximize diversity across multiple dimensions, including grammatical construction, lexical choice, and symbolic representation.

We conduct a comprehensive robustness evaluation of instruction templates with our programmatic template generator, encompassing eight commonly used MLMs. Our experiment results reveal that those MLMs are highly sensitive to instruction template perturbation, with at most 29% performance gap across 100 different templates. We present the performance gap across instruction templates on the MMBench dataset in Figure 2(a). Given these results, we further introduce a simple yet effective method to improve visual instruction tuning that leverages our template generator to augment instruction datasets. We finetune two common MLMs (LLaVA-1.5-7B and LLaVA-1.5-13B) Liu et al. (2024a) using our generated diverse instruction templates and compare them with other MLMs finetuned on a larger scale (at most 75.19x than ours) of instruction-tuning datasets. Our finetuned MLMs achieve the best overall performance, demonstrating our method's capability to improve MLMs in a data-efficient and low-cost way. We show the comparison of our 7B model to other models of similar scale on the SeedBench dataset in Figure 2(b). Our analysis further shows that compared to the original model, after finetuning with our template augmented instruction data, the model's variance drops significantly on various out-of-domain instruction templates, which are not included in the training. Our approach not only validates the practical utility of our template generation framework but also illuminates promising directions for efficiently improving MLMs. On the other hand, our ablation studies show that models achieve the best general capabilities at a specific ratio between templates and training data, which varies with the model scale. We summarize our main contributions as follows.

- We introduce a novel programmatic instruction template generator that enables fast and scalable generation of diverse, semantically equivalent instruction templates.

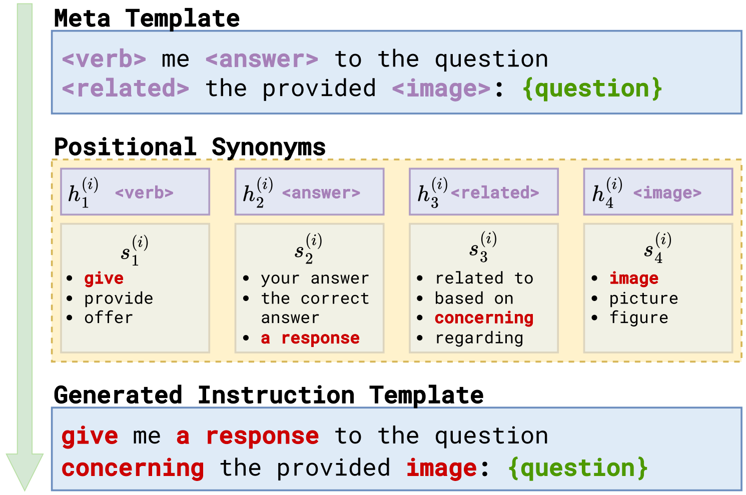


Figure 3: Example of the instruction template generation through a meta template.

- We evaluate the robustness of eight commonly used MLMs to instruction format variations across five benchmarks leveraging our template generator, revealing their high sensitivity to instruction format variations.
- We propose a simple yet effective approach to enhance visual instruction tuning by augmenting the origin instruction-tuning dataset with programmatically scaling instruction templates. Our extensive experiments demonstrate its effectiveness.

2 Programmatically Scaling Instruction Templates

In this work, we propose a programmatic instruction template generator that efficiently produces a diverse array of grammatically correct and semantically consistent instruction templates without modifying the original Question-Answer (QA) pairs. Specifically, we generate instruction templates by programmatically populating placeholders in diverse *meta templates* with randomly sampled positional synonyms (phrases) to ensure flexibility while keeping the original meaning (Sec. 2.1). We organize our meta templates in a *sentence pattern tree* (Sec. 2.2), along with weighted sampling to ensure the sampling probability across all meta templates is uniformly distributed.

2.1 Meta Templates

A meta template $p_i, i \in \{1, ..., N\}$, serves as a formal blueprint for constructing instruction templates, consisting of a sequence of fixed string segments interspersed with placeholder $\langle h_j^{(i)} \rangle, j \in \{1, ..., M_i\}$, where M_i is the number of placeholders. We associate each placeholder $\langle h_j^{(i)} \rangle$ with a predefined set of synonyms (phrases) $s_j^{(i)}$. We design $s_j^{(i)}$ according to the semantic position of $\langle h_j^{(i)} \rangle$, including nouns, verbs, adjectives, or more abstract functional tokens pertinent to the context of the instruction. As illustrated in Figure 3, consider the meta template, "*<verb> me <answer> to the question <related> the <image>*: {*question*}", where each placeholder is associated with a predefined set of positional synonyms, such as *<verb>* corresponds to three different candidates: "*give*", "*provide*", and "*offer*". When generating templates, each placeholder is randomly assigned a candidate, allowing for diverse instruction templates to be produced. For example, one possible generated template is, "*give me a response to the question concerning the provided image*: {*question*}".

2.2 Diverse Template Sampling

Sentence pattern tree. We build a sentence pattern tree to systematically organize our instruction template space and diversely sample our templates. We use $T = (V, E)$ to denote the sentence pattern tree, where V is the set of sentence patterns and E is the edge between related sentence patterns. T consists of four levels, ranging from coarse-grained to fine-grained, according to the taxonomy of sentence patterns. Level 1 represents the highest level of a sentence pattern, including declarative and imperative sentences. Level 2 decomposes Level 1 into simple, complex, and compound sentences. Level 3 further breaks Level 2 into subject-predicate, subject-predicate-object, subject-subject, noun clause, gerund clause, and linking clauses. Leaves in the final level represent the meta templates belonging to the above parent nodes. We then perform weighted sampling on Level 4 according to vertice features in Level 1 to Level 3. We construct two sentence pattern trees, one consists of 24 meta templates for visual questions and another consists of 14 meta templates for choices, yielding an extensive template space encompassing 15K visual question templates and 249K choice-related templates. We present the details of our sentence pattern trees with diverse meta templates in Appendix A.1.

Weighted sampling through sentence pattern tree. To achieve diverse sampling across the extensive template space, we implement a top-down weighted sampling approach within the sentence pattern tree. Specifically, the weight of each leaf node $\ell^{(i)}$ corresponds to the number of potential templates that can be generated by the associated meta template p_i. These weights accumulate progressively up each level of the tree, with the weight w_v of each node $v \in V$ at any level representing the sum of weights of its descendant nodes in the next level. During sampling, we select nodes in a top-down manner, with the probability of sampling each node v at a given level proportional to w_v. This process ensures that the sampling probability across all templates remains uniform, promoting diversity in generated templates while preserving the semantic consistency of each instruction template. We describe the details of our weighted sampling algorithm in Appendix A.2.

3 The Impact of Instruction Templates on MLM Performance

In this section, we leverage our programmatic instruction template generator to conduct a robust evaluation for multimodal language models (MLMs) on multiple-choice VQA tasks, which can quantitatively measure MLMs' visual reasoning and conversational abilities.

3.1 Experiment Setup

Benchmark datasets. To comprehensively evaluate the instruction robustness of MLMs across diverse tasks and domains, we conduct our evaluation using five popular benchmark datasets: BLINK Fu et al. (2024), SeedBench Li et al. (2023b), MMBench Liu et al. (2025), TaskMeAnything Zhang et al. (2024a), and MMMU Yue et al. (2024). Each data point in the above datasets contains an image or multiple images, a question, several choices, and a correct answer. We filter these datasets to retain only the single-image samples for our evaluation. Specifically, we randomly select 100 data points for each dataset according to their category distribution, then combine each data point with (a). three simple instruction templates and (b). 100 randomly generated complex instruction templates, as shown below.

- **Simple**: three most commonly used instruction templates in VQA tasks: (1) {*question*}\n{*choices*}, (2) *Question:* {*question*}\n*Choices:* {*choices*}, and (3) *Question:* {*question*}\n*Select from the following choices:* {*choices*}.
- **Complex**: generated via our programmatic template generator, sampling 100 prompts from an extensive VQA template space to capture instruction format diversity.

Populating data with simple and complex templates yields two new templated datasets with 300 and 10K samples for each original dataset.

Selected models. We evaluate the performance of eight common open-source MLMs, including LLaVA-1.5-{7B, 13B} Liu et al. (2024a), LLaVA-Next-{7B, 13B} Liu et al. (2024b), Qwen-VL and Qwen-VL-Chat Bai et al. (2023), IDEFICS2-8B Laurençon et al. (2024) and InternVL-Chat-v1.5-24B Chen et al. (2024b). We evaluate all models under the same evaluation protocol to ensure fair comparisons. Evaluating the above MLMs can give us a broad overview of open-source MLMs' robustness to instruction formats.

Evaluation protocol. We fix the choice order according to the original dataset to eliminate this confounder and focus solely on the effects of instruction templates on model performance Zheng et al. (2023). To retrieve answers from MLMs' replies, we follow Zhang et al. (2024a) and adopt a two-step approach. First, we apply a string-matching algorithm to determine if the model's output matches any of three specific option representations: (1) the option identifier, e.g., *(A)*; (2) the option content, e.g., *cat*; or (3) both the identifier and the name, e.g., *(A) cat*. If no direct match is identified, we employ a sentence-transformer Reimers (2019) to calculate the embedding similarity between the model's output and each answer option, selecting the option with the highest similarity as the predicted answer. In addition to the accuracy, we follow Sclar et al. (2023) and report the range of maximum minus minimum accuracy (Max-Min) between the highest and lowest accuracy across our generated instruction templates to quantify MLM's sensitivity to instruction format variations.

3.2 MAIN RESULTS

Model		BLINK		MMBench		SeedBench		TMA		MMMU	
		Simple	Complex	Simple	Complex	Simple	Complex	Simple	Complex	Simple	Complex
LLaVA-1.5-7B	Avg.	43.67	37.26	70.00	68.55	60.67	57.35	37.00	42.94	36.67	37.19
	Max-Min	8.00	15.00 (+7.00)	18.00	16.00 (-2.00)	5.00	18.00 (+13.00)	14.00	26.00 (+12.00)	4.00	14.00 (+10.00)
LLaVA-1.5-13B	Avg.	40.00	38.75	**72.33**	73.42	67.00	68.87	54.00	52.38	37.33	39.00
	Max-Min	7.00	16.00 (+9.00)	3.00	12.00 (+9.00)	5.00	9.00 (+4.00)	8.00	16.00 (+8.00)	6.00	16.00 (+10.00)
LLaVA-Next-7B	Avg.	**45.33**	38.92	62.67	60.43	**70.00**	65.29	50.67	44.06	33.67	31.51
	Max-Min	3.00	16.00 (+13.00)	10.00	20.00 (+10.00)	2.00	18.00 (+16.00)	16.00	17.00 (+1.00)	2.00	18.00 (+16.00)
LLaVA-Next-13B	Avg.	39.67	40.72	64.67	63.47	68.33	68.76	**54.67**	51.53	31.00	33.23
	Max-Min	1.00	15.00 (+14.00)	9.00	19.00 (+10.00)	1.00	12.00 (+11.00)	5.00	21.00 (+16.00)	2.00	21.00 (+19.00)
Qwen-VL-7B	Avg.	36.00	34.44	50.67	47.51	30.67	29.66	31.67	29.76	25.67	28.06
	Max-Min	4.00	9.00 (+5.00)	3.00	11.00 (+8.00)	10.00	17.00 (+7.00)	9.00	19.00 (+10.00)	2.00	17.00 (+15.00)
Qwen-VL-Chat-7B	Avg.	31.67	40.09	62.67	74.02	56.00	58.77	39.33	51.55	39.00	36.49
	Max-Min	4.00	21.00 (+17.00)	3.00	17.00 (+14.00)	2.00	20.00 (+18.00)	8.00	17.00 (+9.00)	10.00	16.00 (+6.00)
IDEFICS2-8B	Avg.	39.33	**45.97**	71.00	70.73	43.33	53.36	36.00	47.40	29.33	27.48
	Max-Min	4.00	17.00 (+13.00)	6.00	11.00 (+5.00)	7.00	16.00 (+9.00)	8.00	20.00 (+12.00)	3.00	14.00 (+11.00)
InternVL-Chat-1.5-24B	Avg.	43.33	43.92	67.67	**77.80**	66.33	**72.43**	53.00	**56.34**	**45.33**	**44.59**
	Max-Min	6.00	24.00 (+18.00)	7.00	29.00 (+22.00)	6.00	18.00 (+12.00)	4.00	24.00 (+20.00)	1.00	17.00 (+16.00)

Table 1: Summary of our MLM evaluation results. **Simple** represents evaluating under three commonly used instruction templates, while **Complex** denotes evaluating on 100 instruction templates randomly generated from our template generator. **Avg.** denotes the average accuracy and **Max-Min** denotes the difference between best and worst accuracy across all templates. We further mark the difference of the Max-Min between Simple and Complex beside the value of Complex. The best results are marked in **bold** and the Max-Min values on the Complex are marked with grey. **The results show that MLMs are highly sensitive to slight changes in the instruction template.**

We evaluate eight MLMs across five datasets under two instruction template settings: **Simple** and **Complex**. For each setting, we report the average accuracy and performance range (Max-Min) across all instruction templates, as illustrated in Table 1. We present the following findings.

MLMs exhibit high sensitivity to variations in instruction templates on multiple-choice VQA tasks. As demonstrated in Table 1, most MLMs display substantial performance fluctuations under both simple and complex instruction template settings. For instance, InternVL-Chat-1.5-24B exhibits a performance difference (Max-Min) of 29% on the MMBench dataset under the complex template setting, underscoring the model's pronounced sensitivity to instruction format variations. Furthermore, instruction format sensitivity remains consistently high regardless of the model scale. For example, a comparison between the 7B and 13B variants of both LLaVA-1.5 and LLaVA-Next reveal similarly substantial (Max-Min) values, indicating that increasing model scale doesn't inherently reduce the sensitivity. Even after further vision instruction tuning, MLMs retain a high degree of instruction format sensitivity. Comparing Qwen-VL-7B and its instruction-tuned counterpart, Qwen-VL-Chat-7B, we observe significant (Max-Min) values across datasets for both models. This suggests that conventional vision instruction tuning can't mitigate the instruction format sensitivity, necessitating improved vision instruction tuning.

Model comparisons may reverse depending on instruction template variations. The choices of instruction templates profoundly affect the comparative performance of MLMs, as evidenced by the variability across simple and complex instruction template settings in Table 1. For example, on the BLINK dataset, LLaVA-1.5-7B outperforms LLaVA-1.5-13B under the simple setting, while this trend reverses under the complex setting. Similarly, on the BLINK, SeedBench, and MMMU datasets, LLaVA-Next-7B achieves higher average accuracy than LLaVA-Next-13B in the simple setting, whereas LLaVA-Next-13B surpasses LLaVA-Next-7B in the complex setting. This reversal illustrates that model ranking can vary significantly based on the instruction template variations. Conclusions drawn solely from one single instruction template may lead to inaccurate comparative insights, thus underscoring the need for evaluations across diverse instruction templates to capture a comprehensive view of MLM's performance.

Evaluations on commonly used templates tend to underestimate the performance variability of models. The results reveal a consistent pattern across models, where the performance range (Max-Min) is significantly smaller under the simple template setting than under the complex template setting. For example, in the case of the InternVL-Chat-1.5-24B model on the MMBench dataset, the (Max-Min) values for the simple and complex settings are 7 and 29, respectively, demonstrating that a limited range of instruction templates fails to capture the full extent of performance fluctuation. This disparity highlights a critical limitation in conventional MLM evaluations, as they potentially overlook the influence of instruction template diversity on model robustness. Such overlooked variability renders these evaluations less reliable for real-world applications where instruction formats are inherently diverse.

4 Visual Instruction Tuning with Diverse Instruction Templates

To tackle the issues found in Section 3, the sensitivity of MLMs to subtle changes in instruction templates, we propose a simple yet effective method that improves visual instruction tuning through a data-centric approach. Our method involves applying randomly generated instruction templates from our template generator to the original QA pairs, significantly improving MLMs' performance and reducing their sensitivity to instruction template variations. We further compare the performance of the model tuned on our method against other prominent MLMs of comparable scales (Sec. 4.2). We further conduct an ablation study to investigate how the ratio between templates and the amount of training data affects the performance of our method (Sec. 4.3).

4.1 Experiment Setup

Training configurations. We trained two models based on the pretrained checkpoints: LLaVA-1.5-7B-Base and LLaVA-1.5-13B-Base, which are strong starting points for visual instruction tuning due to the open-source nature of data and models in this series. We used Low-Rank Adaptation (LoRA) Hu et al. (2021) to train all models under the same hyperparameter settings. We used a batch size of 128 and a learning rate of 2×10^{-5} with a cosine decay schedule. The learning rate warmup ratio is set to 0.03. We used the AdamW Loshchilov & Hutter (2019) optimizer and performed fine-tuning with DeepSpeed[1] at stage 3. We trained all models with $16 \times$ A100 (40G).

Our method. We used the 665K multimodal instruction-following data[2] provided by the LLaVA-1.5 series. Without introducing additional data sources or training techniques, we applied instruction templates to the instruction part of the training data, resulting in a template-diversified dataset that maintains the same scale as the original. The enhanced dataset was subsequently used to finetune our pretrained LLaVA-1.5-7B-Base and LLaVA-1.5-13B-Base models.

Baseline. To establish our baseline models, we used original instruction data to perform conventional visual instruction tuning on the LLaVA-1.5-7B-Base and LLaVA-1.5-13B-Base, yielding LLaVA-1.5-7B and LLaVA-1.5-13B, which serve as our primary baselines. In addition, for the 7B model scale, we selected LLaVA-Next-7B, Qwen-VL-7B, Qwen-VL-Chat-7B, and IDEFICS2-8B as additional baselines; for the 13B model, we selected LLaVA-Next-13B as an additional baseline model. Notably, each of these additional baseline models was finetuned on a substantially larger dataset than ours.

Evaluation. We evaluated on the BLINK, MMBench, Seedbench, TaskMeAnything, and MMMU datasets. Given the computational cost associated with evaluating across multiple instruction tem-

[1] `https://github.com/microsoft/DeepSpeed`

[2] `https://huggingface.co/datasets/liuhaotian/LLaVA-Instruct-150K/blob/main/llava_v1_5_mix665k.json`

plates, we randomly selected 100 samples from each dataset. To demonstrate the robustness of our method, we conducted evaluations under the following three instruction template settings.

(1) In-domain templates: We generated 100 templates using our template generator, which our template-tuned models have encountered during training.

(2) Out-of-domain templates: To assess the generalization ability of our method, we manually wrote 25 templates that are outside the template space of our template generator. These templates serve as a held-out set for evaluation.

(3) Commonly used simple templates: To measure the ease of use of our template-tuned model, we selected the three instruction templates from the **Simple** template set in Section 3.

4.2 Instruction Templates Can Improve Vision Instruction Tuning

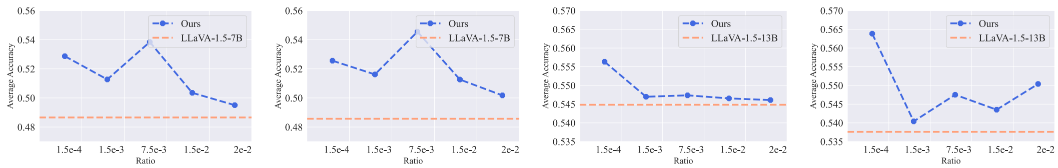
Model	# IT-Data		BLINK			MMB			SeedB			TMA			MMMU			Overall
			S	ID	OOD	S	ID	OOD	S	ID	OOD	S	ID	OOD	S	ID	OOD	
7B / 8B Models																		
LLaVA-1.5-7B	665k	Avg.	43.67	37.26	38.72	70.00	68.55	69.20	60.67	57.35	56.16	37.00	42.94	42.60	36.67	37.19	36.16	48.94
		Max-Min	8.00	15.00	15.00	18.00	16.00	9.00	5.00	18.00	16.00	14.00	26.00	18.00	4.00	14.00	13.00	13.93
LLaVA-Next-7B	760k	Avg.	45.33	38.92	37.64	62.67	60.43	58.08	**70.00**	**65.29**	62.16	50.67	44.06	44.60	33.67	31.51	29.24	48.95
		Max-Min	7.00	16.00	12.00	10.00	20.00	9.00	2.00	18.00	10.00	16.00	17.00	11.00	2.00	18.00	8.00	11.73
Qwen-VL-7B	50M	Avg.	36.00	34.44	34.04	50.07	47.51	47.16	30.67	29.66	28.80	31.67	29.76	30.76	25.67	28.06	28.40	34.18
		Max-Min	4.00	9.00	8.00	3.00	11.00	11.00	10.00	17.00	12.00	9.00	19.00	14.00	2.00	17.09	11.00	10.47
Qwen-VL-Chat-7B	50M	Avg.	31.67	40.09	40.28	62.67	**74.02**	**75.16**	56.00	58.77	58.32	39.33	51.55	51.48	39.00	36.49	36.36	50.08
		Max-Min	4.00	21.00	20.00	3.00	17.00	14.00	2.00	20.00	13.00	8.00	17.00	12.00	10.00	16.00	10.00	12.47
IDEFICS2-8B	1.8M	Avg.	39.33	**45.97**	**46.36**	**71.00**	70.73	70.28	43.33	53.36	54.04	36.00	47.40	46.20	29.33	27.48	28.36	47.28
		Max-Min	4.00	17.00	10.00	6.00	11.00	9.00	7.00	16.00	17.00	8.00	20.00	17.00	3.00	14.00	11.00	11.33
LLaVA-1.5-7B-Base w/ Ours	665k	Avg.	**46.33**	43.19	45.44	68.67	71.66	73.20	64.33	65.13	**64.16**	**52.00**	**51.78**	**52.64**	**39.33**	**37.46**	**37.32**	**54.18**
		Max-Min	5.00	13.00	2.55	10.00	12.00	8.00	3.00	11.00	6.00	4.00	22.00	10.00	9.00	11.00	6.00	8.84
13B Models																		
LLaVA-1.5-13B	665k	Avg.	**40.00**	38.75	41.20	**72.33**	73.42	71.24	67.00	68.87	66.92	54.00	**52.38**	**52.24**	37.33	39.00	37.20	54.13
		Max-Min	7.00	16.00	14.00	3.00	12.00	6.00	5.00	9.00	10.00	8.00	16.00	15.00	6.00	16.00	10.00	10.20
LLaVA-Next-13B	760k	Avg.	39.67	40.72	38.16	64.67	63.47	63.40	68.33	68.76	66.88	**54.67**	51.53	47.68	31.00	33.23	33.80	51.06
		Max-Min	1.00	15.00	13.00	9.00	19.00	15.00	1.00	12.00	11.00	5.00	21.00	14.00	2.00	21.00	10.00	11.27
LLaVA-1.5-13B-Base w/ Ours	665k	Avg.	37.67	**41.22**	**42.68**	70.00	**73.88**	**74.68**	**69.33**	**69.37**	**69.48**	51.33	50.49	50.68	**39.67**	**43.21**	**44.40**	**55.21**
		Max-Min	14.00	15.00	8.00	12.00	10.00	10.00	3.00	7.00	5.00	1.00	12.00	5.00	7.00	15.00	15.00	9.27

Table 2: Comparison of our method applied to LLaVA-1.5-7B-Base / LLaVA-1.5-13B-Base against similar-scale MLMs. **Avg.** denotes the average accuracy and **Max-Min** denotes the difference between best and worst accuracy across all templates. **#IT-Data** is the size of instruction tuning data the model used. **S** indicates the evaluation of three commonly used simple templates, **ID** refers to the evaluation of 100 instruction templates that our template-tuned model has encountered during training, and **OOD** denotes the evaluation of 25 manually crafted templates not included in our instruction template generator's template space. The best results are marked in **red bold** and the second best in blue. **Training with the template-augmented instruction data can boost performance across most benchmarks.**

As shown in Table 2, we compare our 7B and 13B models, trained with template-augmented instruction data, against several prominent MLMs of similar scale, revealing the following two key findings.

Template-augmented instruction data significantly enhances MLM's performance without increasing the scale of training data. In comparison with LLaVA-1.5-7B and LLaVA-1.5-13B, which use the same pretrained models as our tuned models but rely on original instruction data, our approach of applying diverse instruction templates to the instruction part in the training data yields marked improvements across most datasets in all three evaluation settings. Furthermore, our method demonstrates superior overall performance compared to other prominent MLMs of similar scale. Remarkably, these similar-scale models were trained on much larger datasets (at most 75.19x) than ours, highlighting the effectiveness of our method in enhancing visual instruction tuning with a more efficient use of data.

Template-augmented instruction data significantly enhances MLM's robustness to diverse instruction templates. Compared to LLaVA-1.5-7B and LLaVA-1.5-13B, our approach not only improves overall performance but also reduces the performance fluctuation range (Max-Min) across



(a) 7B Models on ID templates. (b) 7B Models on OOD templates. (c) 13B Models on ID templates. (d) 13B Models on OOD templates.

Figure 4: Scaling trend of the ratio of #instruction templates to #training data on the average performance across five benchmarks. **There exists an optimal template-to-data ratio for MLM's general capabilities, with stronger models requiring a smaller ratio.**

multiple instruction templates in most cases. When compared to other prominent MLMs of similar scale, our models trained with template-augmented instruction data exhibit a lower performance fluctuation range in most cases. This reduction in performance range remains stable across both in-domain (**ID**) and out-of-domain (**OOD**) instruction template settings, while counterexamples are more likely to arise with commonly used simple templates (**S**), given the limited use of only three evaluation templates. Notably, even when assessed using our manually written out-of-domain templates, which are outside the template space of our instruction template generator, our models frequently demonstrate a smaller performance fluctuation range. This observation underscores the effectiveness of our method in generalizing beyond the instruction templates encountered during training, rather than merely memorizing them.

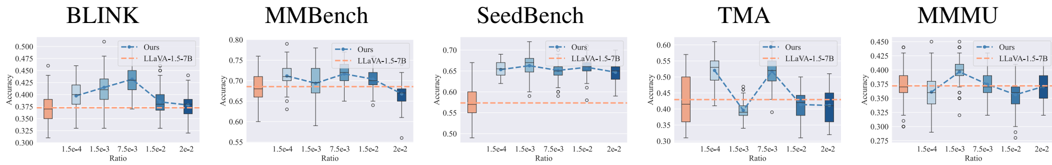
4.3 Ablation on Scaling Ratio between Training Data and Templates

To investigate the impact of the ratio of instruction templates to training data (denoted as template-to-data ratio) on model performance, we created five template-augmented versions of the original 665K dataset by applying randomly sampled 100, 1K, 5K, 10K, and 15K templates. This yielded template-to-data ratios of 1.5×10^{-4}, 1.5×10^{-3}, 7.5×10^{-3}, 1.5×10^{-2}, and 2.2×10^{-2}, while keeping the overall dataset size constant. Using these template-augmented datasets, we trained **ten** models (five with 7B parameters and five with 13B parameters) and evaluated their performance across all five benchmark datasets in both in-domain and out-of-domain template settings. Figure 4 shows the scaling curves for average performance across all datasets, while Figure 5 presents the scaling curves for each dataset. These results reveal three main findings.

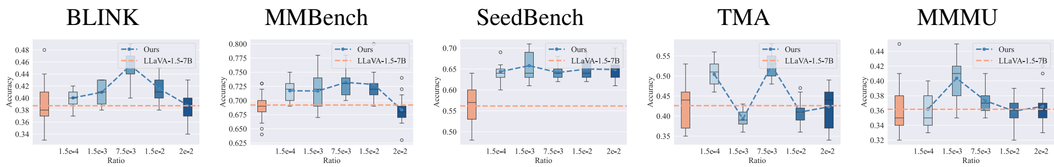
MLMs perform best at specific template-to-data ratios. As shown in Figure 4, our models, which were trained with diverse instruction templates, consistently outperform models that rely on original instruction tuning data, as reflected in the average performance across five benchmark datasets. This holds across different model scales (7B and 13B), as well as for both in-domain and out-of-domain evaluation template settings, highlighting the effectiveness of our approach. Furthermore, we observe that at the 7B scale, the model achieves peak performance when the template-to-data ratio is 7.5×10^{-3}, for both in-domain and out-of-domain evaluation template settings. At the 13B scale, however, the optimal ratio stabilizes at 1.5×10^{-4}. The consistent scaling trends suggest the existence of a specific optimal template-to-data ratio for MLM's general capabilities, with the model exhibiting stronger base capacity requiring a smaller optimal ratio.

Optimal template-to-data ratios vary across datasets. As shown in Figure 5, the scaling trend of the template-to-data ratio exhibits significant variability across different datasets, with the optimal ratio differing for each dataset. Furthermore, we observed that an inappropriate template-to-data ratio can lead to a decrease in performance or an increase in performance fluctuation range compared to the original model on certain datasets, revealing the limitations of our approach in specific scenarios.

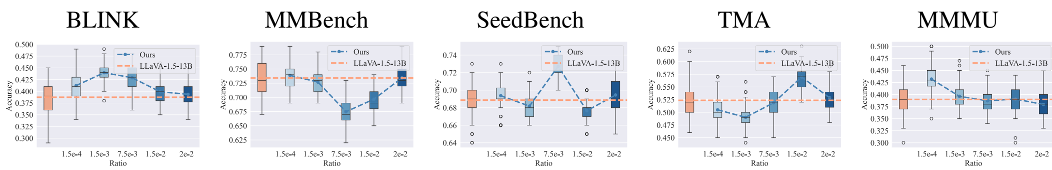
Template-to-data ratio scaling trends are broadly generalizable. Whether considering the average performance in Figure 4 or the performance on individual datasets in Figure 5, both the 7B and 13B template-tuned models exhibit consistent scaling trends across in-domain and out-of-domain evaluation template settings. This consistency demonstrates that the effect of the template-to-data ratio is generalizable and doesn't overfit the instruction templates used during finetuning.



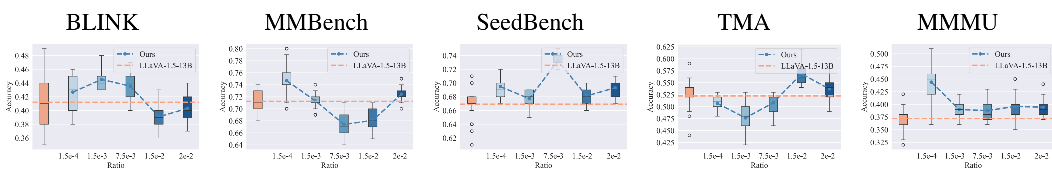
(a) Evaluation of 7B models on in-domain templates.



(b) Evaluation of 7B models on out-of-domain templates.



(c) Evaluation of 13B models on in-domain templates.



(d) Evaluation of 13B models on out-of-domain templates.

Figure 5: Scaling trends of the ratio of #instruction templates to #training data on each dataset. We also show the performance spread across models and datasets. **Optimal template-to-data ratios vary across datasets.**

5 Related Work

Multimodal language model. In recent years, multimodal language models (MLMs) have advanced visual-language learning by integrating visual encoders within various pretrained large language models Sun et al. (2023); Lyu et al. (2023); Tang et al. (2023); Wang et al. (2023); Bi et al. (2023); Chen et al. (2023a); Liu et al. (2024d); Peng et al. (2023); Chen et al. (2023b); Shukor et al. (2023); Lin et al. (2023); Lu et al. (2023); Li et al. (2023a); Sun et al. (2024b); Moor et al. (2023); Awadalla et al. (2023); Sun et al. (2024a); Xue et al. (2024a). With the increasing availability of open-sourced LLM backbones and extensive visual instruction-tuning data, models like the BLIP series Dai et al. (2024); Li et al. (2022; 2023c); Panagopoulou et al. (2023); Xue et al. (2024a), QwenVL series Bai et al. (2023); Wang et al. (2024a), LLaVA series Liu et al. (2024c; 2023; 2024b), and InternVL series Chen et al. (2023c; 2024a), have achieved unprecedented performance in a wide range of visual tasks Lin et al. (2024); Luo et al. (2024); Ma et al. (2024); Xue et al. (2024b); Wang et al. (2024b); An et al. (2024); Zhang et al. (2023). These models, which take both visual content and language as input and output language, are now considered a new type of foundation model with exceptional visual understanding capabilities. However, these MLMs largely overlooked the significance of instruction templates of prompts, resulting in unreliable, unstable evaluation results.

Influence of template perturbation. Recent research illustrated how prompt perturbations affect the performance and robustness of large language models (LLMs) and MLMs Gonen et al. (2022); Lu et al. (2021); Madaan et al. (2023); Zhuo et al. (2024); Gan & Mori (2023). Liang et al. (2022) performed a comprehensive examination of MLM outputs under diverse prompt designs, emphasizing the importance of systematic evaluation to ensure MLM robustness. Liu et al. (2024e) highlight that MLMs often produce incorrect responses when presented with nuanced, leading questions, underlining their susceptibility to prompt design variations. To solve this problem, Chatterjee et al. (2024) propose a prompt sensitivity index method that captures the relative change in log-likelihood of the given prompts, making it a more reliable measure of prompt sensitivity. Some former methods Leidinger et al. (2023); Mizrahi et al. (2024); Voronov et al. (2024) also have proposed to extend the evaluation benchmarks from a single prompt to multiple variants for each prompt. However, these former methods are all based on hand-crafted methods, which are not comprehensive enough to evaluate LLMs and MLMs. Meanwhile, most existing benchmarks, such as BLINK Fu et al. (2024), SeedBench Li et al. (2023b), MMBench Liu et al. (2025), TaskMeAnything Zhang et al. (2024a), and MMMU Yue et al. (2024), still keep using a single template of the prompts for the performance evaluation.

6 Discussion

6.1 Limitation

Designing the template space requires manual effort. The development of meta templates and the association of placeholders with semantically equivalent synonyms demand significant manual intervention. Despite the automation of template generation, ensuring semantic consistency and grammatical correctness across diverse templates is labor-intensive.

Evaluation across multiple instruction templates is cost-prohibitive. Evaluating MLMs with extensive template spaces incurs high computational costs due to the increased number of evaluations per dataset. This limits the scalability of testing, especially for large datasets or when comparing multiple models. The high costs associated with such exhaustive evaluations often necessitate trade-offs, limiting the breadth of experimentation and potentially overlooking optimal template configurations.

An imbalance in the template-to-data ratio during training can degrade model performance on specific datasets. The results in Sec. 4.3 indicate that models achieve peak performance at specific template-to-data ratios, which vary based on model scale and dataset. Disproportionate scaling of either templates or data can lead to performance variability and generalization challenges.

6.2 Future Work

Budget-constrained instruction template optimization tailored to specific models and tasks. The findings in Sec. B indicate that no universal optimal instruction template exists across all models. However, for a specific model and dataset, it is practical and valuable to identify the most effective instruction template from a large pool of predefined options within a constrained computational budget. Our future work will explore developing efficient methods for optimizing instruction templates to enhance task-specific model performance.

Enhancing the generalization of template-augmented training. The conclusions present in Sec. 4.3 highlight the limitations of our approach when faced with an imbalanced template-to-data ratio. To address this, our future research will explore developing advanced techniques to enhance the generalization capabilities of our template augmentation methods, ensuring its robustness across diverse scenarios and datasets.

7 Conclusion

We introduce a programmatic instruction template generator to efficiently produce diverse, high-quality instruction templates at scale, aimed at enhancing the understanding of the critical role instruction templates play in MLM evaluation and training. Using this instruction template generator, we conduct a comprehensive evaluation of MLMs' robustness to instruction template perturbations, demonstrating the high sensitivity of MLMs to variations in instruction templates. Additionally, we propose a simple yet effective method to improve visual instruction tuning by augmenting the origin instruction-tuning dataset with programmatically scaling instruction templates, offering an efficient and cost-effective solution to improve MLMs. Our ablation studies show that models achieve the best general capabilities at a specific ratio between instruction templates and training data, which varies with the model scale.

REFERENCES

Ruichuan An, Sihan Yang, Ming Lu, Kai Zeng, Yulin Luo, Ying Chen, Jiajun Cao, Hao Liang, Qi She, Shanghang Zhang, et al. Mc-llava: Multi-concept personalized vision-language model. *arXiv preprint arXiv:2411.11706*, 2024.

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo: An open-source framework for training large autoregressive vision-language models, 2023.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

Jing Bi, Nguyen Manh Nguyen, Ali Vosoughi, and Chenliang Xu. Misar: A multimodal instructional system with augmented reality, 2023.

Anwoy Chatterjee, HSVNS Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. Posix: A prompt sensitivity index for large language models. *arXiv preprint arXiv:2410.02185*, 2024.

Houlun Chen, Xin Wang, Hong Chen, Zihan Song, Jia Jia, and Wenwu Zhu. Grounding-prompter: Prompting llm with multimodal information for temporal sentence grounding in long videos, 2023a.

Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, Daniel Salz, Xi Xiong, Daniel Vlasic, Filip Pavetic, Keran Rong, Tianli Yu, Daniel Keysers, Xiaohua Zhai, and Radu Soricut. Pali-3 vision language models: Smaller, faster, stronger, 2023b.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023c.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24185–24198, 2024b.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. *arXiv preprint arXiv:2404.12390*, 2024.

Chengguang Gan and Tatsunori Mori. Sensitivity and robustness of large language models to prompt template in japanese text classification tasks. *arXiv preprint arXiv:2305.08714*, 2023.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*, 2024.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models?, 2024.

Alina Leidinger, Robert Van Rooij, and Ekaterina Shutova. The language of prompting: What linguistic properties make a prompt successful? *arXiv preprint arXiv:2311.01967*, 2023.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Mimic-it: Multi-modal in-context instruction tuning, 2023a.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023b.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023c.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Kevin Lin, Faisal Ahmed, Linjie Li, Chung-Ching Lin, Ehsan Azarnasab, Zhengyuan Yang, Jianfeng Wang, Lin Liang, Zicheng Liu, Yumao Lu, Ce Liu, and Lijuan Wang. Mm-vid: Advancing video understanding with gpt-4v(ision), 2023.

Weifeng Lin, Xinyu Wei, Ruichuan An, Peng Gao, Bocheng Zou, Yulin Luo, Siyuan Huang, Shanghang Zhang, and Hongsheng Li. Draw-and-understand: Leveraging visual prompts to enable mllms to comprehend what you want. *arXiv preprint arXiv:2403.20271*, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024a.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024b. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024c.

Shikun Liu, Linxi Fan, Edward Johns, Zhiding Yu, Chaowei Xiao, and Anima Anandkumar. Prismer: A vision-language model with multi-task experts, 2024d.

Yexin Liu, Zhengyang Liang, Yueze Wang, Muyang He, Jian Li, and Bo Zhao. Seeing clearly, answering incorrectly: A multimodal robustness benchmark for evaluating mllms on leading questions. *arXiv preprint arXiv:2406.10638*, 2024e.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pp. 216–233. Springer, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*, 2019.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models, 2023.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021.

Yulin Luo, Ruichuan An, Bocheng Zou, Yiming Tang, Jiaming Liu, and Shanghang Zhang. Llm as dataset analyst: Subpopulation structure discovery with large language model. *arXiv preprint arXiv:2405.02363*, 2024.

Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration, 2023.

Zixian Ma, Weikai Huang, Jieyu Zhang, Tanmay Gupta, and Ranjay Krishna. m&m's: A benchmark to evaluate tool-use for multi-step multi-modal tasks. In *Synthetic Data for Computer Vision Workshop@ CVPR 2024*, 2024.

Aman Madaan, Katherine Hermann, and Amir Yazdanbakhsh. What makes chain-of-thought prompting effective? a counterfactual study. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 1448–1535, 2023.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024.

Michael Moor, Qian Huang, Shirley Wu, Michihiro Yasunaga, Cyril Zakka, Yash Dalmia, Eduardo Pontes Reis, Pranav Rajpurkar, and Jure Leskovec. Med-flamingo: a multimodal medical few-shot learner, 2023.

Artemis Panagopoulou, Le Xue, Ning Yu, Junnan Li, Dongxu Li, Shafiq R. Joty, Ran Xu, Silvio Savarese, Caiming Xiong, and Juan Carlos Niebles. X-instructblip: A framework for aligning x-modal instruction-aware representations to llms and emergent cross-modal reasoning. *ArXiv*, abs/2311.18799, 2023. URL `https://api.semanticscholar.org/CorpusID:265506093`.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world, 2023.

N Reimers. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*, 2023.

Mustafa Shukor, Corentin Dancette, Alexandre Rame, and Matthieu Cord. Unival: Unified model for image, video, audio and language tasks, 2023.

Guangzhi Sun, Wenyi Yu, Changli Tang, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Fine-grained audio-visual joint representations for multimodal large language models, 2023.

Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners, 2024a.

Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality, 2024b.

Yunlong Tang, Jinrui Zhang, Xiangchen Wang, Teng Wang, and Feng Zheng. Llmva-gebc: Large language model with video adapter for generic event boundary captioning, 2023.

Anton Voronov, Lena Wolf, and Max Ryabinin. Mind your format: Towards consistent evaluation of in-context learning improvements. *arXiv preprint arXiv:2401.06766*, 2024.

Junke Wang, Dongdong Chen, Chong Luo, Xiyang Dai, Lu Yuan, Zuxuan Wu, and Yu-Gang Jiang. Chatvideo: A tracklet-centric multimodal and versatile video understanding system, 2023.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Weizhi Wang, Khalil Mrini, Linjie Yang, Sateesh Kumar, Yu Tian, Xifeng Yan, and Heng Wang. Finetuned multimodal language models are high-quality image-text data filters. *arXiv preprint arXiv:2403.02677*, 2024b.

Yuxuan Xie, Tianhua Li, Wenqi Shao, and Kaipeng Zhang. Tp-eval: Tap multimodal llms' potential in evaluation by customizing prompts. *arXiv preprint arXiv:2410.18071*, 2024.

Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, Shrikant B. Kendre, Jieyu Zhang, Can Qin, Shu Zhen Zhang, Chia-Chih Chen, Ning Yu, Juntao Tan, Tulika Awalgaonkar, Shelby Heinecke, Huan Wang, Yejin Choi, Ludwig Schmidt, Zeyuan Chen, Silvio Savarese, Juan Carlos Niebles, Caiming Xiong, and Ran Xu. xgen-mm (blip-3): A family of open large multimodal models. *ArXiv*, abs/2408.08872, 2024a. URL https://api.semanticscholar.org/CorpusID:271891872.

Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, et al. xgen-mm (blip-3): A family of open large multimodal models. *arXiv preprint arXiv:2408.08872*, 2024b.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024.

Jieyu Zhang, Weikai Huang, Zixian Ma, Oscar Michel, Dong He, Tanmay Gupta, Wei-Chiu Ma, Ali Farhadi, Aniruddha Kembhavi, and Ranjay Krishna. Task me anything. *arXiv preprint arXiv:2406.11775*, 2024a.

Jieyu Zhang, Le Xue, Linxin Song, Jun Wang, Weikai Huang, Manli Shu, An Yan, Zixian Ma, Juan Carlos Niebles, silvio savarese, Caiming Xiong, Zeyuan Chen, Ranjay Krishna, and Ran Xu. Provision: Programmatically scaling vision-centric instruction data for multimodal language models, 2024b. URL https://arxiv.org/abs/2412.07012.

Qizhe Zhang, Bocheng Zou, Ruichuan An, Jiaming Liu, and Shanghang Zhang. Split & merge: Unlocking the potential of visual adapters via sparse training. *arXiv preprint arXiv:2312.02923*, 2023.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. On large language models' selection bias in multi-choice questions. *arXiv preprint arXiv:2309.03882*, 2023.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. Prosa: Assessing and understanding the prompt sensitivity of llms. *arXiv preprint arXiv:2410.12405*, 2024.

A DETAILS OF INSTRUCTION TEMPLATE GENERATOR

Our instruction template generator can produce an extensive template space comprising 15K visual question templates and 249K choice-related templates, culminating in a comprehensive VQA instruction template space of 3.9B unique combinations. Our method operates by sampling meta templates from the sentence pattern tree with a weighted sampling algorithm and then programmatically populating placeholders in meta templates with randomly sampled positional synonyms. In this section, we present the details of our sentence pattern trees with diverse meta templates (Sec. A.1) and the weighted sampling algorithm (Sec. A.2).

A.1 SENTENCE PATTERN TREE WITH DIVERSE META TEMPLATES

We construct two sentence pattern trees, one consisting of 24 meta templates for visual questions and the other consisting of 14 meta templates for choices. To accommodate the distinct sentence structure preferences of visual questions and choice-related instruction templates, the taxonomy of these two sentence pattern trees differs slightly. We present the sentence pattern tree for visual questions in Figure 6a and the sentence pattern tree for choices in Figure 6b.

A.2 WEIGHED SAMPLING ALGORITHM

Algorithm 1 Weight Accumulation

procedure ACCUMULATEWEIGHTS(T)
  for each leaf node v in T **do**
    $w(v) \leftarrow \text{NumTemplates}(v)$ ▷ Set weight to number of potential generated templates in the leaf
  end for
  for each non-leaf node v in T in reverse topological order **do**
    $C \leftarrow \text{children}(v)$ ▷ Retrieve children of v
    $w(v) \leftarrow \sum_{c \in C} w(c)$ ▷ Sum the weights of child nodes
  end for
  return T ▷ Return tree with accumulated weights
end procedure

Algorithm 2 Weighted Sampling and Template Generation

procedure GENERATETEMPLATE(T)
  $v \leftarrow v_0$ ▷ Initialize at the root node of T
  while v is not a leaf node **do**
    $C \leftarrow \text{children}(v)$ ▷ Retrieve child nodes of v
    $W \leftarrow \{w(c) : c \in C\}$ ▷ Collect weights of child nodes
    $v \leftarrow \text{WeightedRandomChoice}(C, W)$ ▷ Select a child node based on weights
  end while
  $p \leftarrow \text{pattern}(v)$ ▷ Retrieve the meta template from the selected leaf node
  for each placeholder $\langle h_j \rangle$ in p **do**
    $S_j \leftarrow \text{synonyms}(\langle h_j \rangle)$ ▷ Retrieve synonyms for the placeholder
    $s_j \leftarrow \text{UniformRandomChoice}(S_j)$ ▷ Randomly select a synonym
    Replace $\langle h_j \rangle$ in p with s_j ▷ Substitute placeholder with synonym
  end for
  return p ▷ Return the constructed instruction template
end procedure

To ensure diverse meta template sampling, we propose a weighted sampling algorithm within the sentence pattern tree that guarantees a uniform probability distribution across all meta templates.

We begin by implementing an automatic weight accumulation algorithm for the sentence pattern tree. Each leaf node (meta template) is assigned a weight corresponding to the number of templates it can potentially generate. These weights are then propagated upward, with the weight of each

```
+ QuestionTemplate (weight: 15785)
    + Empty (weight: 1)
        - (weight: 1): {question}
    + Declarative (weight: 7220)
        + Simple (weight: 1092)
            + Subject-Verb-Object (weight: 640)
                - (weight: 640): The<is_following>question <related_to> the<is_provided><image> <verb> <object>:<is_line_breaking>{question}
            + Subject-LinkingVerb-Complement (weight: 452)
                - (weight: 2): Question:<is_line_breaking>{question}
                - (weight: 240): <is_the>question <related_to> the<is_provided><image><is><is_line_breaking>{question}
                - (weight: 10): <intro> is the question:<is_line_breaking>{question}
                - (weight: 200): <intro> is the question <related_to> the<is_provided><image>:<is_line_breaking>{question}
        + Compound (weight: 3780)
            + Joined-By-Coordinating-Conjunctions (weight: 2520)
                - (weight: 120): The question is <given> <below> <conjunction> you should <answer> it:<is_line_breaking>{question}
                - (weight: 2400): The question <related_to> the<is_provided><image> is <given> <below> <conjunction> you should <answer> it:<is_line_breaking>{question}
            + Joined-By-Semicolons (weight: 1260)
                - (weight: 60): The question is <given> <below>; you should <answer> it:<is_line_breaking>{question}
                - (weight: 1200): The question <related_to> the<is_provided><image> is <given> <below>; you should <answer> it:<is_line_breaking>{question}
        + Complex (weight: 2348)
            + Noun-Clauses (weight: 2220)
                - (weight: 60): The question <given> <below>  is what you should <answer>:<is_line_breaking>{question}
                - (weight: 2160): The question <given> <below> is what you should <answer> <considering><what_you_see>the<is_provided><image>:<is_line_breaking>{question}
            + Adjective-Clauses (weight: 128)
                - (weight: 128): The question <which> <adjective><is_provided><image> is<is_as_follows><is_line_breaking>{question}
    + Imperative (weight: 8564)
        + Simple (weight: 1684)
            + Subject-Predicate (weight: 592)
                - (weight: 16): <is_please><answer_directly>:<is_line_breaking>{question}
                - (weight: 576): <is_please><answer_directly> <considering><what_you_see>the<is_provided><image>:<is_line_breaking>{question}
            + Subject-Verb-Object (weight: 840)
                - (weight: 40): <is_please><answer> the<is_following>question:<is_line_breaking>{question}
                - (weight: 800): <is_please><answer> the <is_following>question <related_to> the<is_provided><image>:<is_line_breaking>{question}
            + Subject-Verb-IndirectObject-DirectObject (weight: 252)
                - (weight: 12): <verb> me <the_answer> to the question:<is_line_breaking>{question}
                - (weight: 240): <verb> me <the_answer> to the question <related_to> the<is_provided><image>:<is_line_breaking>{question}
        + Compound (weight: 2880)
            + Joined-By-Coordinating-Conjunctions (weight: 1440)
                - (weight: 1440): <is_please><verb><what_you_see>the<is_provided><image> and <answer> the<is_following>question:<is_line_breaking>{question}
            + Joined-By-Semicolons (weight: 1440)
                - (weight: 1440): <is_please><verb><what_you_see>the<is_provided><image>; <answer> the<is_following>question:<is_line_breaking>{question}
        + Complex (weight: 4000)
            + Adverbial-Clauses (weight: 3360)
                - (weight: 1920): <verb><what_you_see>the<is_provided><image>,<is_please><answer> the<is_following>question:<is_line_breaking>{question}
                - (weight: 1440): <prep><what_you_see>the<is_provided><image>,<is_please><answer> the<is_following>question:<is_line_breaking>{question}
            + Adjective-Clauses (weight: 640)
                - (weight: 640): <is_please><answer> the question <which> <adjective><is_provided><image>:<is_line_breaking>{question}
```

(a) Sentence pattern tree with meta templates for visual questions.

```
+ ChoiceTemplate (weight: 249595)
    + Empty (weight: 1)
        - (weight: 1): {choices}
    + Declarative (weight: 22776)
        + Simple (weight: 1032)
            + Subject-LinkingVerb-Complement (weight: 1032)
                - (weight: 96): <is_the><is_available><choices><are><is_line_breaking>{choices}
                - (weight: 48): <is_the><is_available><choices> are as follows:<is_line_breaking>{choices}
                - (weight: 768): <is_the><is_available><choices> are <provided><below><is_line_breaking>{choices}
                - (weight: 120): <adv> are the<is_available><choices>:<is_line_breaking>{choices}
        + Compound (weight: 15552)
            + Joined-By-Coordinating-Conjunctions (weight: 10368)
                - (weight: 10368): <is_the><is_available><choices> are <provided> <below> <conjunction> you should <verb> <object>:<is_line_breaking>{choices}
            + Joined-By-Semicolons (weight: 5184)
                - (weight: 5184): <is_the><is_available><choices> are <provided> <below>; you should <verb> <object>:<is_line_breaking>{choices}
        + Complex (weight: 6192)
            + Adjective-Clauses (weight: 6192)
                - (weight: 576): <is_the><is_available><choices> <which> include only one <correct> answer<are><is_line_breaking>{choices}
                - (weight: 288): <is_the><is_available><choices> <which> include only one <correct> answer are as follows:<is_line_breaking>{choices}
                - (weight: 4608): <is_the><is_available><choices> <which> include only one <correct> answer are <provided><below><is_line_breaking>{choices}
                - (weight: 720): <adv> are the<is_available><choices> <which> include only one <correct> answer:<is_line_breaking>{choices}
    + Imperative (weight: 226818)
        + Simple (weight: 32418)
            + Subject-Predicate (weight: 18)
                - (weight: 18): <is_please><verb> from:<choice>{choices}
            + Subject-Verb-Object (weight: 32400)
                - (weight: 32400): <is_please><verb> the<adj><answer><from><to> the question.<choice>{choices}
        + Complex (weight: 194400)
            + Adjective-Clauses (weight: 194400)
                - (weight: 194400): <is_please><verb> the<adj><answer><from><which> include only one <correct> answer <to> the question.<choice>{choices}
```

(b) Sentence pattern tree with meta templates for choices.

Figure 6: Sentence pattern trees with meta templates. Each tree uses distinct colors to denote different levels. Placeholders are marked in red, while static segments are marked in black. We further mark the weight of each node (# generated templates).

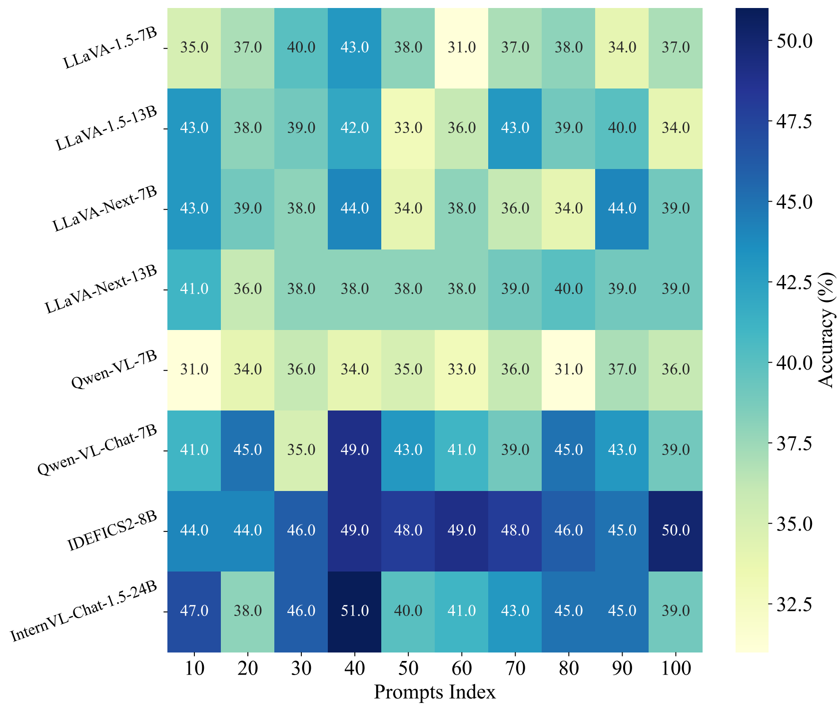


Figure 7: Heat map illustrating the performance variations of eight MLMs on the BLINK dataset across ten instruction templates (selected from the **Complex** templates set). The darker the color, the better the performance. **No single instruction template performs optimally for all MLMs.**

non-leaf node calculated as the sum of the weights of its child nodes. The detailed procedure for this algorithm is outlined in Algorithm 1.

After constructing the weighted sentence pattern tree, we perform top-down sampling, selecting nodes at each layer with probabilities proportional to their weights. Upon reaching a leaf node, we programmatically populate the placeholders in the corresponding meta template with randomly selected positional synonyms, resulting in grammatically correct and diverse instruction templates. We present the details of the procedure in Algorithm 2.

B EXPERIMENTS ON MLM'S SENSITIVITY TO INSTRUCTION TEMPLATES

In this section, we explore whether a universally effective instruction template for most MLMs exists. To this end, we analyze the performance of eight prominent MLMs on the BLINK dataset using ten instruction templates selected from the **Complex** templates set as described in Section 3. A heat map is presented in Figure 7 to illustrate the performance variations, where darker shades correspond to superior performance.

The results reveal substantial performance variability across MLMs for the same instruction template, as indicated by the diverse color gradients within each column of the heat map. This variability highlights a critical observation: **no single instruction template consistently performs optimally for all MLMs**. This lack of universality implies that each model exhibits distinct sensitivities and preferences toward different instruction template, which complicates the task of designing or selecting a universally effective template.

The observed variations have important implications for the instruction template design and evaluation of MLMs. Specifically, they underscore the limitations of a one-size-fits-all approach to instruction optimization. Efforts to identify an ideal instruction template through a single, static search are unlikely to yield universally effective results. Instead, tailored strategies that consider the specific characteristics of individual MLMs can be necessary to achieve optimal performance.