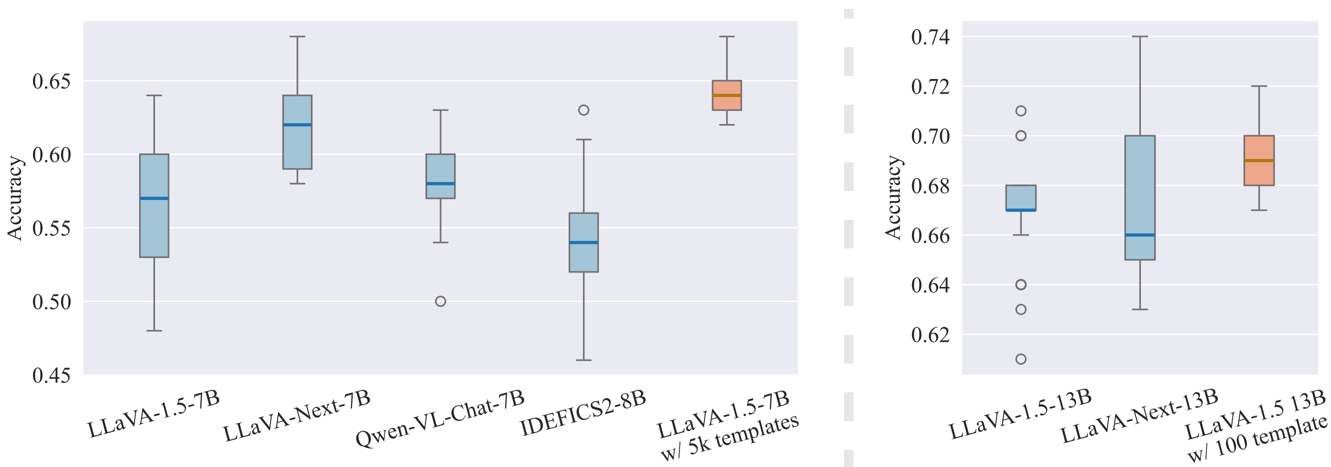
Investigating the Scaling Effect of Instruction Templates for Training Multimodal Language Model

Anonymous CVPR submission

Paper ID *****



(a) 7B / 8B models

(b) 13B models

Figure 1. **Training with the optimal template scale significantly improves MLM's performance and reduces the performance variance.** LLaVA-1.5-7B trained with 5K templates and LLaVA-1.5-13B trained with 100 templates achieve the highest average performance and the lowest performance variance among similar-scale MLMs on the SeedBench [19] dataset, evaluated across 25 held-out instruction templates that are not included in the visual instruction tuning.

Abstract

Current multimodal language model (MLM) training approaches overlook the influence of instruction templates. Previous research deals with this problem by leveraging hand-crafted or model-generated instruction templates, failing to investigate the scaling effect of instruction templates on MLM training. In this work, we propose a programmatic instruction template generator capable of producing over 15K unique instruction templates by filling randomly sampled positional synonyms into weighted sampled meta templates, enabling us to comprehensively explore MLM's performance across various template scales in the training process. Our investigation into scaling instruction templates for MLM training demonstrates that MLM capabilities do not consistently improve with increasing template scale. Instead, optimal performance is achieved at a medium template scale. Models trained with data augmented at the optimal template scale achieve performance gains of up to 10% over those trained on the original data and achieve the best overall performance compared with the similar-scale MLMs tuned on at most 75 times the scale of our augmented dataset.

1. Introduction

Multimodal Language Models (MLMs) have revolutionized vision-language learning by performing visual instruction tuning on diverse, high-quality multimodal instruction data [21, 30, 62, 65]. However, previous studies [32, 48, 61] reveal a critical limitation: MLMs exhibit substantial performance variability across different instruction templates (as shown in Figure 2). For instance, a succinct instruction and a detailed instruction can yield performance gaps exceeding 40% [61]. This pronounced sensitivity to instruction templates compromises the reliability of MLM evaluation and diminishes the practical utility of MLMs in downstream applications.

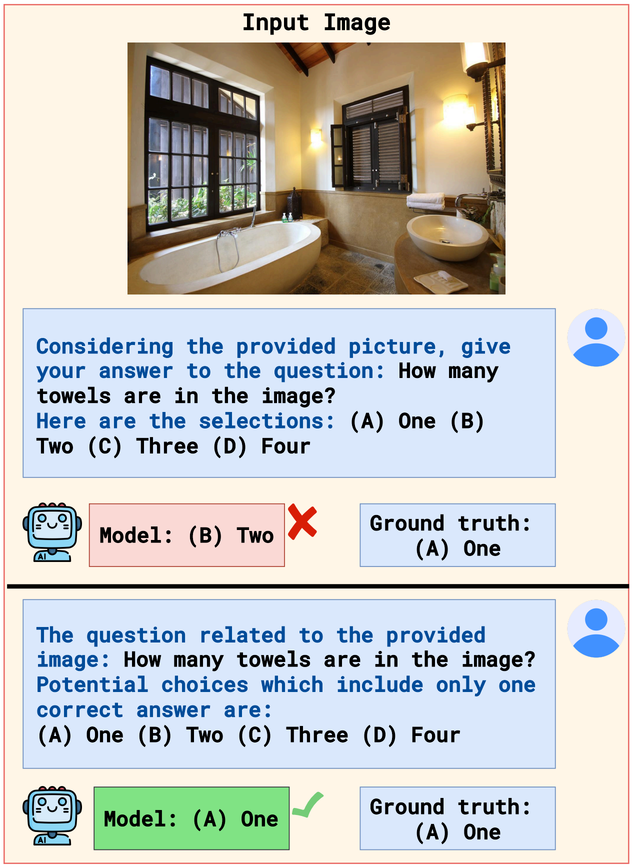


Figure 2. An example of using different instruction templates to prompt MLM without changing the original QA pairs. The instruction templates are marked in **blue**. **Prompting MLM with different instruction templates can twist the output of MLM**.

Recent studies have empirically demonstrated that incorporating multiple instruction templates during MLM's training process improves model performance and reduces instruction sensitivity [45, 47, 62]. However, existing approaches primarily depend on either human-designed or model-generated small-scale templates, which suffer from limitations such as high costs, inherent design biases, and limited diversity in instruction formulations. Considering the success of scaling up training data significantly improves model's performance [11, 16] and the fact that multi-template training can improve MLM, this raises a critical question: *How many instruction templates should be used during training to optimize MLM performance*?

To investigate the scaling effect of instruction templates for MLM training, we propose a ***programmatic instruction template generator*** that leverages diverse meta templates to produce semantically equivalent instruction templates automatically and scalably. Our template generator 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 instruction templates. To ensure the diversity of sampled instruction templates from our template generator, we use a sentence-pattern tree organizational framework based on grammatical structures complemented by an efficient diverse sampling algorithm. This programmatic approach ensures the generation of instruction templates that maximize diversity across multiple dimensions, including grammatical construction, lexical choice, and symbolic representation.

Leveraging our programmatic instruction template generator, we finetune two widely-used MLMs (LLaVA-1.5-7B and LLaVA-1.5-13B) [28] and conduct a series of experiments by performing visual instruction tuning on the same dataset while varying the scale of instruction templates (from 10 to 15K). Our study reveals that the performance of MLMs does not consistently improve with the increasing scale of instruction templates. Instead, MLMs achieve the best general capabilities at a medium template scale, which varies with the model's parameter size. We find LLaVA-1.5-7B's performance peaks at 5K templates and LLaVA-1.5-13B peaks at 100 templates. We further compare our models trained under the optimal template scale with other MLMs fine-tuned on a significantly larger scale—up to 75.19 times the size of our instruction tuning datasets. Evaluation across five benchmarks reveals that our tuned models achieve the best overall performance (We showcase the comparison results on the SeedBench [19] dataset in Figure 1), thereby demonstrating the capacity of training with appropriate template scale to enhance MLMs in a data-efficient and cost-effective manner. Additionally, our analysis reveals that, compared to the original model, fine-tuning with the optimal template scale results in a substantial reduction in performance variance across various out-of-domain instruction templates. Our approach not only confirms the practical utility of the scaling effect of instruction templates but also provides promising insights into efficient strategies for improving MLMs. 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.
- We comprehensively investigate the scaling effect of instruction templates for MLM training, demonstrating that MLM capabilities do not monotonically improve with increasing template scale and instead peak at a medium template scale.
- We propose a simple yet effective approach to enhance visual instruction tuning by augmenting the original instruction tuning dataset with the optimal scale of templates we investigated. Our extensive experiments

demonstrate its effectiveness.

2. Programmatically Scaling Instruction Templates

To investigate the scaling effect of instruction templates in MLM's visual instruction tuning, we propose a programmatic instruction template generator. Our template generator can efficiently produce diverse, grammatically correct, and semantically consistent instruction templates. Specifically, we generate instruction templates by programmatically filling the pre-defined placeholders in a *meta template* with randomly sampled positional synonyms (phrases), ensuring flexibility and diversity while keeping the original meaning (Sec. 2.1). We organize our meta templates in a *sentence pattern tree*, along with a diverse template sampling algorithm to ensure the sampling probability across all instruction templates is uniformly distributed (Sec. 2.2).

2.1. Meta Templates

We design meta template $p_i, i \in \{1, ..., N\}$ 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. The potential template variations $\mathcal{T}(p_i)$ grow combinatorially as $\mathcal{T}(p_i) = \prod_{j=1}^{M_i} |s_j^{(i)}|$, where $|s_j^{(i)}|$ is the size of each synonym set. 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*}" Fixed strings establish the foundational sentence structure, ensuring grammatical correctness and semantic coherence, while placeholders introduce flexibility and diversity, enabling the rapid generation of varied, high-quality instruction templates. To ensure the diversity of generated visual instruction templates, we design 24 meta templates, yielding a template space capable of producing 15K distinct instruction templates.

2.2. Diverse Template Sampling

Sentence pattern tree. We build a sentence pattern tree to systematically organize our meta templates. We use

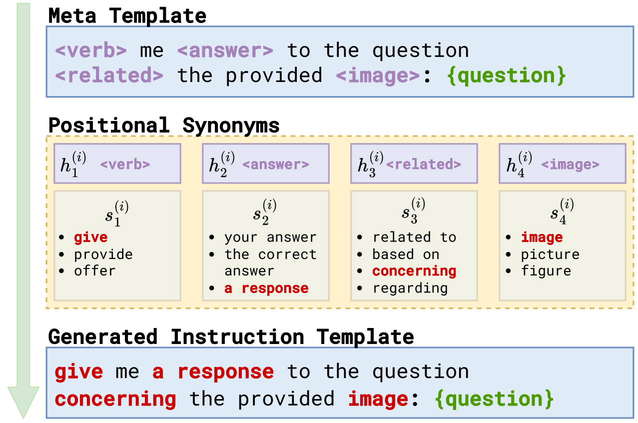


Figure 3. Example of the instruction template generation through a meta template.

$T = (V, E)$ 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. We use level 1 to represent 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 4 represent the meta templates belonging to the above parent nodes. Building on the sentence pattern tree framework, we can perform weighted sampling on Level 4 according to vertex features from Level 1 to Level 3.

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, our approach begins by assigning a weight to each tree node. 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. The weight w_v of each node $v \in V$ at any level represents the sum of weights of its descendant nodes in the next level. The detailed procedure for weight accumulation is outlined in Algorithm 1. During the template sampling process, we select nodes in a top-down manner, with the probability of sampling each node v at a given level proportional to w_v. Upon reaching a leaf node corresponding to a meta template, we programmatically fill the placeholders in the meta template with randomly selected positional synonyms. This process ensures that the sampling probability across all instruction templates remains uniform, promoting diversity in generated templates while preserving the semantic consistency

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

of each instruction template. We describe details of the weighted sampling algorithm in Algorithm 2.

3. Investigating Scaling Instruction Templates on MLM Training

To investigate the scaling effect of instruction templates in MLM's visual instruction tuning, we train multiple model variants using the same instruction tuning dataset while varying the scale of instruction templates. We then evaluate these template-tuned models across various benchmark datasets to observe the impact of the instruction template scale on MLM performance. We first present our experimental setup (Sec 3.1), followed by the experimental results and analysis (Sec 3.2).

3.1. Experiment Setup

Training configurations. We trained our template-tuned models based on the two 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) [15] 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 [34] optimizer and performed fine-tuning with DeepSpeed[1] at stage 3. We trained all models with 16 $\times$ A100 (40G).

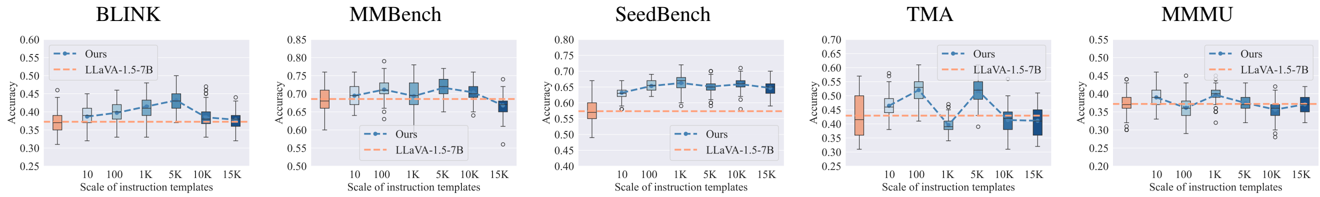
Scaling instruction templates in training data. We constructed six template-augmented versions of the original 665K-scale multimodal instruction-following data[2] (provided by the LLaVA-1.5 series) by applying randomly sampled 10, 100, 1K, 5K, 10K, and 15K templates from our programmatic template generator. Without introducing additional data sources, we applied instruction templates to the instruction part of the training data, resulting in template-diversified training datasets that maintain the same size as the original. The enhanced datasets were subsequently used to finetune the pretrained LLaVA-1.5-7B-Base and LLaVA-1.5-13B-Base models. We trained a total of **twelve** models, comprising six models with 7B parameters and six models with 13B parameters.

Benchmark datasets. To comprehensively examine the performance of our template-tuned models trained with different template scales across diverse tasks and domains, we conduct the evaluation using five popular Visual Question Answering (VQA) benchmark datasets: BLINK [12], SeedBench [19], MMBench [33], TaskMeAnything [61], and MMMU [60]. Each data point in the above benchmark 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 instruction templates to test. To evaluate the robustness of these template-tuned models, we conducted evaluations under the following two evaluation template settings.

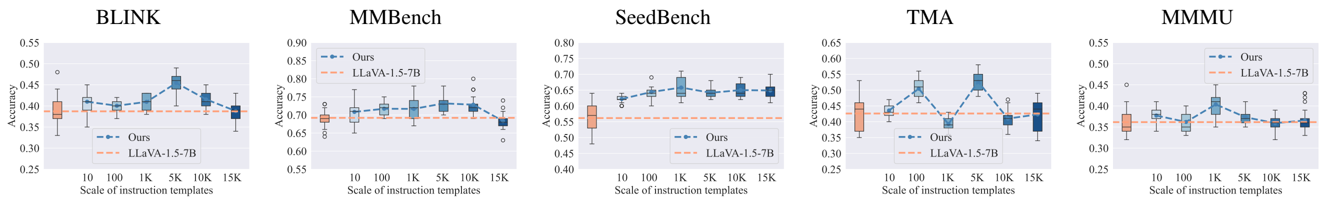
(1) In-domain templates: We generated 100 templates using our template generator, which our template-tuned models have encountered during training.

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

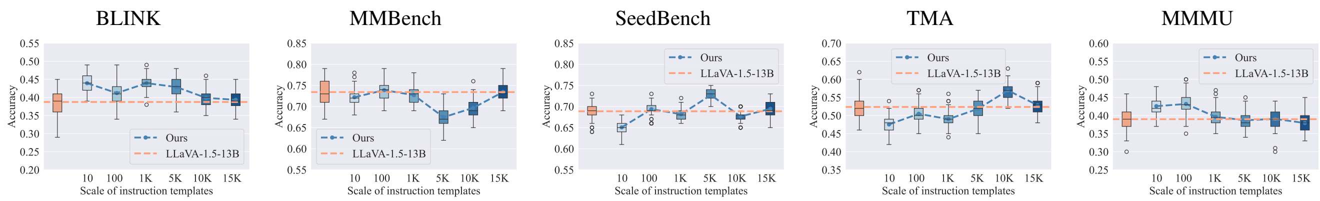
[2] https://huggingface.co/datasets/liuhaotian/LLaVA-Instruct-150K/blob/main/llava_v1_5_mix665K.json



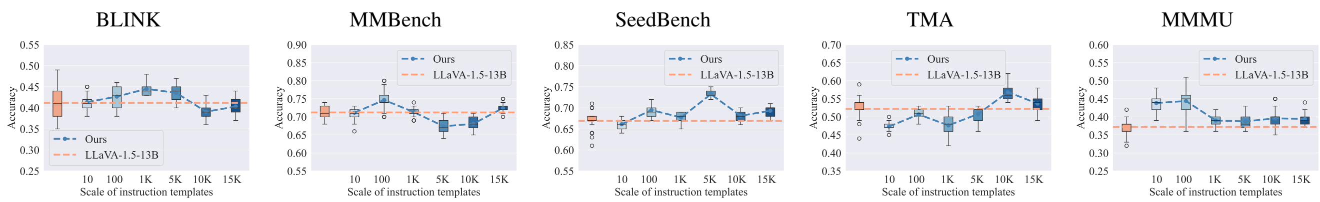
(a) Evaluation of 7B models on in-domain templates.



(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 4. Scaling trends of MLM performance with increasing template scale on each benchmark dataset. We also show the performance spread across models and datasets. **Optimal template scale vary across different datasets.**

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

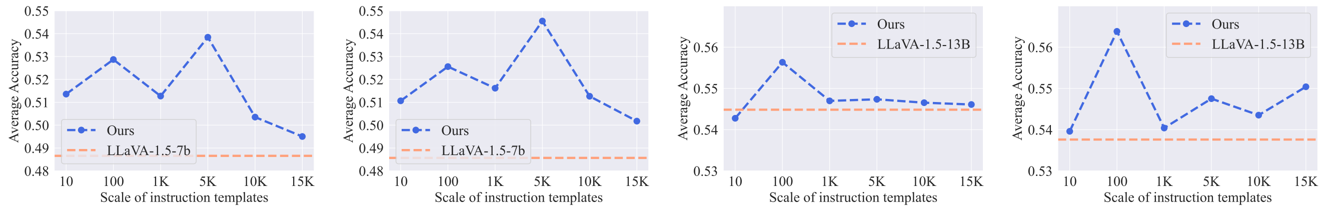
Populating evaluation data with the two template sets yields two new templated benchmark datasets with 10K and 2.5K samples for each original dataset.

Evaluation Protocol. We fix the choice order according to the original dataset to eliminate this confounder and focus solely on the effects of template scale on model performance [64]. To retrieve answers from MLMs' replies, we follow [61] 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 [46] 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. We adopt the answer accuracy on each dataset as our evaluation metric.

3.2. Comparing MLMs on Different Template Scales

Figure 4 provides detailed scaling curves of MLM performance with increasing template scale on each individual benchmark, while Figure 5 illustrates the scaling curves of the average performance across all datasets with increasing template scale. These results reveal three main findings.

Training with diverse templates can improve MLMs. As illustrated in Figure 4 and Figure 5, models trained with a diverse range of instruction templates, spanning from 10 to 15K templates, consistently outperform those trained exclusively on the original instruction tuning data. This improvement is clearly observable in both the average performance and individual performance across all five bench-



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

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

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

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

Figure 5. Scaling trend of MLM performance with increasing template scale on the average performance across five benchmarks. **There exists an optimal template scale for MLM's general capabilities, with stronger models requiring a smaller template scale.**

mark datasets. Furthermore, this trend holds true for models with varying parameter sizes (7B and 13B) and remains consistent for both in-domain and out-of-domain evaluation template settings. These results highlight the value of our exploration into the scaling effects of instruction templates in MLM training, showing that incorporating a broader set of instruction templates can lead to more robust and generalized model performance.

Optimal template scale vary across datasets. As shown in Figure 4, the scaling trend of MLM performance with increasing template scale exhibits significant variability across different datasets, with the optimal template scale differing for each dataset. Furthermore, we observed that an inappropriate template scale can lead to a decrease in performance or an increase in performance fluctuation range compared to the original model on certain datasets, highlighting the significance of finding the optimal template scale to improve model performance.

MLM capability presents clear scaling trend with increasing template scale. As illustrated in Figure 5, the model's average performance across all five datasets exhibits a consistent scaling trend, initially increasing before declining, with peak performance achieved at a medium template scale. This trend holds across different model sizes (7B and 13B parameters) and evaluation settings (in-domain and out-of-domain templates). However, the optimal template scale varies depending on model capacity: the 7B model reaches peak performance at 5K templates, whereas the 13B model achieves its best results at a significantly smaller scale of 100 templates. This discrepancy suggests that models with stronger baseline capabilities (e.g., the 13B model) require fewer templates to attain optimal performance. Furthermore, while Figure 4 demonstrates that model performance exhibits dataset-specific variability at smaller template scales, the performance consistently declines as the template scale increases beyond a certain threshold, demonstrating that the optimal template scale lies within a medium range, eliminating the need for exhaustive large-scale searches.

4. Visual Instruction Tuning on the Optimal Template Scale

To demonstrate the practical impact of the scaling effect of instruction templates for MLM's visual instruction tuning, we compare the performance of our template-tuned models trained on the optimal template scale against other prominent MLMs of similar parameter sizes. We first outline the experimental setup (Sec. 4.1), then detail the comparison results and analysis (Sec. 4.2).

4.1. Experiment Setup

Our method. We selected our best-performing template-tuned models—LLaVA-1.5-7B trained with 5K templates, and LLaVA-1.5-13B trained with 100 templates—to compare against other prominent MLMs of comparable scales.
Baselines. To establish our baseline models, we used original visual instruction data to perform conventional visual instruction tuning on the LLaVA-1.5-7B-Base and LLaVA-1.5-13B-Base models, yielding LLaVA-1.5-7B and LLaVA-1.5-13B [28], which serve as our primary baseline models. In addition, for the 7B parameter size, we selected LLaVA-Next-7B [29], Qwen-VL-7B and Qwen-VL-Chat-7B [3], and IDEFICS2-8B [17] as additional baseline models; for the 13B parameter size, we selected LLaVA-Next-13B [29] as an additional baseline model. Notably, as shown in Table 1, each of these additional baseline models was finetuned on a substantially larger training dataset than ours. We evaluate all models under the same evaluation protocol to ensure fair comparisons.
Benchmark datasets. We evaluated on the BLINK, MMBench, Seedbench, TaskMeAnything, and MMMU datasets. For consistency, we employed both *in-domain templates* and *out-of-domain templates* in Sec. 3.1 as evaluation templates. To further measure the ease of use of the template-tuned models, we selected three most commonly-used *simple templates* in VQA tasks: (1) *{question}\n{choices}*, (2) *Question: {question}\nChoices: {choices}*, and (3) *Question: {question}\nSelect from the following choices: {choices}*.

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 w/ 5K templates	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 w/ 100 templates	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 1. Comparison of our tuned models trained under the optimal template scale 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 optimal template scale can boost performance across most benchmarks.**

Evaluation Protocol. In this section, our evaluation settings are consistent with those in Sec. 3.1. For the evaluation metric, in addition to the answer accuracy, we follow [48] and report the range (Max-Min) between the best and worst accuracy across all evaluation instruction templates to quantify MLM's performance fluctuation to instruction template variations.

4.2. Main Results

As presented in Table 1, we compare the performance of our tuned 7B and 13B models, which we trained with the optimal template scale, against several prominent MLMs of similar scale, revealing the following two key findings.

Training on the optimal template scale significantly enhances MLM's performance without increasing the scale of training data. Compared to LLaVA-1.5-7B and LLaVA-1.5-13B, which utilize the same pretrained models as our template-tuned models but rely on original instruction tuning data, training with the optimal template scale achieves substantial performance improvements across most datasets in all three evaluation settings. Additionally, our tuned models trained with the optimal template scale outperforms other prominent MLMs of similar scale, despite these models being trained on significantly larger datasets (up to 75.19 times larger). This underscores the efficiency and effectiveness of our approach of training MLMs with the optimal template scale to achieve superior performance without the need for extensive data scaling. By focusing on the quality and diversity of instruction templates rather than the quantity of training data, our method demonstrates a more resource-efficient pathway to enhancing visual instruction tuning.

Training on the optimal template scale significantly mitigates MLM's sensitivity to diverse instruction templates. Compared to LLaVA-1.5-7B and LLaVA-1.5-13B, which rely on original instruction tuning data, our approach of training MLMs under the optimal template scale not only achieves superior overall performance but also significantly reduces the performance fluctuation range (Max-Min) across multiple evaluation instruction templates in most cases. This reduction in fluctuation range indicates that training on the optimal template scale enhances model stability and adaptability when faced with varying instruction formats, a critical requirement for real-world applications where input instructions can vary widely. Furthermore, when compared to other prominent MLMs of similar scale, our tuned models trained with the optimal template scale consistently exhibit a lower performance fluctuation range. This consistency holds true across both in-domain (**ID**) and out-of-domain (**OOD**) instruction template settings, demonstrating the robustness of our approach across diverse evaluation scenarios. However, counterexamples are more likely to arise with commonly used simple templates (**S**), likely due to the limited diversity of only three evaluation templates. Notably, even when evaluated using manually crafted out-of-domain templates—which lie entirely outside the template space of our instruction template generator, our template-tuned models frequently demonstrate a smaller performance fluctuation range. This observation underscores the ability of training on the optimal template scale to generalize beyond the specific instruction templates encountered during training, rather than merely memorizing them.

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 [2, 4, 6, 7, 20, 25, 31, 35, 38, 42, 44, 49–53, 55, 58]. With the increasing availability of open-sourced LLM backbones and extensive visual instruction-tuning data, models like the BLIP series [10, 22, 23, 43, 58], QwenVL series [3, 56], LLaVA series [27, 29, 30], and InternVL series [8, 9], have achieved unprecedented performance in a wide range of visual tasks [1, 26, 37, 39, 57, 59, 63]. 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 [13, 14, 36, 40, 66]. Liang et al. [24] performed a comprehensive examination of MLM outputs under diverse prompt designs, emphasizing the importance of systematic evaluation to ensure MLM robustness. Liu et al. [32] 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. [5] 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 [18, 41, 54] 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 [12], SeedBench [19], MMBench [33], TaskMeAnything [61], and MMMU [60], 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 synonyms demand minimal manual intervention. Despite the automation of template generation, ensuring semantic consistency and grammatical correctness across diverse templates demands human checking.

An inappropriate template scale during training can degrade model performance on specific datasets. The results in Sec. 3 indicate that models achieve peak performance at a medium template scale, which varies based on model scale. Disproportionate scaling templates can lead to performance variability and generalization challenges.

6.2. Future Work

Budget-constrained instruction template optimization tailored to specific models and tasks. 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. 3 highlight the limitations of our approach when faced with an inappropriate template scale. 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 benchmark datasets.

7. Conclusion

We introduced a programmatic instruction template generator to efficiently produce diverse, high-quality instruction templates at scale, aimed at investigating the scaling effect of instruction templates for MLM's visual instruction tuning. Our investigation into scaling instruction templates for MLM training showed that MLM capabilities did not monotonically improve with increasing template scale and instead peaked at a medium template scale, which varies with the model's parameter size. Additionally, using this instruction template generator, we proposed a simple yet effective method to improve visual instruction tuning by augmenting the original instruction tuning dataset at the original template scale, offering an efficient and cost-effective solution to improve MLMs.

References

[1] 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.

[2] 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.

[3] 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.

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

[5] 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.

[6] 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, 2023.

[7] 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, 2023.

[8] 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*, 2023.

[9] 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*, 2024.

[10] 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.

[11] Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7382–7392, 2024.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

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

[18] 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.

[19] 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*, 2023.

[20] 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, 2023.

[21] 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.

[22] 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.

[23] 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*, 2023.

[24] 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.

[25] 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.

[26] 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.

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

[28] 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*, pages 26296–26306, 2024.

[29] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024.

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

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

[32] 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*, 2024.

[33] 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*, pages 216–233. Springer, 2025.

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

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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*, pages 1448–1535, 2023.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] Amirhossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. Benchmarking prompt sensitivity in large language models. *arXiv preprint arXiv:2502.06065*, 2025.

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

[47] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

[48] 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.

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

[50] 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.

[51] 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, 2024.

[52] 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, 2024.

[53] 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.

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

[55] 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.

[56] 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*, 2024.

[57] 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*, 2024.

[58] 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, 2024.

[59] 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*, 2024.

[60] 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*, pages 9556–9567, 2024.

[61] 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*, 2024.

[62] Jieyu Zhang, Le Xue, Linxin Song, Jun Wang, Weikai Huang, Manli Shu, An Yan, Zixian Ma, Juan Carlos Niebles, Silvio Savarese, et al. Provision: Programmatically scaling vision-centric instruction data for multimodal language models. *arXiv preprint arXiv:2412.07012*, 2024.

[63] 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.

[64] 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.

[65] 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.

[66] 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.