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

Anonymous ACL submission

Abstract

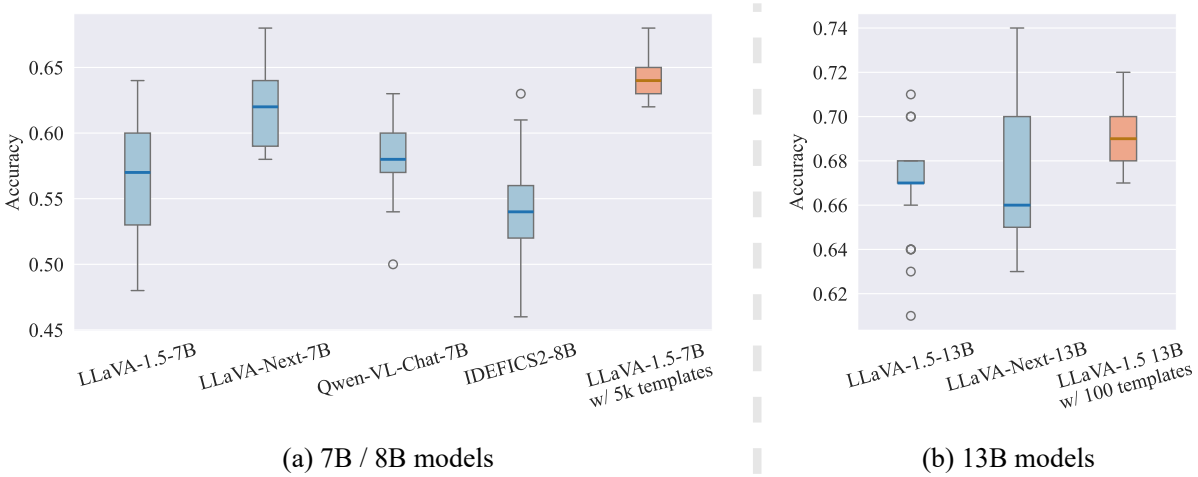
Current multimodal language model (MLM) training approaches overlook the influence of instruction templates, leading to prompt sensitivity and suboptimal performance when confronted with diverse instruction formats in real-world scenarios. Previous research deals with this problem by empirically leveraging small-scale hand-crafted or model-generated templates at training time, limiting investigation into the scaling effect of instruction templates on MLM training. In this paper, we propose a programmatic instruction template generator capable of rapidly 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 consistently achieve the best overall performance compared with the similar-scale MLMs tuned on at most 75 times the scale of our dataset. Furthermore, our results reveal that training with the optimal template scale significantly mitigates MLM's sensitivity to various instruction templates.

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; Li et al., 2024; Zhang et al., 2024b). However, previous studies (Zhang et al., 2024a; Liu et al., 2024e; Sclar et al., 2023) reveal a critical limitation: MLMs exhibit substantial performance variability across different instruction templates. For instance, a succinct instruction and a detailed instruction can yield performance gaps exceeding 40% (Zhang et al., 2024a). This pronounced sensitivity to instruction templates compromises the reliability of MLM evaluation and diminishes the practical utility of MLMs in applications.

Recent studies have empirically demonstrated that incorporating multiple instruction templates during MLM's training process can improve model performance (Zhang et al., 2024b; Razavi et al., 2025; Sanh et al., 2021). 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 (Fan et al., 2024; Kaplan et al., 2020) and the fact that multi-template training can improve MLM (Sanh et al., 2021), this raises a critical question: *How many instruction templates should be used during training to optimize MLM performance*?

To systematically 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 templates from our template generator, we use a sentence-pattern tree organizational framework based on grammatical structures complemented by an efficient diverse



(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 (Li et al., 2023b) dataset, evaluated across 25 held-out instruction templates that are not included in the visual instruction tuning process.

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 template generator, we finetune two widely-used MLMs (LLaVA-1.5-7B and LLaVA-1.5-13B) (Liu et al., 2024a) 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 training datasets. Evaluation across five benchmarks reveals that our tuned models achieve the best overall performance (We showcase the comparison results on the SeedBench (Li et al., 2023b) dataset in Figure 1), thereby demonstrating the capacity of training with appropriate template scale to enhance MLM in a data-efficient 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 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.

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

(2) We comprehensively investigate the scaling effect of instruction templates for training MLMs, demonstrating that MLM capabilities peak at medium template scale rather than improving monotonically.

(3) We propose a simple yet effective approach to enhance visual instruction tuning by augmenting the original dataset with appropriate scale of templates. Our extensive experiments demonstrate its effectiveness.

2 Programmatic Template Scaling

To investigate the scaling effect of instruction templates for MLM training, 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),

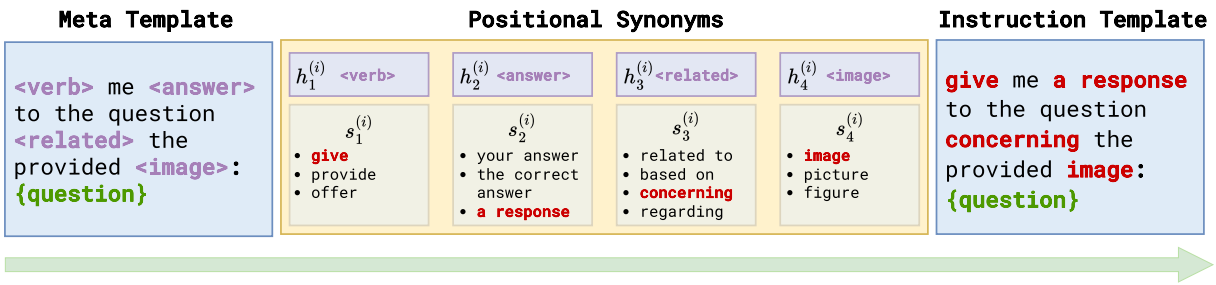


Figure 2: Example of the instruction template generation through a meta template.

Algorithm 1 Weight Accumulation

procedure ACCUMULATEWEIGHTS(T)
  for each leaf node v in T **do**
    $w(v) \leftarrow \text{NumTemplates}(v)$
  end for
  for each non-leaf node v in T in reverse topological order **do**
    $C \leftarrow \text{children}(v)$
    $w(v) \leftarrow \sum_{c \in C} w(c)$
  end for
  return T
end procedure

Algorithm 2 Weighted Template Sampling

procedure GENERATETEMPLATE(T)
  $v \leftarrow v_0$
  while v is not a leaf node **do**
    $C \leftarrow \text{children}(v)$
    $W \leftarrow \{w(c) : c \in C\}$
    $v \leftarrow \text{WeightedRandomChoice}(C, W)$
  end while
  $p \leftarrow \text{pattern}(v)$
  for each placeholder $\langle h_j \rangle$ in p **do**
    $S_j \leftarrow \text{synonyms}(\langle h_j \rangle)$
    $s_j \leftarrow \text{UniformRandomChoice}(S_j)$
    Replace $\langle h_j \rangle$ in p with s_j
  end for
  return p
end procedure

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 Template

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. 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 templates.

We present the example of instruction template generation through a meta template in Figure 2. 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 $T = (V, E)$ 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 rep-

resent the meta templates belonging to the above parent nodes. Building on the tree framework, we can perform weighted sampling on Level 4 according to vertex features from Level 1 to Level 3. We present the details of our sentence pattern trees with diverse meta templates in Appendix A.

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 node in the tree. 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 templates remains uniform, promoting diversity in generated templates while preserving the semantic consistency of each instruction template. We describe details of the weighted sampling process in Algorithm 2. We further quantitatively validate the diversity of templates generated from our template generator in Appendix B.

3 Investigating Template Scaling

To investigate the scaling effect of instruction templates for MLM training, we train multiple model variants using the same training dataset while varying the template scale. 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

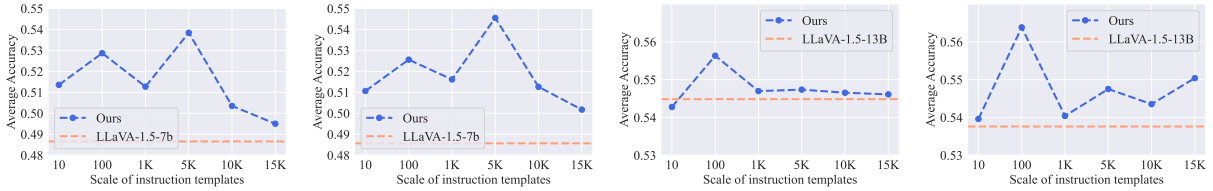
Training configurations. We trained template-tuned models based on the pretrained LLaVA-1.5-7B-Base and LLaVA-1.5-13B-Base backbones, which serve as strong starting points for visual instruction tuning due to their open-source nature and fully accessible training data. Notably, we focus on the LLaVA family as most existing open-source MLMs do not completely release their training datasets, which restricts their applicability for our template-tuning exploration. All models were fine-tuned using Low-Rank Adaptation (LoRA) (Hu et al., 2021) under consistent hyperparameter settings, including a batch size of 128, learning rate of 2×10^{-5} with cosine decay schedule, and a warmup ratio of 0.03. We employed the AdamW (Loshchilov and Hutter, 2019) optimizer and conducted training with DeepSpeed at stage 3 optimization across 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 (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 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 (Fu et al., 2024), Seed-Bench (Li et al., 2023b), MMBench (Liu et al., 2025), TaskMeAnything (Zhang et al., 2024a), and MMMU (Yue et al., 2024). Each sample in the above datasets contains an image, a question, several choices, and a correct answer. Specifically, we randomly select 100 samples 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 generator, which our template-tuned models have encountered during training.

(2) *Out-of-domain templates*: To assess the general-



(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 3: Scaling trend of model performance with increasing training template scale on the average accuracy across all five benchmarks (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)). **Optimal performance is achieved at a medium template scale, with stronger models requiring a smaller template scale.**

ization ability of these template-tuned models, we manually wrote 25 templates that are outside the space of our template generator. These templates serve as a held-out set for evaluation.

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

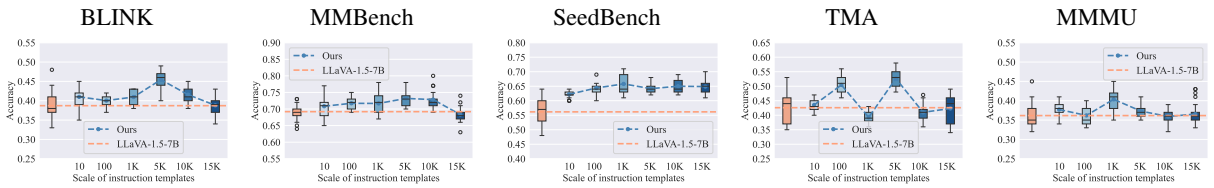
Evaluation Protocol. We fix the choice order according to the original dataset to eliminate this confounder and focus solely on the effects of template scale on model performance (Zheng et al., 2023). To retrieve answers from MLMs' replies, we follow (Zhang et al., 2024a) and adopt a two-step approach. 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. We adopt the answer accuracy on each dataset as our evaluation metric.

3.2 Comparing MLM on Various Scales of Instruction Templates

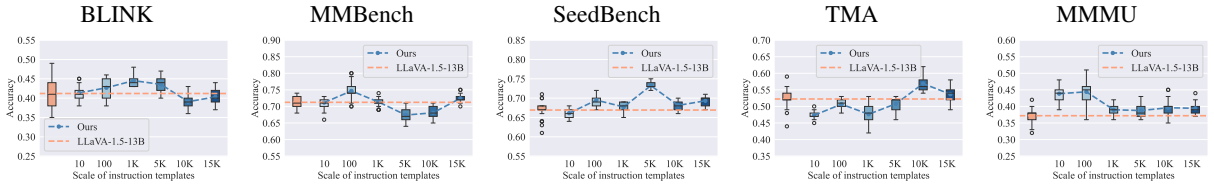
Figure 3 presents scaling curves for average performance across all five benchmarks as template scale increases, under both *in-domain* and *out-of-domain* evaluation template settings. Figure 4 shows per-dataset scaling curves under *out-of-domain* settings. Our analysis yields two key findings.

Training with diverse instruction templates substantially enhances MLM performance. Our empirical analysis, as demonstrated in Figure 3 and Figure 4, reveals that models trained with diversified instruction templates (spanning 10 to 15K distinct templates) consistently outperform baselines trained exclusively on original instruction tuning data under most cases. This improvement is evident in the average performance and individual performance across all five benchmark datasets and holds true across different model parameter sizes (7B and 13B) as well as for both in-domain and out-of-domain evaluation template settings. The consistency of these improvements across varied experimental conditions provides compelling evidence for the critical role of template diversity and underscore the significance of our investigation into scaling effects for MLM training.

MLM capability presents clear scaling trend: optimal template scale falls on the middle scale. As illustrated in Figure 3, the model's average performance across all five benchmarks 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) and evaluation settings (*in-domain* and *out-of-domain*). 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 smaller scale of 100 templates. This discrepancy suggests that models with stronger base capabilities require fewer templates to perform optimally. 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. To further support this finding, we present additional template scaling results in Appendix C and a theoretical analysis in Appendix D.



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



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

Figure 4: Scaling trends of model performance on *out-of-domain* evaluation templates with increasing training template scale across each individual benchmark dataset. We also show the performance spread across models and datasets. **Optimal template scale always falls on the middle scale.**

4 Evaluating MLM on Synthetic Instruction Templates

To demonstrate the practical impact of scaling effect of instruction templates for training MLMs, we compare model performance trained on the optimal template scale we observed 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 training 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 (Liu et al., 2024a), which serve as our primary baselines. In addition, for the 7B parameter size, we selected LLaVA-Next-7B (Liu et al., 2024b), Qwen-VL-7B and Qwen-VL-Chat-7B (Bai et al., 2023), and IDEFICS2-8B (Laurençon et al., 2024) as additional baselines; for the 13B parameter size, we selected LLaVA-Next-13B (Liu et al., 2024b) as an additional baseline model. Notably, as shown in Table 1, each of these additional baseline models was finetuned on a substantially larger 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*}\n*Choices*: {*choices*},
(3) *Question*: {*question*}\n*Select from the following choices*: {*choices*}.

Evaluation Protocol. For the evaluation metric, in addition to the answer accuracy, we employ the following two measures to quantify MLM's performance fluctuation degree to instruction template variations, where lower values indicate better robustness to various instruction templates.
(1) *Max-Min*, which we follow (Sclar et al., 2023) to report the range between the best and worst accuracy across all evaluation instruction templates.
(2) *POSIX* (Chatterjee et al., 2024), which measures prompt sensitivity through the average log-probability ratio of model responses across various evaluation templates.

4.2 Main Results

As detailed in Table 1 and Table 2, We compare performance of our template-tuned 7B and 13B models against several prominent MLMs of similar parameter size. Table 1 presents model accuracy across three evaluation templates: *in-domain*, *out-of-domain*, and *simple* templates and Table 2 quan-

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	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
LLaVA-Next-7B	760k	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
Qwen-VL-7B	50M	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
Qwen-VL-Chat-7B	50M	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
IDEFICS2-8B	1.8M	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
LLaVA-1.5-7B w/ Ours	665K	**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**
13B Models																	
LLaVA-1.5-13B	665K	**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
LLaVA-Next-13B	760k	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
LLaVA-1.5-13B w/ Ours	665K	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**

Table 1: Comparing average accuracy of our tuned models trained under the optimal template scale against similar-scale MLMs. The 7B model was trained with 5K templates and the 13B model was trained with 100 templates. # IT-Data is the size of training data the model used. S indicates the evaluation of three commonly used *simple* templates, ID refers to the evaluation of 100 *in-domain* instruction templates that our template-tuned model has encountered during training, and OOD denotes the evaluation of 25 manually crafted *out-of-domain* templates not included in our instruction template generator's template space. The best results are marked in **bold** and the second best in underline. **Training with optimal template scale can boost performance.**

tifies prompt sensitivity of MLMs under *out-of-domain* templates, which reveal two key insights.

Training on the optimal template scale yields substantial performance gains without data augmentation. As presented in Table 1, our template-tuned models, which leverage the same pretrained backbones as LLaVA-1.5-7B/13B but are trained with optimally scaled instruction templates, consistently achieve substantial performance improvements across most benchmarks and evaluation settings. Notably, our models trained under the optimal template scale surpass other leading MLMs of comparable parameter size, even these models being trained on significantly larger datasets (up to 75.19 times larger). These findings highlight the efficiency of our approach, demonstrating that appropriate scaling of instruction templates enables significant performance enhancement without reliance on extensive data scaling. By prioritizing the quality and diversity of instruction templates over brute-force increases in training data volume, our method provides a more resource-efficient pathway to improve MLM.

Training on the optimal template scale significantly mitigates MLMs' sensitivity to diverse instruction templates. Table 2 presents a comprehensive prompt sensitivity study on our models trained at the optimal template scale and other MLMs of comparable parameter sizes, evaluated on *out-of-domain* templates using *Max-Min* and *POSIX* (Chatterjee et al., 2024) metrics. Our empirical results demonstrate substantial improvements in model robustness. Compared to the baseline models LLaVA-1.5-7B/13B, which were trained on original training data, our approach consistently reduces both the *Max-Min* range and *POSIX* scores across most cases. Notably, when benchmarked against other strong MLMs of similar parameter size, our template-tuned models exhibit markedly lower prompt sensitivity degrees, indicating superior stability when encountering varied instruction formats. Crucially, the observed sensitivity reduction on *out-of-domain* templates—which lie entirely outside the template space of our template generator during training process—demonstrates the strong generalization capability of our approach in enhancing model robustness. This finding suggests that training on the optimal template scale enables models to develop more robust instruction-following capabilities that transfer effectively to unseen template variations, rather than simply memorizing specific formatting patterns. Such generalization is particularly valuable for real-world deployment scenarios where instruction formats can vary unpredictably across different users and applications. We provide additional MLM prompt sensitivity experiments in Appendix E.

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 model (LLM) backbones (Lyu et al., 2023; Bi et al.,

Model	# IT-Data	BLINK		MMB		SeedB		TMA		MMMU		Overall	
		Max-Min	*POSIX*	*Max-Min*	*POSIX*	*Max-Min*	*POSIX*	*Max-Min*	*POSIX*	*Max-Min*	*POSIX*	*Max-Min*	*POSIX*
7B / 8B Models													
LLaVA-1.5-7B	665K	15.00	**1.75**	9.00	2.57	16.00	3.64	18.00	**1.86**	13.00	2.69	14.20	2.50
LLaVA-Next-7B	760k	12.00	2.16	9.00	2.21	10.00	3.25	11.00	2.44	8.00	2.48	10.00	2.51
Qwen-VL-7B	50M	8.00	2.04	11.00	**1.47**	12.00	3.38	14.00	2.66	11.00	2.60	11.20	2.43
Qwen-VL-Chat-7B	50M	20.00	3.22	14.00	2.73	13.00	3.00	12.00	3.37	10.00	4.34	13.80	3.33
IDEFICS2-8B	1.8M	10.00	1.88	9.00	2.14	17.00	2.51	17.00	2.45	11.00	1.23	12.80	2.04
LLaVA-1.5-7B w/ Ours	665K	**2.55**	1.86	**8.00**	2.45	**6.00**	**2.44**	**10.00**	2.15	**6.00**	**1.13**	**6.51**	**2.01**
13B Models													
LLaVA-1.5-13B	665K	14.00	3.03	**6.00**	2.80	10.00	1.59	15.00	1.90	**10.00**	2.33	11.00	2.33
LLaVA-Next-13B	760k	13.00	2.55	15.00	1.78	11.00	2.21	14.00	**1.42**	**10.00**	2.18	12.60	2.03
LLaVA-1.5-13B w/ Ours	665K	**8.00**	**2.49**	10.00	**1.72**	**5.00**	**1.57**	**5.00**	1.85	15.00	**1.63**	**8.60**	**1.85**

Table 2: Comparing prompt sensitivity of models trained with the optimal template scale under *out-of-domain* templates against similar-scale MLMs. The 7B model was trained with 5K templates and the 13B model was traind with 100 templates. The best results are marked in **bold** and the second best in underline. **Training with optimal template scale can mitigate prompt sensitivity at most cases.**

2023; Liu et al., 2024d; Peng et al., 2023; Chen et al., 2023a; Lin et al., 2023; Lu et al., 2023; Li et al., 2023a; Awadalla et al., 2023; Sun et al., 2024; Xue et al., 2024a; An et al., 2025). 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), QwenVL series (Bai et al., 2023; Wang et al., 2024), LLaVA series (Liu et al., 2024c, 2023, 2024b), and InternVL series (Chen et al., 2023b, 2024), 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; 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 on the visual instruction tuning process, resulting in trained MLMs that are sensitive to subtle variations in instructions.

Influence of template perturbation. Recent research illustrated how prompt perturbations affect the performance and robustness of large language models (LLMs) and MLMs (Zhang et al., 2024a; Liu et al., 2024e; Madaan et al., 2023; Zhuo et al., 2024; Gan and 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 (Mizrahi et al., 2024; Voronov et al., 2024) have also 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 Conclusion

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

Limitations

Designing the template space requires manual effort. The development of meta templates and the association of placeholders with semantically equivalent synonyms demand minimal manual intervention. While automated methods streamline template generation, a small yet critical amount of human oversight is required to maintain quality.

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.

Ethics Statement

This research contributes to responsible AI development by improving multimodal language model robustness through instruction template scaling. All datasets used are publicly available and comply with their respective licensing agreements, with no personally identifiable information collected or processed. While our programmatic template generation aims to enhance model generalization and reduce prompt sensitivity, we acknowledge that automatically generated templates may inadvertently perpetuate biases present in underlying linguistic patterns. We emphasize that improved model performance should not eliminate the need for careful evaluation in deployment scenarios, as more capable models could potentially be misused without appropriate safeguards.

References

Ruichuan An, Kai Zeng, Ming Lu, Sihan Yang, Renrui Zhang, Huitong Ji, Qizhe Zhang, Yulin Luo, Hao Liang, and Wentao Zhang. 2025. Concept-as-tree: Synthetic data is all you need for vlm personalization. *arXiv preprint arXiv:2503.12999*.

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. 2023. Openflamingo: An open-source framework for training large autoregressive vision-language models. *Preprint*, arXiv:2308.01390.

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

Jing Bi, Nguyen Manh Nguyen, Ali Vosoughi, and Chenliang Xu. 2023. Misar: A multimodal instructional system with augmented reality. *Preprint*, arXiv:2310.11699.

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

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. 2023a. Pali-3 vision language models: Smaller, faster, stronger. *Preprint*, arXiv:2310.09199.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, and 1 others. 2024. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*.

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. 2023b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*.

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

Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. 2024. 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.

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

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

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

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

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

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

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2023a. Mimic-it: Multi-modal in-context instruction tuning. *Preprint*, arXiv:2306.05425.

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

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

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

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

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

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. 2023. Mm-vid: Advancing video understanding with gpt-4v(ision). *Preprint*, arXiv:2310.19773.

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

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

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

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

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

Shikun Liu, Linxi Fan, Edward Johns, Zhiding Yu, Chaowei Xiao, and Anima Anandkumar. 2024d. Prismer: A vision-language model with multi-task experts. *Preprint*, arXiv:2303.02506.

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

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2025. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pages 216–233. Springer.

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

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. *Preprint*, arXiv:2304.09842.

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

Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. 2023. Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration. *Preprint*, arXiv:2306.09093.

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

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

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

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

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. 2023. Kosmos-2: Grounding multimodal large language models to the world. *Preprint*, arXiv:2306.14824.

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

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, and 1 others. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. 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*.

Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. 2024. Generative multimodal models are in-context learners. *Preprint*, arXiv:2312.13286.

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

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. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

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, and 8 others. 2024a. xgen-mm (blip-3): A family of open large multimodal models. *ArXiv*, abs/2408.08872.

Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, and 1 others. 2024b. xgen-mm (blip-3): A family of open large multimodal models. *arXiv preprint arXiv:2408.08872*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. 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.

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

Jieyu Zhang, Le Xue, Linxin Song, Jun Wang, Weikai Huang, Manli Shu, An Yan, Zixian Ma, Juan Carlos Niebles, Silvio Savarese, and 1 others. 2024b. Provision: Programmatically scaling vision-centric instruction data for multimodal language models. *arXiv preprint arXiv:2412.07012*.

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

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

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

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

Overview. In the supplementary material, we first detail the implementation of our programmatic instruction template generator in Sec. A, including details of the *sentence pattern tree* along with diverse *meta templates*. Subsequently, we present a quantitative analysis that examine the diversity of instruction templates generated by our template generator in Sec. B. Sec. C reports additional experiments exploring the effects of template scaling. In Sec. D, we offer a theoretical analysis elucidating why the optimal template scale is situated in the mid-range. Finally, Sec. E presents further experiments and analyses on the sensitivity of MLMs to instruction templates.

A Details of Our Programmatic Instruction Template Generator

Our instruction template generator is capable of producing an extensive template space, encompassing approximately 15,000 unique visual instruction templates. The generation process begins by sampling meta templates from a sentence pattern tree using a weighted sampling algorithm. Subsequently, the sampled *meta templates* are programmatically populated by replacing placeholders with randomly selected positional synonyms, thereby promoting diversity in the instructions. In this section, we provide a comprehensive overview of our sentence pattern trees and the design of the meta templates. Specifically, we construct sentence pattern trees tailored for visual instructions that comprise a total of 24 distinct meta-templates. The taxonomy and specific meta templates utilized in our sentence pattern tree for visual instructions are illustrated in Figure 5.

B Quantitative Analysis of Instruction Template Diversity

We quantitatively compare the diversity of templates from our programmatic instruction template generator with an LLM-based generation method (leveraging GPT-4o (Hurst et al., 2024)), which measured by both *lexical diversity* and *syntactic diversity*.

Evaluation protocol. To establish a rigorous comparative evaluation, we employ a standardized experimental setup where both our method and GPT-4o generate exactly 100 VQA instruction templates under identical constraints. Specifically, each template must satisfy three strict requirements: (1) explicit reference to visual input, (2) inclusion of precisely two placeholders (`{question}` and `{choices}`), and (3) structural uniqueness without placeholder modification. We prompt GPT-4o using the carefully designed query template we show. Our programmatic template generation approach operates under identical constraints to ensure fair comparison.

Query Template

Task: Generate 100 instruction templates for multi-choice VQA. Each template must:

- answer a question based on an image
- Include exactly two placeholders in the template body: `{question}` and `{choices}`
- Require selecting an answer from options
- Maintain consistent semantic meaning

Output Format: Return a list of 100 templates.
Constraints:

- Do not add/remove placeholders
- Avoid duplicate phrasing patterns

Diversity metrics. We evaluate template diversity through two complementary and well-established metrics: (1) *lexical diversity*, quantified as the cardinality of unique vocabulary tokens across all generated templates, capturing the breadth of linguistic expressions; and (2) *syntactic diversity*, measured as the mean depth of constituency parse trees, reflecting structural complexity and grammatical variation in template constructions.

Diversity Metric	GPT-4o	Ours
Lexical Diversity	57	**84**
Syntactic Diversity	4.66	**5.16**

Table 3: Template diversity comparison on 100 generated VQA instruction templates.

Results and discussion. Table 3 presents compelling evidence for the superiority of our programmatic approach. Our method achieves substantial improvements across both diversity dimensions, with both increase in lexical diversity (84 vs. 57

```
+ 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}
```

Figure 5: 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).

unique tokens) and in syntactic diversity (5.16 vs. 4.66 average parse depth). These results demonstrate that **our programmatic template generation method consistently generates more diverse instruction templates compared to state-of-the-art LLM-based generation**. Beyond diversity gains, our programmatic framework offers significant practical advantages in terms of computational efficiency and scalability. Unlike LLM-based approaches that require expensive API calls and substantial inference time, our method enables rapid, large-scale template generation with minimal computational overhead.

C Additional Experiments on Impact of Template Scaling

To establish the generalizability of our template scaling findings, we extend our investigation to the LLaVA-NeXT (Liu et al., 2024b) architecture. Since the official LLaVA-NeXT training data remains proprietary—preventing direct replication as discussed in Sec. 3.1—we utilize a community reproduction version, Open-LLaVA-Next[1]. While this reproduced implementation may differ from the official version in terms of training data, it enables us to validate our findings across different model architectures, thereby strengthening the generalizability of our conclusions.

Experimental Setup. We trained two Open-LLaVA-NeXT-7B variants using 5K and 15K templates respectively, employing identical template-tuning protocols as outlined in Sec. 3. Evaluation encompasses 25 *out-of-domain* templates across five established benchmarks: BLINK, MMBench, SeedBench, TaskMeAnything, and MMMU, with the vanilla Open-LLaVA-NeXT-7B serving as baseline model.

Results. Table 4 demonstrates our principal findings: Optimal performance is achieved at a medium template scale. The 5K template configuration achieves optimal overall performance (44.11%), surpassing both baseline (42.54%) and the 15K variant (43.46%). These results demonstrate the

[1] https://github.com/xiaoachen98/Open-LLaVA-NeXT

Model	BLINK	MMB	SeedB	TMA	MMMU	Overall
Open-LLaVA-NeXT-7B	35.16	54.04	52.96	41.68	28.84	42.54
+ 5K templates	34.62	55.81	55.16	42.08	32.88	**44.11**
+ 15K templates	35.92	55.24	52.20	42.84	31.08	43.46

Table 4: Template scaling experiments on Open-LLaVA-NeXT-7B across five benchmarks. **Optimal performance is achieved at a medium template scale**

generalizability of our conclusion across different architectures and strengthen our contribution by proving the robustness of our findings beyond the LLaVA 1.5 framework.

D Theoretical Analysis of Optimal Instruction Template Scale

Problem Formulation. We propose a theoretical analysis to study the effect of instruction template scale on MLM performance. Let S be the number of distinct instruction templates, and let $\mathcal{L}(S)$ denote the expected risk as a function of S. Motivated by the bias-variance trade-off, we posit that the risk can be decomposed as follows:

$$\mathcal{L}(S) = \text{Bias}(S) + \text{Variance}(S) + \sigma^2. \quad (1)$$

Here, σ^2 represents the irreducible error. We model the two main components based on the following properties of template diversity:
(1) *Bias*: We model the bias as $\text{Bias}(S) = \alpha S$ for a constant $\alpha > 0$. An excessively large template set introduces task-irrelevant linguistic noise, which can obscure the underlying task semantics and thus increase the model's bias.
(2) *Variance*: We model the variance as $\text{Variance}(S) = \frac{\beta}{S}$ for a constant $\beta > 0$. This captures the principle that a small template set provides insufficient coverage of linguistic variations, making the model prone to overfitting on specific phrasings and thus increasing variance.
Combining these terms, the total risk is modeled by:

$$\mathcal{L}(S) = \alpha S + \frac{\beta}{S} + \sigma^2. \quad (2)$$

The constant α captures sensitivity to task-irrelevant variations and β represents template-specific overfitting risk.
Analysis of the Optimal Scale. We seek the scale S^* that minimizes the risk $\mathcal{L}(S)$. Differentiating Eq. (2) with respect to S gives:

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}S} = \alpha - \frac{\beta}{S^2}. \quad (3)$$

Setting the derivative to zero yields a unique critical point for $S > 0$:

$$\alpha - \frac{\beta}{S^2} = 0 \quad \Longrightarrow \quad S^* = \sqrt{\frac{\beta}{\alpha}}. \quad (4)$$

The second derivative, $\frac{\mathrm{d}^2\mathcal{L}}{\mathrm{d}S^2} = \frac{2\beta}{S^3}$, is strictly positive for all $S > 0$. This confirms that $\mathcal{L}(S)$ is strictly convex, and thus S^* is its unique global minimum. The minimum achievable risk is found by substituting S^* back into Eq. (2):

$$\mathcal{L}(S^*) = \alpha\left(\sqrt{\frac{\beta}{\alpha}}\right) + \frac{\beta}{\sqrt{\frac{\beta}{\alpha}}} + \sigma^2 = 2\sqrt{\alpha\beta} + \sigma^2. \quad (5)$$

The excess risk incurred by deviating from S^* can be approximated by a second-order Taylor expansion around S^*, which shows a quadratic penalty:

$$\begin{aligned}\mathcal{L}(S) - \mathcal{L}(S^*) &\approx \frac{1}{2}\left(\frac{2\beta}{(S^*)^3}\right)(S - S^*)^2 \\ &= \frac{\alpha}{S^*}(S - S^*)^2.\end{aligned} \quad (6)$$

Implications of the Analysis. This theoretical analysis provides a formal grounding for the empirically observed phenomenon that a middle template scale is optimal. We draw three main conclusions:

(1) **An Optimal Scale Exists and is Task-Dependent.** The optimal number of templates, $S^* = \sqrt{\beta/\alpha}$, is determined by the intrinsic properties of the learning problem, specifically the ratio of variance sensitivity (β) to bias sensitivity (α). This refutes the intuition that performance should monotonically increase with template quantity.

(2) **A Fundamental Performance Bound.** The minimum achievable risk is lower-bounded by $\mathcal{L}^* = 2\sqrt{\alpha\beta} + \sigma^2$. This limit depends on the irreducible error (σ^2) and the best possible balance between template-induced bias and variance.

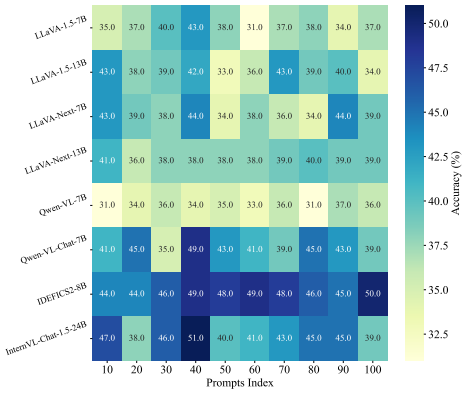


Figure 6: Heat map illustrating the performance variations of eight MLMs on the BLINK dataset across ten instruction templates (selected from the *in-domain* templates set). The darker the color, the better the performance. **No single instruction template performs optimally for all MLMs.**

(3) **Quadratic Penalty for Deviation.** The risk increases quadratically with the distance from the optimal scale S^*. This provides a theoretical explanation for why performance degrades sharply for both overly small (variance-dominated) and overly large (bias-dominated) template sets.

E Additional Experiments on MLM's Sensitivity to Instruction Templates

In this section, we investigate whether a universally effective instruction template exists across widely used multimodal language models (MLMs). To this end, we evaluate the performance of eight prominent MLMs on the BLINK dataset, utilizing ten instruction templates sampled from the *in-domain* template set. Figure 6 presents a heat map depicting performance variations, where darker shades denote higher accuracy.

The results reveal pronounced performance heterogeneity across MLMs for identical instruction templates, as reflected by the diverse intensities within each column of the heat map. Notably, **no single instruction template emerges as consistently optimal across all models**. This lack of a universal template highlights the unique sensitivities and preferences inherent to individual MLM architectures.

These findings carry significant implications for the design and evaluation of instruction templates in MLMs. In particular, they underscore the limitations of applying uniform instruction optimization strategies across different models. A one-size-fits-all approach is unlikely to yield robust or transferable gains. As such, developing tailored instruction engineering procedures—sensitive to the idiosyncrasies and underlying mechanisms of each MLM—may be indispensable for attaining optimal empirical performance.