

MM-IFEngine: Towards Multimodal Instruction Following

Shengyuan Ding^{1,2*}, Shenxi Wu^{1,2*}, Xiangyu Zhao^{2,3}, Yuhang Zang^{2✉},
 Haodong Duan², Xiaoyi Dong², Pan Zhang², Yuhang Cao², Dahua Lin^{2,4,5}, Jiaqi Wang^{2,6✉}
¹Fudan University ²Shanghai AI Laboratory ³Shanghai Jiaotong University
⁴The Chinese University of Hong Kong ⁵CPII under InnoHK ⁶Shanghai Innovation Institute

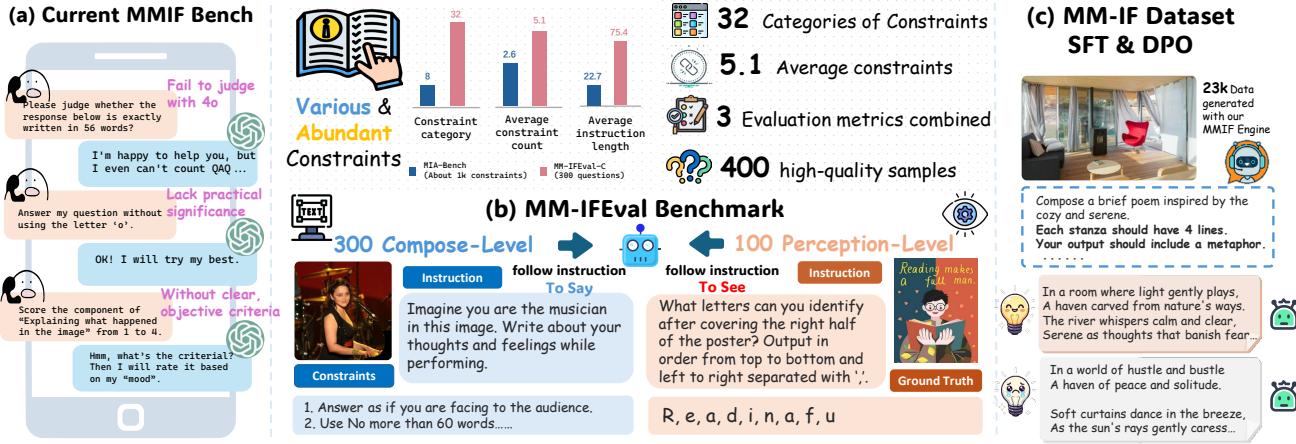


Figure 1. **(a)** Limitations of existing Multimodal Instruction Following (IF) benchmarks. **(b)** Overview of the MM-IFEval benchmark, which significantly surpasses existing benchmarks in terms of constraint diversity, quantity, and instruction complexity. Our benchmark consists of Compose-Level (C-Level) problems that impose constraints on model outputs (e.g., format requirements, keyword limits) and Perception-Level (P-Level) problems that require reasoning about specific visual elements in images. **(c)** Our MM-IFEngine generates a large-scale, diverse training dataset suitable for both Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO).

Abstract

The Instruction Following (IF) ability measures how well Multi-modal Large Language Models (MLLMs) understand exactly what users are telling them and whether they are doing it right. Existing multimodal instruction following training data is scarce, the benchmarks are simple with atomic instructions, and the evaluation strategies are imprecise for tasks demanding exact output constraints. To address this, we present MM-IFEngine, an effective pipeline to generate high-quality image-instruction pairs. Our MM-IFEngine pipeline yields large-scale, diverse, and high-quality training data MM-IFIInstruct-23k, which is suitable for Supervised Fine-Tuning (SFT) and extended as MM-IFDPO-23k for Direct Preference Optimization (DPO). We further introduce MM-IFEval, a challenging and diverse multi-modal instruction-following benchmark that includes (1) both compose-level constraints for output responses and perception-level constraints tied to the input images, and (2) a comprehensive evaluation pipeline incorporating both rule-based assessment and judge model. We con-

duct SFT and DPO experiments and demonstrate that fine-tuning MLLMs on MM-IFIInstruct-23k and MM-IFDPO-23k achieves notable gains on various IF benchmarks, such as MM-IFEval (+10.2%), MIA (+7.6%), and IFEval (+12.3%). The full data and evaluation code will be released on <https://github.com/SYuan03/MM-IFEngine>.

1. Introduction

Instruction Following (IF) is a fundamental ability in Large Language Models (LLMs) [14, 27, 35, 51, 55] and Multimodal Large Language Models (MLLMs) [2, 34], which involves accurately interpreting and executing user-provided instructions. This ability is crucial for deploying models in real-world applications where users expect precise and context-aware responses, such as code generation [43], visual question answering [17], robots [38], and creative content creation [56]. For instance, in a VQA scenario, when a user asks an MLLM what is the object and how do I use it, return the object name and the usage instructions

in a JSON format, accurate IF ensures the model provides a response like `{object': 'hammer', 'usage': 'use it to drive nails'}` instead of the plain text.

Achieving precise IF in multimodal, diverse, and open-ended environments presents significant challenges for both *model training* and *benchmark evaluation*. One significant limitation is the scarcity of high-quality IF training data to train open-source MLLMs. In addition, current multimodal IF benchmarks [2, 34] merely have simple, atomic instructions, and the constraints are weakly correlated with visual content (see Fig. 1 (a)). Consequently, existing benchmarks lack the diversity required for real-world applications, leading to saturated results where nearly all models achieve over 80%. Furthermore, the evaluation method in existing benchmarks often relies on LLM-as-a-judge [54], which is imprecise for instructions demanding exact output constraints, such as word counts. Therefore, the combination of *limited training data*, *simple benchmarks*, and *imprecise evaluation strategy* strongly restricts the progress of current MLLMs in IF.

To address the lack of high-quality IF training data and challenging benchmarks, we propose **MM-IFEngine**, an effective pipeline for generating high-quality image-instruction pairs. MM-IFEngine collects diverse image sources, including natural scenes, UI interfaces, diagrams, charts, and mathematical problems. We then employ a structured approach using a predefined set of 16 task descriptions and 32 constraints to guide the LLM in crafting tailored instructions for each image. Using MM-IFEngine, we generated a comprehensive dataset of image-instruction pairs, collected responses from open-source MLLMs, and applied rigorous post-processing to retain only high-quality instruction-answer pairs, thus constructing **MM-IFIInstruct-23k** for Supervised Fine-Tuning (SFT). We also generate negative responses by selectively removing constraints from the original data, constructing the preference dataset **MM-IFDPO-23k** for preference optimization algorithms such as Direct Preference Optimization (DPO) [36].

To facilitate the evaluation of multimodal IF, we present **MM-IFEval**, a benchmark comprising 400 challenging problems with diverse compose-level and perception-level instructions. MM-IFEval is derived from the images and instructions generated by MM-IFEngine with human-labeled annotations. As presented in Fig. 1 (b), our MM-IFEval has the following three distinctive features: (1) **Diverse Instruction Types**: MM-IFEval has 32 distinct constraints, ensuring a wide range of instruction complexities and surpassing the scope of prior benchmarks. (2) **Hybrid Evaluation**: we use a hybrid strategy including both rule-based verification and judge model. For subjective instructions (e.g., mimicking tone), we design a *comparative* judgment for precise evaluation. Specifically, a control output is generated without the constraint, and the LLM judge compares both outputs

for precise evaluation. (3) **Challenging**: the leading proprietary model (GPT-4o at 64.6%) and open-source model (Qwen2-VL-72B at 50.8%) demonstrating substantial room for improvement on our benchmark, highlights a significant opportunity for improvement in multimodal instruction following.

We further demonstrate that fine-tuning MLLMs on either MM-IFIInstruct-23k or MM-IFDPO-23k consistently boosts the performance of MLLMs on instruction following benchmarks, without compromising their original capabilities on other Visual Question Answering (VQA) benchmarks. Specifically, fine-tuning Qwen2-VL-7B on MM-IFDPO-23k with the DPO results in performance gains of 10.2%, 7.6%, and 12.3% on MM-IFIInstruct-23k, MIA-Bench [34], and IFEval [55], respectively.

Our contributions include: (1) a MM-IFEngine pipeline for generating multimodal constraint-rich image-instruction pairs; (2) a large-scale training dataset MM-IFIInstruct-23k and preference optimization dataset MM-IFDPO-23k derived from MM-IFEngine; (3) a challenging multimodal instruction following benchmark MM-IFEval with diverse constraints and comprehensive evaluation approaches; and (4) empirical evidence showing significant performance gains on both our MM-IFEval and existing benchmarks when training MLLMs on MM-IFIInstruct-23k via SFT and MM-IFDPO-23k via DPO.

2. Related Work

Instruction Following in LLMs. Various benchmarks and training approaches have been proposed to make Large Language Models (LLMs) better align with human instructions. While existing Instruction Following (IF) benchmarks like [14, 35, 51, 55] all aim to evaluate instruction following, they differ significantly in their *dataset construction pipelines*, driven by their unique constraint taxonomies. CFBench [51], for instance, constructs its dataset using a combination of taxonomic and statistical methodologies to establish comprehensive constraints. This divergence extends to their *evaluation strategies*. For example, InFoBench [35] adopts a strategy of decomposing complex instructions into simpler assessment standards. Beyond benchmarks, various training approaches aim to enhance LLMs’ instruction-following capabilities [29, 43], including in-context learning [56] and preference optimization [52]. However, the aforementioned research is limited to the text modality, whereas our work focuses on multi-modal instruction following with vision inputs.

Instruction Following Benchmarks in MLLMs. Numerous benchmarks [18] have been proposed to evaluate diverse capabilities of Multi-modal Large Language Models (MLLMs), including general knowledge [5, 24, 47, 49], document understanding [15, 25, 30], perception [50], multi-image comprehension [26, 39, 40], and instruction following

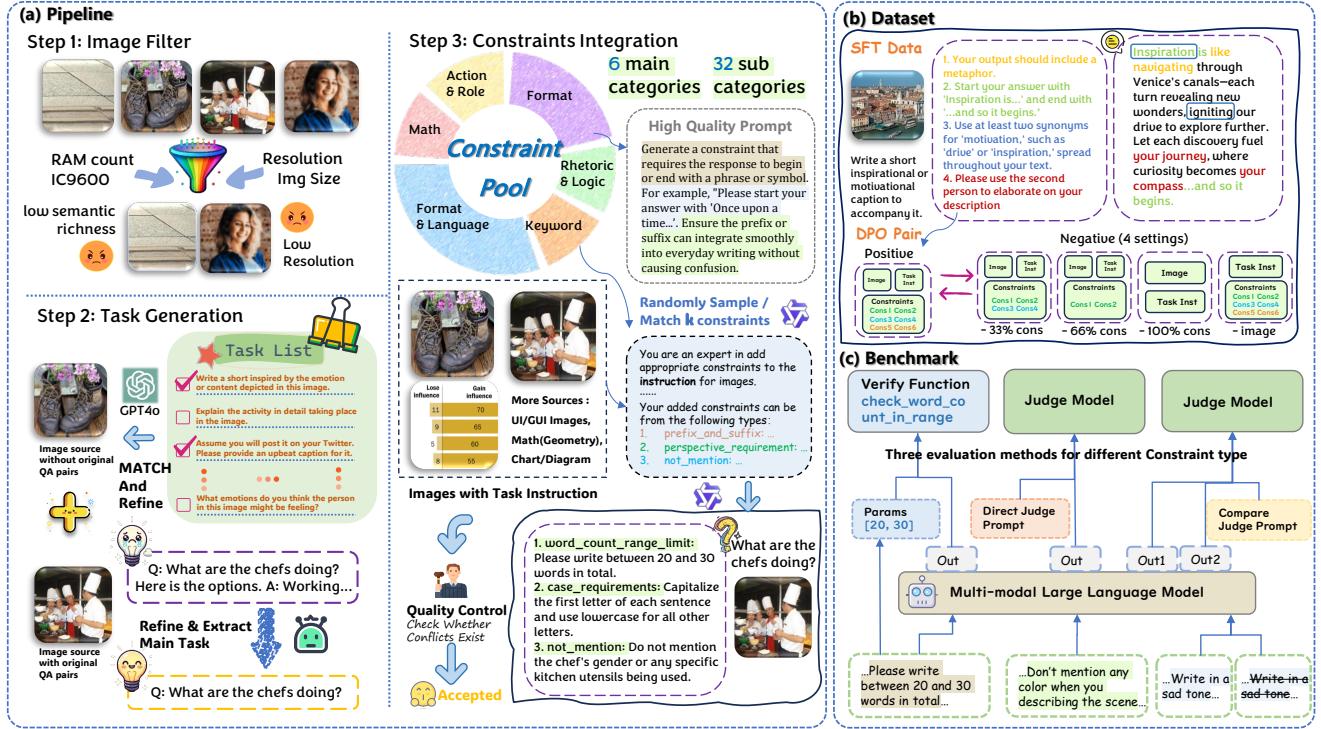


Figure 2. Overall pipeline of MM-IFEngine. Part (a) demonstrates the three-stage workflow of our engine: (1) Image filter; (2) Task generation using GPT-4o for images without QA pairs and instruct refinement for existing annotations; and (3) Constraints integration incorporating 6 main categories and 32 subcategories, ensuring compatibility between constraints and tasks. MM-IFEngine is employed to generate SFT and DPO training datasets and MM-IFEval benchmark, as shown in part (b) and (c). MM-IFEval implements three evaluation metrics combining rule-based verification functions and a judge model to ensure accurate assessment.

(IF) [2, 34]. MIA-Bench [34] and VisIT-Bench [2] are representative IF benchmarks that employ GPT-4 [32] for question generation and evaluation. In contrast to existing IF benchmarks, our MM-IFEval introduces significant improvements in diversity (32 constraint categories covering compositional and perceptual aspects), difficulty (averaging 5.1 constraints per question), and evaluation precision (using both judge models and rule-based verification).

Instruction Tuning Data for MLLMs. Recent advancements in multi-modal instruction tuning data aim to improve cross-modal alignment and increase the variety of tasks handled by MLLMs [4, 8, 20, 26, 44, 45]. For example, some previous works [3, 4, 23] build synthetic instruction tuning data generated using GPT-4V [33], enabling open-source MLLMs to achieve performance comparable to proprietary models across multiple benchmarks. However, existing instruction tuning data are mainly designed for general knowledge or visual perception, and data for improving the IF abilities is scarce. The scarcity of training data for enhancing IF abilities motivated the development of our MM-IFEngine pipeline.

3. MM-IFEngine

We employ the MM-IFEngine pipeline to generate image-instruction pairs, which are the foundation for creating instruction tuning data and our benchmark. As shown in Fig. 2 (a), the pipeline has three steps: (1) image filtering for the selection of diverse image sources, (2) task generation to incorporate constraints into existing instruction tuning data, and (3) constraints integration to produce high-quality constrained instructions for images lacking annotated instructions.

3.1. Image Filter

Our image filtering strategy selects only high-quality images by removing those with low resolution or limited semantic richness. For unannotated pure image datasets (e.g., CC3M [37]), we prioritize natural scene images. Rich semantic content in these images enables the creation of more comprehensive and insightful QA pairs, which is crucial for designing diverse and complex instruction following tasks. We use the IC9600 and RAM metric proposed in the previous method [53] to select the images that have rich semantic content.

Furthermore, we analyze existing annotated datasets, such as ALLaVA [3]. Our analysis reveals that some images suffer from low resolution, making them inadequate for the instruction-following task. Given our intention to design more intricate and varied instruction following tasks based on this data, we filter out data items containing low-quality images.

3.2. Task Generation

Image Source without Original QA Pairs. For image datasets lacking original annotated task instructions (e.g., CC3M [37]), we first design appropriate task instructions for the data items. We first develop a series of task instructions tailored to the data items. These instructions are crafted to elicit long-form responses that can be subsequently modified or refined using various constraints, for instance, *Provide a detailed analysis of the image, including the setting, characters, and notable objects*. The final task pool \mathcal{P}_T comprises a total of 16 distinct tasks, with further details available in Appendix A.1.2.

Given the task pool \mathcal{P}_T , we randomly select k tasks as examples of task types for each image I . We then prompt a powerful language model \mathcal{M} (e.g., GPT-4o) to generate an appropriate task list T_l that aligns with the image content. The process is formulated as:

$$\{T_l^*\} = \mathcal{M}(I, T_e) \quad (1)$$

where $T_e = \{T_1, T_2, \dots, T_k\}$ and each $T_i \in \mathcal{P}_T$. The model \mathcal{M} is tasked with either choosing relevant tasks from T_e or supplementing reasonable tasks to construct the appropriate task list T_l^* , ensuring that all tasks in T_l^* are in line with the image content. After generating the T_l^* , a sampling step is incorporated to guarantee task diversity. For each image, tasks are sampled. This sampling process is crucial as it enriches the variety of tasks associated with each image.

Image Source with QA Pairs. In the case of image datasets that have QA pairs (e.g., ALLaVA [3]), we adopt certain strategies for processing the original question annotations. We choose ALLaVA as the primary dataset for this type of image source due to its rich and diverse image content, which is accompanied by a variety of task types. First, we conduct an analysis of the original question annotations. We find that some of the questions are accompanied by some few-shot examples. Additionally, some questions in ALLaVA have options in their original annotations, which are not suitable for our instruction-following task. Since we need to incorporate certain constraints into the original instructions in the subsequent steps, we use regular expressions and length limits to filter the questions in ALLaVA. Specifically, we select those questions that do not have few-shot examples associated with them. Mathematically, if we let Q be the set of all questions in ALLaVA, Q_{fs} be the subset of questions with few-shot examples, and Q_{op} be the subset of questions

with options. We aim to find the subset Q_s of questions that satisfy the conditions:

$$Q_s = \{q \in Q | q \notin Q_{fs} \wedge q \notin Q_{op}\} \quad (2)$$

where the filtering based on the absence of few-shot examples and options is achieved using regular expressions and length limits. Then, we get the expected T^* in our filter Q_s set for the images.

3.3. Constraints Integration

Constraints Pool (\mathcal{P}_C) We use *instruction* to refer to the entire textual input, which in our paper can generally be viewed as a composition of *a task instruction* and *multiple constraints instruction*. Tasks and constraints are rich and diverse, with a certain complexity in our work. All the constraints in our work can be further classified into six major categories, each with its own unique characteristics and applications: Text Length Requirements, Mathematical Requirements, Language & Formatting Requirements, Rhetoric & Logic Requirements, Action Requirements, and Keyword Requirements. Please refer to the Appendix Fig. 5 for more details of all the constraints.

Given the constraints pool \mathcal{P}_C and task instructions, a straightforward approach for composing full instruction is to first set several constraints for each constraint type and then randomly select one constraint from some of the types to compose the constraint list, and finally concatenate the constraint list with the task instruction to form the full instruction. But this direct method has two problems: (1) The constraints are not diverse enough, which may not be able to fully evaluate the ability of the model. (2) The contradiction between the constraints and also between the constraints and the task instruction may exist. For the first problem, an LLM is employed to generate concrete content of constraint instruction for the specific constraint type in our method. In order to avoid the generated content being too divergent or hard to control its difficulty, we carefully design some cases or requirements of details that needed to be paid attention to when generating the content for each constraint type (Appendix A.1.1). For the second problem, we also use a powerful LLM to help keep the correlation of constraints with its instruction and filter out those that cause total contradiction. Finally, we prompt an LLM to check whether the constraints and the task instruction are compatible and filter out those failing to pass the check. Our method not only ensures the compatibility of constraints and instructions but also enriches the diversity of constraints.

In our actual practice process, we find that although we prompt the LLM to select appropriate constraints that should be compatible with the task instruction and other constraints, the generated constraints still have some contradiction with the task instruction, especially on those existing datasets with various kinds of annotations. The reason is that these datasets

are designed for overall question-answering tasks, and the question(or named task instruction) tends to be contradictory with the constraints, which are mostly compatible with those tasks of creating or answering in non-short form. So, we decouple the selection and generation steps for this type of data source. Specifically, we first select the constraints from the constraints pool \mathcal{P}_C and then provide the selected mostly compatible constraints to the LLM to select secondly and generate final constraints. But for image datasets without original QA pairs, in other words, for which we generate task instructions for them using \mathcal{P}_T , we directly sample k constraint types for the LLM to generate concrete content because they are mostly compatible with the pre-designed task instruction. The uniform process is formulated as:

$$C_l^* = \mathcal{L}(C_s, T^*), C_f^* = \mathcal{V}(C_l^*, T^*) \quad (3)$$

where T^* is the task applicable to the image. The model \mathcal{L} is tasked with both choosing appropriate constraint types from C_s again and generating concrete constraints for some of them, whose output is a list of concrete constraint descriptions. To ensure that the generated constraints remain compatible with the given task instruction T^* , we employ a final validation step using another LLM process, denoted as \mathcal{V} . This validation function checks whether each constraint in C_l^* aligns with T^* and filters out those that contradict or do not fit the task instruction. The resulting set of fully verified and compatible constraints is represented as C_f^* .

MM-IFIInstruct-23k Construction. By applying the MM-IFEngine pipeline, we construct the MM-IFIInstruct-23k dataset, which contains 23k high-quality multi-modal instruction-following training data. We first take an analysis of the performance of the current open-source MLLMs and proprietary MLLMs on several benchmarks [25, 34], and find that for instruction-following capability, the most powerful open-source MLLM like InternVL2.5-78B-MPO [42] is nearly equivalent to GPT-4o, and the performance on general VQA benchmarks are even higher then GPT-4o. Thus, we use InternVL2.5-78B-MPO to generate responses for our MM-IFIInstruct-23k dataset. Despite its capabilities, the InternVL2.5-78B-MPO model encounters difficulties in ensuring 100% compliance with our constraints, a challenge attributed to the complexity, number, and comprehensiveness. Consequently, we implement a post-processing stage to filter out responses that do not meet the specified criteria. Acknowledging that achieving perfect constraint adherence might be challenging even for human annotators on this task, we set a practical accuracy threshold of 80%. Finally, our MM-IFIInstruct-23k comprises 23k data items, with 16k constructed from the training set of CC3M, 6k from ALLaVA, and 4k from the training set of MultiUI, Geo170k[12] and ChartQA[31]. We show the distribution of constraints number of MM-IFIInstruct-23k in Fig. 3.

MM-IFDPO-23k Construction. To comprehensively ex-

plore and make full use of our high-quality data, we also utilize MM-IFEngine to construct MM-IFDPO-23k, a preference dataset comprising chosen and rejected samples suitable for Direct Preference Optimization (DPO) [36]. Our high-quality data can be directly employed as the chosen samples. Regarding rejected samples, we opt to utilize Qwen2-VL-7B-Instruct to answer the variant of the question for generating rejected pairs. Specifically, we have four distinct settings for generating negative pairs, which mainly differ in the input to Qwen2-VL-7B-Instruct. These settings include (1) With image, but randomly remove one-third of the number of constraints in the prompt; (2) With image, but randomly remove two-thirds of the number of constraints in the prompt; (3) With image, but randomly remove all the constraints in the prompt; and (4) Full prompt, but without the image; We use these four types of input to feed into Qwen2-VL-7B-Instruct model, and collect the rejected responses to construct the MM-IFDPO-23k.

4. MM-IFEval

Existing benchmarks for multi-modal instruction following are scarce. The majority focus on simple and atomic instructions, resulting in performance saturation across models. To address this limitation, we introduce **MM-IFEval**, a human-annotated, comprehensive, and challenging benchmark designed for evaluating multi-modal IF.

4.1. MM-IFEval Construction

To construct the MM-IFEval, we first use our MM-IFEngine to generate the question-answer (QA) pairs for images. The generated instructions may inherently contain potential conflicts. Consequently, human annotation remains critical for constructing this benchmark, as human annotators possess the cognitive capacity for comprehensive assessment of these complex situations. After the human annotation, we further use an extra post-processing step that prompts the LLMs to double-check and mitigate the occurrence of constraint conflicts as much as possible. Finally, we construct the MM-IFEval bench of 400 questions, 300 of which are *compose-level* open-ended questions and 100 *perception-level* questions with ground truth.

Diverse Constraints. With 32 distinct constraint categories and an average of 5.1 constraints per question, MM-IFEval presents a more challenging evaluation task compared to earlier benchmarks (e.g., [34], which has 8 categories and 2.6 average constraints per question). Furthermore, our benchmark incorporates essential constraints such as “Output in JSON format”, which is prevalent and practical in real-world scenarios, a feature not found in previous multi-modal instruction following benchmarks.

Compose-level and Perception-level Questions. *Compose-level* questions involve textual constraints, while *perception-level* questions require greater visual perception ability to

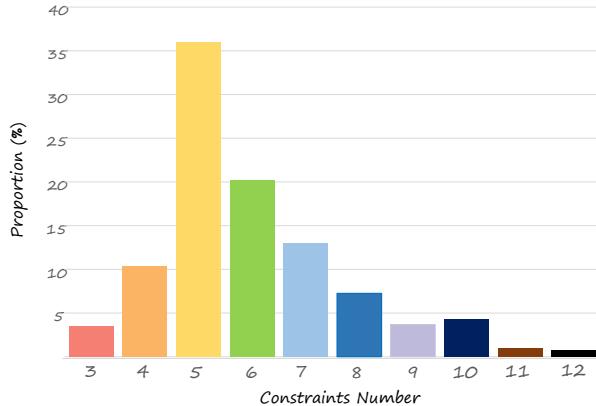


Figure 3. **Constraint Quantity Distribution in MM-IFInstruct-23k.** Our MM-IFInstruct-23k exhibits systematic variation in constraint complexity, with each sample containing 3-12 constraints per instruction.

solve. The perception-level questions incorporate a variety of image sources, such as natural scenes, user interfaces, diagrams, table charts, and mathematical expressions, which we believe are representative of real-world applications. Please refer to the Appendix for examples of compose-level and perception-level questions.

4.2. Hybrid Evaluation

Current multi-modal instruction following benchmarks often rely solely on GPT-4o for evaluation. However, accurately assessing certain constraints, such as numerical conditions (e.g., ‘output in 200 words’, ‘Answer in 5 paragraphs’, ‘Use the word ‘cat’ in the answer twice’), remains challenging even for GPT-4o. In contrast, verifiable functions like string matching offer greater precision than judge models for such constraints. To address this, we propose a hybrid evaluation strategy (see Fig. 2(c)) that employs three methods, including both rule-based Verification and judge models for more robust and precise evaluation.

(1) **Rule-based Verification.** For constraints that adhere to a fixed format and involve specific content that can be objectively verified—yet remain challenging for an LLM to assess accurately—we employ a rule-based approach. Specifically, we design a set of predefined functions for different constraint types. The LLM is first prompted to extract the relevant parameters, denoted as *Params*, from the constraint description. When evaluating a constraint that falls within the scope of our rule-based framework, we use *Params* and the model’s output as inputs to the predefined function to determine compliance.

(2) **LLM-based Direct Judgment.** This method is primarily used for evaluating constraints that can be easily and

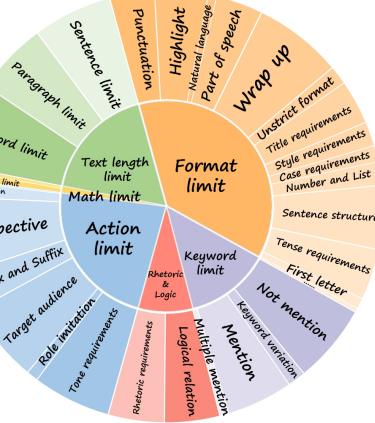


Figure 4. **Constraint Category Distribution in Compose-Level Problems of MM-IFEval.** This part comprises six primary constraint categories with 32 subcategories, forming a multi-level taxonomy for instruction-following evaluation.

unambiguously verified based on the model’s output. It is applicable to constraints where correctness is straightforward to determine, such as those requiring the inclusion of specific words or phrases. For instance, a constraint like “Use the word ‘inspiration’ or its synonyms at least twice in the response” does not follow a strict format and cannot be assessed using a rule-based approach. Instead, we directly leverage an LLM to determine whether the constraint is satisfied.

(3) **LLM-based Comparative Judgment.** Some constraints, particularly those related to tone, style, or role-playing, are difficult to evaluate directly. To improve judgment accuracy, we adopt a comparative approach. Specifically, we generate a second model output using a nearly identical prompt but without the constraint under evaluation. The LLM-based evaluator is then provided with both outputs and asked to compare them, determining whether the model’s response with the constraint in the prompt adheres more closely to the expected requirement.

5. Experiments

Benchmarks. We select the following benchmarks to demonstrate that models fine-tuned on MM-IFInstruct-23k and MM-IDPPO-23k enhance instruction following without compromising performance on other VQA tasks: (1) **Instruction Following benchmarks**, including MIA-Bench [34], IFEval [55], and our proposed MM-IFEval. To be noted, IFEval is a language-only benchmark while others are both multi-modal benchmarks. (2) **VQA Benchmarks**, including MMMU [49], MMBench [24], MMStar [5], AI2D [15], OCRBench [25], MMVet [48], POPE [19] and MMT-Bench [47].

Implementation Details. We conducted SFT and DPO fine-

Table 1. Main results on Instruction Following benchmarks, including our proposed MM-IFEval, MIA-Bench [34], and IFEval [55]. The symbol ^M refers to multimodal benchmarks, and ^T denotes text-only benchmarks. We report both compose-level (“C”) and perception-level (“P”) for MM-IFEval, prompt-level accuracy (“Prompt.”) and Inst-level accuracy (“Inst.”) for IFEval, and the averaged results across all three benchmarks in the rightmost column.

Model	Parameter	MM-IFEval ^M (ours)			MIA ^M	IFEval ^T			Avg.
		C	P	Avg.		Prompt.	Inst.	Avg.	
LLaVA-NeXT-7B [21]	7B	36.8	16.0	31.6	73.2	32.0	43.3	37.7	47.5
LLaVA-OneVision-Qwen2-7B-OV [16]	8B	37.4	24.0	34.0	84.5	43.3	54.8	49.0	55.8
InternVL2-8B [7]	8B	45.2	32.0	41.9	86.2	44.6	57.0	50.8	59.6
InternVL2.5-8B [6]	8B	49.6	36.0	46.2	88.5	52.2	62.4	57.3	64.0
LLaVA-NeXT-Llama3-8B [21]	8B	45.9	21.0	39.7	83.3	45.0	56.4	50.7	57.9
w. MM-IFInstruct-23k	-	59.3	19.0	49.2 +9.5	86.5 +3.2	50.8	61.8	56.3 +5.6	64.0 +6.1
w. MM-IFDPO-23k	-	58.7	21.0	49.3 +9.6	90.0 +6.7	64.5	73.7	69.1 +18.4	69.5 +11.6
Qwen2-VL-7B-Instruct [41]	8B	42.7	40.0	42.0	80.5	42.4	52.5	47.4	56.6
w. MM-IFInstruct-23k	-	57.0	38.0	52.3 +10.3	87.7 +7.2	46.8	58.4	52.6 +5.2	64.2 +7.6
w. MM-IFDPO-23k	-	55.2	43.0	52.2 +10.2	88.1 +7.6	55.2	64.3	59.7 +12.3	66.7 +10.1

Table 2. Main results on VQA benchmarks, including general knowledge (MMMU [49], MMBench [24], MMStar [5], MMT-Bench [47]), document understanding (AI2D [15], OCRBench [25]), Chat (MMVet [48]) and Hallusion (POPE [19]). Fine-tuning models on MM-IFDPO-23k achieve comparable performance across these benchmarks.

Model	General				Document		Chat	Hallusion	Avg.
	MMMU _{val}	MMBench _{dev}	MMStar	MMT-Bench _{val}	AI2D	OCRBench			
LLaVA-NeXT-Llama3-8B [21]	43.7	72.5	43.6	53.1	73.1	55.0	43.3	87.2	58.9
w. MM-IFInstruct-23k	45.8	69.3	44.2	53.3	71.2	55.3	46.3	88.8	59.3
w. MM-IFDPO-23k	44.1	72.1	43.7	53.1	72.3	56.7	43.9	86.8	59.1
Qwen2-VL-7B-Instruct [41]	53.9	81.0	60.8	63.2	82.9	86.7	63.3	86.3	72.3
w. MM-IFInstruct-23k	54.0	79.3	57.1	61.0	81.6	81.8	61.6	89.2	70.7
w. MM-IFDPO-23k	54.0	81.3	58.5	63.7	83.3	86.8	66.1	85.7	72.4

tuning experiments on two representative MLLMs: Qwen2-VL-7B-Instruct [41] and LLaVA-Next-Llama3-8B [21], using our custom datasets MM-IFInstruct-23k for supervised fine-tuning (SFT) and MM-IFDPO-23k for direct preference optimization (DPO). For the SFT phase, we used a batch size of 128 and a learning rate of 1e-5. For the DPO phase, we used a learning rate of 5e-7 with the batch size of 16. We implemented our training pipeline with the help of LLaMA-Factory and evaluation pipeline under VLMEvalkit [10].

5.1. Results about MM-IFInstruct-23k and MM-IFDPO-23k

Consistently Improvements on Instruction Following Benchmarks. As shown in Tab. 1, both MM-IFInstruct-23k and MM-IFDPO-23k significantly enhance the model’s performance in instruction following benchmarks. Fine-tuning LLaVA-Next and Qwen2-VL on MM-IFInstruct-23k yielded significant averaging performance gains of 6.1% and 7.6% points, respectively. Furthermore, applying DPO with MM-IFDPO-23k also led to notable improvements for LLaVA-Next and Qwen2-VL, with average gains of 11.6% and 10.1% points. Such improvements demonstrate the effectiveness of MM-IFEngine in constructing high-quality training data.

Comparable Results on VQA Benchmarks. To show that

fine-tuning on MM-IFInstruct-23k and MM-IFDPO-23k improves instruction following without degrading performance on other VQA tasks, we analyzed model performance on other widely used benchmarks, as detailed in Tab. 2. Results indicate that models fine-tuning with MM-IFInstruct-23k and MM-IFDPO-23k demonstrate comparable performance across these benchmarks.

SFT vs DPO. As evidenced by Tab. 1 and Tab. 2, DPO using MM-IFDPO-23k significantly surpasses SFT on MM-IFInstruct-23k. This is likely due to negative samples of DPO, which are essential for training models to respect constraints, particularly in our data with multiple and diverse constraints. Additionally, the Kullback–Leibler (KL) divergence in DPO preserves the model’s generalization, as demonstrated in Tab. 2.

5.2. Leaderboard of MM-IFEval

We present the performance comparison results of various MLLMs on our MM-IFEval in Tab. 3, including both proprietary MLLMs such as GPT-4o [13] and Claude-3.5 [1] and open-source MLLMs such as LLaVA-Next [21], LLaVA-OneVision [16], InternVL [6, 7], and Qwen2-VL [41].

MM-IFEval is Challenging. Results on Tab. 3 demonstrate that multimodal instruction following is still a challenging and unsolved task for current MLLMs, specifically for the

Table 3. Evaluation of various MLLMs on MM-IFEval. We report the accuracy of easy and difficult problems and the average accuracy across all problems. The C-Level and P-Level refer to the compose-level and perception-level problems, respectively. The best performance in each section is highlighted in **bold**.

Model	Param	C-Level	P-Level	Avg.
<i>Proprietary MLLMs</i>				
Claude-3.5V-Sonnet [1]	-	67.5	44.0	61.7
GPT-4o-mini [13]	-	70.4	40.0	62.8
GPT-4o (20240806) [13]	-	71.5	44.0	64.6
<i>Open-Source MLLMs</i>				
LLaVA-NeXT-7B [21]	7B	36.8	16.0	31.6
LLaVA-OneVision-Qwen2-7b-OV [16]	8B	37.4	24.0	34.0
MiniCPM-V-2.6 [46]	8B	39.2	32.0	37.4
InternVL2-8B [7]	8B	45.2	32.0	41.9
InternVL2-40B [7]	40B	48.0	36.0	45.0
InternVL2.5-8B [6]	8B	49.6	36.0	46.2
InternVL2.5-26B [6]	8B	53.5	32.0	48.1
Qwen2-VL-72B-Instruct [41]	72B	53.4	43.0	50.8
LLaVA-NeXT-Llama3-8B [21]	8B	45.9	21.0	39.7
+ MM-IFDPO-23k	-	58.7	21.0	49.3
Qwen2-VL-7B-Instruct [41]	8B	42.7	40.0	42.0
+ MM-IFDPO-23k	-	55.2	43.0	52.2

perception-level problems. The proprietary models GPT-4o and Claude-3.5V-Sonnet establish top-tier average performance with scores of 64.6 and 61.7, respectively. The leading open-source MLLM, Qwen2-VL-72B merely achieves an overall accuracy of 50.8. We attribute the performance gap between proprietary and open-source models to the scarcity of high-quality open-source training data for instruction following. As a result of our MM-IFDPO-23k, Qwen2-VL-7B fine-tuned via our optimized DPO approach achieves a score of 52.2, demonstrating a 24.3% relative improvement over its baseline (42.0), and even surpasses the larger Qwen2VL-72B model. We hope our MM-IFEval benchmark motivates further exploration into improving MLLM instruction-following.

Benchmark Examples. Please refer to the Appendix for visual examples of MM-IFEval, including images and instructions with constraints for both compose-level and perception-level problems.

5.3. Ablation Studies

Ablation Studies on Different DPO Settings. In Tab. 4, we present an ablation study on various strategies for constructing pairwise preference data for Direct Preference Optimization (DPO). These strategies primarily include: (1) generating rejected responses by randomly removing constraints from the instruction (second to fourth rows), and (2) prompting MLLMs without providing image inputs to generate rejected responses (bottom row).

We conduct experiments on both the Qwen2-VL-7B-Instruct and LLaVA-NeXT-Llama3-8B models. As shown in Tab. 4, all DPO variants exhibit strong robustness, consis-

Table 4. Ablation studies across different DPO settings, including randomly deleting constraints (second row to fourth row) or prompting MLLMs without images (bottom row) to generate negative responses. Avg. refers to the average score of three IF benchmarks.

Model	MM-IFEval	MIA	IFEval	Avg.
Qwen2-VL-7B-Instruct	42.0	80.5	47.4	56.6
+ DPO (-33% cons)	51.5	88.2	57.9	65.8
+ DPO (-66% cons)	51.2	88.0	58.4	65.9
+ DPO (-100% cons)	52.2	88.1	59.7	66.7
+ DPO (w/o img)	48.4	86.9	54.7	63.4
LLaVA-NeXT-Llama3-8B	39.7	83.3	50.7	57.9
+ DPO (-33% cons)	50.4	87.2	64.3	67.3
+ DPO (-66% cons)	48.7	86.8	69.7	68.4
+ DPO (-100% cons)	49.3	90.0	69.1	69.5
+ DPO (w/o img)	44.7	85.9	64.8	65.2

tently outperforming the baseline. Among the four evaluated strategies, removing 100% of the constraints to generate rejected responses achieves the best performance, whereas omitting image inputs yields the weakest performance. Furthermore, we observe a consistent trend: as the proportion of removed constraints increases from 33% to 100%, the performance of the resulting DPO models improves accordingly. This suggests that removing more constraints amplifies the semantic gap between preferred and rejected responses, thereby enhancing the effectiveness of contrastive learning during DPO training.

Based on these findings, we adopt the 100%-constraint removal strategy as the default approach for constructing the DPO data in MM-IFDPO-23k.

6. Conclusion

This paper contributes to the field of multimodal instruction-following by exploring pipelines for training data collection and proposing a challenging benchmark. We present MM-IFEngine, a pipeline designed to generate image-instruction pairs, subsequently used to construct MM-IFIInstruct-23k for SFT and MM-IFDPO-23k for DPO. We also analyze the limitations of existing multimodal instruction following benchmarks and propose MM-IFEval, a benchmark featuring diverse instruction types and a hybrid evaluation strategy that combines rule-based methods with an LLM-based judge. We hope this work inspires further research into improving the instruction-following ability of Multimodal Large Language Models, a critical step towards realizing their potential in diverse and impactful applications.

References

- [1] Anthropic. Claude 3.5 sonnet. 2024. [7](#), [8](#)
- [2] Yonatan Bitton, Hritik Bansal, Jack Hessel, Rulin Shao, Wanrong Zhu, Anas Awadalla, Josh Gardner, Rohan Taori, and Ludwig Schmidt. VisIT-Bench: A benchmark for vision-

- language instruction following inspired by real-world use. In *NeurIPS, Datasets and Benchmarks*, 2023. 1, 2, 3
- [3] Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4v-synthesized data for lite vision-language models. *arXiv preprint arXiv:2402.11684*, 2024. 3, 4, 2
 - [4] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions, 2023. 3
 - [5] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? In *NeurIPS*, 2024. 2, 6, 7
 - [6] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jingguo Zhu, Shenglong Ye, Hao Tian, ZhaoYang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 7, 8
 - [7] 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. 7, 8
 - [8] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 3
 - [9] Biplob Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pages 845–854, 2017. 2
 - [10] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 11198–11201, 2024. 7
 - [11] Xinyu Fang, Zhijian Chen, Kai Lan, Shengyuan Ding, Yingji Liang, Xiangyu Zhao, Farong Wen, Zicheng Zhang, Guofeng Zhang, Haodong Duan, et al. Creation-mmbench: Assessing context-aware creative intelligence in mllm. *arXiv preprint arXiv:2503.14478*, 2025. 3
 - [12] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. G-l lava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*, 2023. 5, 2
 - [13] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 7, 8
 - [14] Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. Followbench: A multi-level fine-grained constraints following benchmark for large language models. In *ACL*, 2024. 1, 2
 - [15] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016. 2, 6, 7
 - [16] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 7, 8
 - [17] Huayang Li, Siheng Li, Deng Cai, Longyue Wang, Lemao Liu, Taro Watanabe, Yujiu Yang, and Shuming Shi. TextBind: Multi-turn interleaved multimodal instruction-following in the wild. In *ACL Findings*, 2024. 1
 - [18] Jian Li, Weiheng Lu, Hao Fei, Meng Luo, Ming Dai, Min Xia, Yizhang Jin, Zhenye Gan, Ding Qi, Chaoyou Fu, et al. A survey on benchmarks of multimodal large language models. *arXiv preprint arXiv:2408.08632*, 2024. 2
 - [19] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models, 2023. 6, 7
 - [20] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 3
 - [21] 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. 7, 8
 - [22] Junpeng Liu, Tianyue Ou, Yifan Song, Yuxiao Qu, Wai Lam, Chenyan Xiong, Wenhui Chen, Graham Neubig, and Xiang Yue. Harnessing webpage uis for text-rich visual understanding, 2024. 2
 - [23] Yangzhou Liu, Yue Cao, Zhangwei Gao, Weiyun Wang, Zhe Chen, Wenhai Wang, Hao Tian, Lewei Lu, Xizhou Zhu, Tong Lu, Yu Qiao, and Jifeng Dai. Mminstruct: a high-quality multi-modal instruction tuning dataset with extensive diversity. *Science China Information Sciences*, 67(12), 2024. 3
 - [24] 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 *ECCV*, 2024. 2, 6, 7
 - [25] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. OCRBench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 2024. 2, 5, 6, 7
 - [26] Ziyu Liu, Tao Chu, Yuhang Zang, Xilin Wei, Xiaoyi Dong, Pan Zhang, Zijian Liang, Yuanjun Xiong, Yu Qiao, Dahua Lin, et al. MMDU: A multi-turn multi-image dialog understanding benchmark and instruction-tuning dataset for lvlms. In *NeurIPS Datasets and Benchmarks Track*, 2024. 2, 3
 - [27] Renze Lou, Kai Zhang, and Wenpeng Yin. A comprehensive survey on instruction following. *arXiv preprint arXiv:2303.10475*, 2023. 1
 - [28] Yujie Lu, Dongfu Jiang, Wenhui Chen, William Yang Wang, Yejin Choi, and Bill Yuchen Lin. Wildvision: Evaluating vision-language models in the wild with human preferences. *arXiv preprint arXiv:2406.11069*, 2024. 3

- [29] Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Dixin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. *arXiv preprint arXiv:2306.08568*, 2023. 2
- [30] Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. MMLongBench-Doc: Benchmarking long-context document understanding with visualizations. In *NeurIPS Datasets and Benchmarks Track*, 2024. 2
- [31] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 5, 2
- [32] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. Accessed: 2025-02-23. 3
- [33] OpenAI. GPT-4V(ision) System Card. 2023. Accessed: 2025-02-23. 3
- [34] Yusu Qian, Hanrong Ye, Jean-Philippe Fauconnier, Peter Grasch, Yinfei Yang, and Zhe Gan. MIA-Bench: Towards better instruction following evaluation of multimodal llms. In *ICLR*, 2025. 1, 2, 3, 5, 6, 7
- [35] Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. InFoBench: Evaluating instruction following ability in large language models. *arXiv preprint arXiv:2401.03601*, 2024. 1, 2
- [36] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023. 2, 5
- [37] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 3, 4, 2
- [38] Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, et al. Hi Robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*, 2025. 1
- [39] Dingjie Song, Shunian Chen, Guiming Hardy Chen, Fei Yu, Xiang Wan, and Benyou Wang. Milebench: Benchmarking mllms in long context, 2024. 2
- [40] Fei Wang, Xingyu Fu, James Y. Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, Tianyi Lorena Yan, Wenjie Jacky Mo, Hsiang-Hui Liu, Pan Lu, Chunyuan Li, Chaowei Xiao, Kai-Wei Chang, Dan Roth, Sheng Zhang, Hoifung Poon, and Muhan Chen. Muirbench: A comprehensive benchmark for robust multi-image understanding, 2024. 2
- [41] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 7, 8
- [42] Weiyun Wang, Zhe Chen, Wenhui Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, et al. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*, 2024. 5
- [43] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Dixin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023. 1, 2
- [44] Zhiyang Xu, Ying Shen, and Lifu Huang. Multiinstruct: Improving multi-modal zero-shot learning via instruction tuning, 2023. 3
- [45] Zhiyang Xu, Chao Feng, Rulin Shao, Trevor Ashby, Ying Shen, Di Jin, Yu Cheng, Qifan Wang, and Lifu Huang. Vision-flan: Scaling human-labeled tasks in visual instruction tuning, 2024. 3
- [46] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. MiniCPM-V: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 8
- [47] Kaining Ying, Fanqing Meng, Jin Wang, Zhiqian Li, Han Lin, Yue Yang, Hao Zhang, Wenbo Zhang, Yuqi Lin, Shuo Liu, Jiayi Lei, Quanfeng Lu, Runjian Chen, Peng Xu, Renrui Zhang, Haozhe Zhang, Peng Gao, Yali Wang, Yu Qiao, Ping Luo, Kaipeng Zhang, and Wenqi Shao. Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi, 2024. 2, 6, 7
- [48] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. MM-Vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023. 6, 7
- [49] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoxi 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 *CVPR*, 2024. 2, 6, 7
- [50] Yuhang Zang, Wei Li, Jun Han, Kaiyang Zhou, and Chen Change Loy. Contextual object detection with multimodal large language models. *IJCV*, 2025. 2
- [51] Tao Zhang, Yanjun Shen, Wenjing Luo, Yan Zhang, Hao Liang, Fan Yang, Mingan Lin, Yujing Qiao, Weipeng Chen, Bin Cui, et al. CFBench: A comprehensive constraints-following benchmark for llms. *arXiv preprint arXiv:2408.01122*, 2024. 1, 2
- [52] Xinghua Zhang, Haiyang Yu, Cheng Fu, Fei Huang, and Yongbin Li. Iopo: Empowering llms with complex instruction following via input-output preference optimization, 2024. 2
- [53] Xiangyu Zhao, Shengyuan Ding, Zicheng Zhang, Haian Huang, Maosong Cao, Weiyun Wang, Jiaqi Wang, Xinyu Fang, Wenhui Wang, Guangtao Zhai, Haodong Duan, Hua Yang, and Kai Chen. Omnialign-v: Towards enhanced alignment of mllms with human preference, 2025. 3
- [54] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. In *NeurIPS Datasets and Benchmarks Track*, 2023. 2

- [55] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. [1](#), [2](#), [6](#), [7](#)
- [56] Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. Controlled text generation with natural language instructions. In *ICML*, 2023. [1](#), [2](#)

MM-IFEngine: Towards Multimodal Instruction Following

Supplementary Material

A. MM-IFEval

A.1. An overview of Constraints and Instructions

A.1.1. Constraints

Based on daily use cases and existing research, we have identified six main categories of constraints, which can be further divided into 32 specific constraint types shown in Fig. 5. In this section, we introduce and exemplify these six major constraint categories. For detailed descriptions and examples of all 32 subcategories, please refer to Table 5.

Text Length Requirements. In this category, we focus on the length of the response, including the number of paragraphs, sentences, and words. We also consider the length of the response in the aspect of poetry or “Use yes or no to answer the question”. It must be noted that we do not require the model to follow the strict requirement in exact numbers like “*The response must be exactly 56 words*”. The constraints we propose in this category are based on reality, with precise numerical requirements only at the sentence or paragraph level, and of moderate size; the rest of the constraints are used to limit by ranges like “*The response must be between 100 and 150 words*”, which aligns with the task that people tend to encounter in real-world scenarios.

Mathematical Requirements. This category includes constraints related to the most common part of answering mathematical problems like precision, scientific notation, and other mathematical requirements. For example, “*Keep two decimal places for the number in the answer*”, “*Please round up all the numbers in the answer*”, or “*Don’t include specific numbers in your answers. Compare numbers with their relative sizes*”.

Language & Formatting Requirements. This category includes constraints related to the language and formatting of the response, such as answering in a specific language, using a specific format like JSON, or using a specific style like poetry. Requirements for tense, writing style, numbering, list, and other language-related or formatting-related aspects are also included in this category.

Rhetoric & Logic Requirements. “Rhetoric” refers to the art of using language to persuade or influence, while “Logic” refers to the principles of reasoning and argumentation. This category includes constraints related to the rhetoric and logic of the response, such as the use of metaphor, simile, cause-and-effect relationship, conditional statement, and other rhetoric and logic-related aspects.

Action Requirements. “Action” refers to the action that the model should take like a human. We define this category as the constraints that require the model to perform a specific

action, such as tone, role imitation, use specific prefix or suffix, or acting like under some specific situation. We hope this category can help us to evaluate the ability of the model to follow instructions and perform actions in more complex and realistic scenarios.

Keyword Requirements. “Keyword” refers to the specific words or phrases that the model should include or avoid in the response. This category includes constraints related to the response keyword, such as the use of specific keywords, the avoidance of specific keywords, or the variation of specific keywords. For example, “Use at least three synonyms for ‘innovation,’ such as ‘breakthrough,’ ‘new approach,’ or ‘invention,’ spread throughout your text.”

A.1.2. Instruction Tasks

For source datasets lacking original task instructions, we constructed a diverse task pool containing 18 instructions that encourage open-ended responses from models. These instructions can be categorized into five task types: Descriptive Analysis, Emotional & Perspective, Creative Writing, Social Media & Content, and Roleplay. The classification information and examples of the instructions are shown in Table 6.

A.2. Perception-level Problems

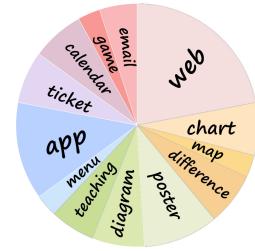


Figure 6. **Image Source Distribution in perception-level problems.** Perception-level problems in MM-IFEval presents a systematic categorization of 100 challenging vision-based instruction-following tasks, organized into 13 distinct classes according to image content characteristics and task complexity.

Perception-level problems in MM-IFEval comprise 100 carefully crafted questions with strong image-constraint correlations. The images can be categorized into 13 information-rich and complex domains shown in Figure 6. Figures 11, 12, 13, and 14 present representative examples from the web interface, diagram, poster, and visual difference categories, respectively, demonstrating the diverse visual challenges incorporated in our benchmark.

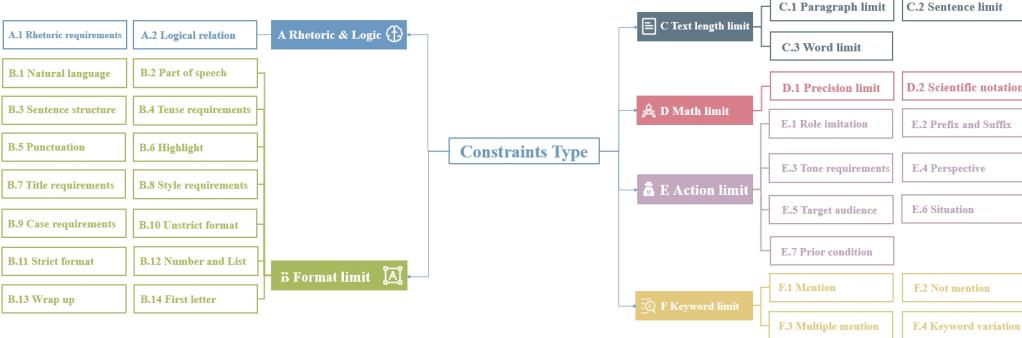


Figure 5. **Demonstration of constraints categories.** We designed 6 main categories for all the constraints used, with a total of 32 subcategories

B. Image Sources

The quality of the image source is crucial for the performance of the model. Except of this, the diversity of the image source is also important to fully utilize or evaluate the ability of the model. We use the following image source:

- **Natural Scene:** The natural scene is the most common image source, which is most used in the real-world like the image of a beautiful landscape, a busy street, or a crowded cafe. In this part, we sample images from CC3M[37] and ALLaVA[3].
- **UI Interface:** The UI interface is the image from the UI interface of the website and mobile application. It is crucial because it represents a significant portion of real-world multimodal interactions where users need to understand and interact with digital interfaces. We collected diverse mobile app UI images from the RICO[9] dataset and web UI images from the MultiUI[22] dataset.
- **Diagram & Chart:** The diagram and chart are the image that contains some specific information like the data, the relationship between the data, or the change of the data. We collect diagram and chart images from ChartQA[31] dataset, which contains diverse diagram and chart images.
- **Mathematic:** The math problem is the image that contains a math problem, which is a common task in the real-world like the problem of the math, the solution of the math problem, or the calculation of the math problem. We collect math problem images from Geo170k[12] dataset, which contains diverse geometry problem images.

C. MM-IFEngine Prompt Template

MM-IFEngine provides a scalable pipeline for mass-producing instruction-following datasets for multimodal large language models, functioning effectively regardless of whether source datasets contain original instructions. This engine enables systematic augmentation of existing visual datasets with diverse instruction-following tasks. Figures 15 and 16 demonstrate representative prompt templates from

MM-IFEngine’s two core components: the instruction generation module and the constraint integration module, respectively, illustrating the methodology behind our automated data construction process.

D. MM-IFInstruct and MM-IFDPO Dataset

Our MM-IFInstruct dataset integrates three distinct data sources: CC3M (without original instructions), ALLaVA (with pre-existing questions), and a diversity collection composed of MultiUI, ChartQA, and Geo170k. To create the MM-IFDPO dataset for preference optimization, we randomly removed 33% of constraints from the MM-IFInstruct samples to generate rejected examples. Figures 17, 18, and 19 illustrate representative samples derived from CC3M, ALLaVA, and our diversity collection, respectively, while Figure 20 demonstrates an example pair from the MM-IFDPO dataset showing both preferred and rejected instructions.

E. Evaluation

E.1. Rule-based

We identified 10 constraint subcategories from our taxonomy of 32 that could be algorithmically verified. For these selected constraints, we developed specialized verification functions with targeted parameters. For efficiency, we employed large language models to analyze each constraint specification, select the most appropriate verification function, and extract the necessary parameters. All selections were subsequently validated through manual review to ensure the accuracy and quality of both the function selection and their parameters. The prompt template used for function selection and parameter extraction is illustrated in Figure 21, while Table 7 provides a comprehensive overview of all verification functions with their corresponding parameter examples.

E.2. Compare Judge Method

Recent works[11, 28] have shown that GPT-4o has the ability to compare two responses from models. For constraint types lacking objective evaluation metrics (such as tone requirements or role imitation), we implemented a comparative assessment method. This approach requires the model under evaluation to generate two responses: one adhering to the target constraint and another without the constraint. A judge model then analyzes both outputs to determine whether significant differences exist between them, thereby more accurately assessing whether the model has successfully followed these subjective constraints. Figure 22 illustrates the prompt used in this comparative evaluation process.

E.3. Direct Judge Method

The Direct Judge method provides the constraint and answer of the model under test directly to the Judge model, and its prompt template is shown in Figure 23.



Instruction

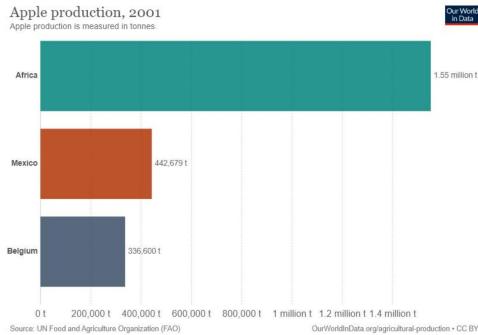
What might have led to the dog's behavior as depicted in this image?



Constraints

1. target_audience_requirement: Your audience is a dog lover.
2. tense_requirements: Use present tense in the first paragraph and past tense in the second.
3. tone_requirement: Adopt a reassuring, empathetic tone as if consoling someone.
4. paragraph_number_limit: Your response must consist of exactly 3 paragraphs.
5. mention: Mention the term 'sorry' at least twice throughout your description.
6. highlight_requirements: Use **bold** for the first occurrence of the term 'aggressive behavior' in each paragraph.
7. wrap_up_requirement: Provide a final paragraph summarizing the key arguments.
8. perspective_requirement: Please answer the question in the second person.

Figure 7. A compose-level problem example from the MM-IFEval benchmark in the general image category.



Instruction

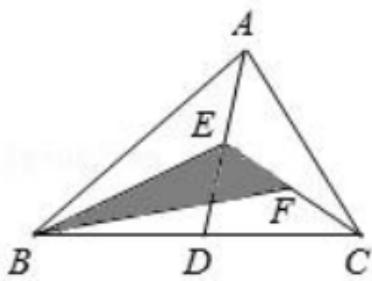
Which region has the highest value of apple production? Give the answer, and analyze the reasons for the large yield of apples in this area.



Constraints

1. precision: In the answer, plot the output in the same unit.
2. title_requirements: Provide a concise title that summarizes the main idea.
3. perspective_requirement: Give your answer from the perspective of a Mexican agricultural expert.
4. sentence_number_limit: Each paragraph should contain between 3 and 5 sentences.
5. unstrict_formatting_requirements: Number the reasons for your analysis.

Figure 8. A compose-level problem example from the MM-IFEval benchmark in the chart image category.



Instruction

In triangle ABC, D is the midpoint of BC, E is the midpoint of AD, and F is the midpoint of CE. Given that the area of triangle ABC is 28 square centimeters, consider the impact of these midpoints on the subdivisions of the triangle. Analyze how these midpoints affect the areas of triangles within triangle ABC and provide a detailed explanation to find the area of the shaded region that is formed within triangle BEC and triangle AEC. Finally, deduce and conclude which part of the interior triangles contribute to the shaded area.



Constraints

- 1.target_audience_requirement: Write your answer for a liberal arts student. You're tutoring her in math.
- 2.word_count_range_limit: Please write between 150 and 200 words in total.
- 3.paragraph_number_limit: Your response must consist of exactly 4 paragraphs.
- 4.sentence_number_limit: Each paragraph should contain between 3 and 5 sentences.
- 5.not_mention: Please do not mention the words 'formula' or 'equation' in your answer.
- 6.mention: Mention the word 'midpoint' at least three times throughout your description.
- 7.tone_requirement: Write your answer in a positive and encouraging tone, emphasizing the simplicity of the geometric concepts involved.

Figure 9. A compose-level problem example from the MM-IFEval benchmark in the geometry image category.

MISTER RON'S BASEMENT

Mister Ron's Basement II
Exploring the Universe of Public Domain Humor, featuring the greatest writers in the world, seven days a week!
Indexed at http://ironenvy.com/Mister_Rons_Full_Catalog.html

Categories: podcasts general
Syndication: RSS
Archives: Keyword Search
Search: [Search]

June 2004
S M T W T F S
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29

2011
October
September
August
July
June
May
April
March
February
January

2010
December
November
October
September

Instruction

Please examine the image and provide a detailed answer to the question: What is the purpose of the textbox?

Constraints

- 1. **word_count_range_limit:** Please write between 80 and 100 words in total.
- 2. **numbering_and_list_requirements:** Present all key points as a main numbered list, with each item followed by a brief bulleted sub-list.
- 3. **mention:** Your answer must mention what the user of the page can do with the text box.
- 4. **not_mention:** Please do not mention the words 'design' or 'aesthetic' in your answer.
- 5. **wrap_up_requirement:** Provide a final paragraph summarizing the key arguments.
- 6. **sentence_number_limit:** Each paragraph should contain between 3 and 5 sentences.

Figure 10. A compose-level problem example from the MM-IFEval benchmark in the website image category.

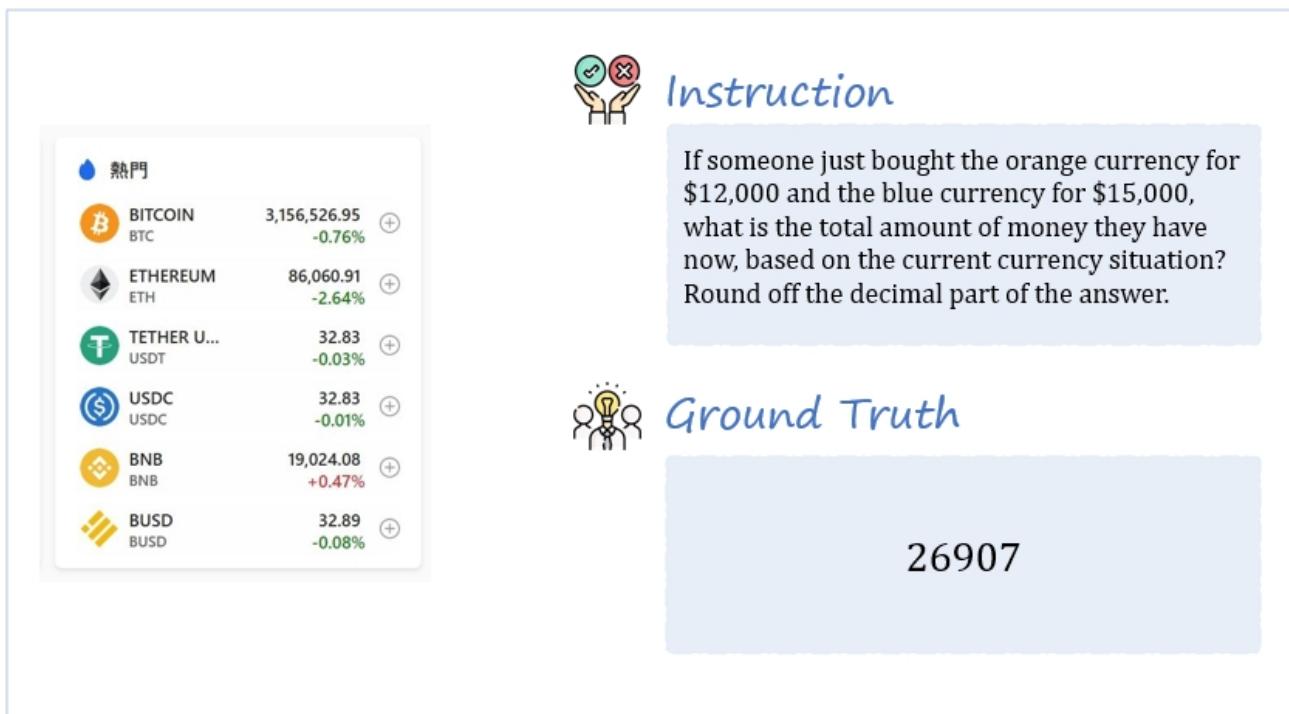


Figure 11. A perception-level problem example from the MM-IFEval benchmark in the web category.

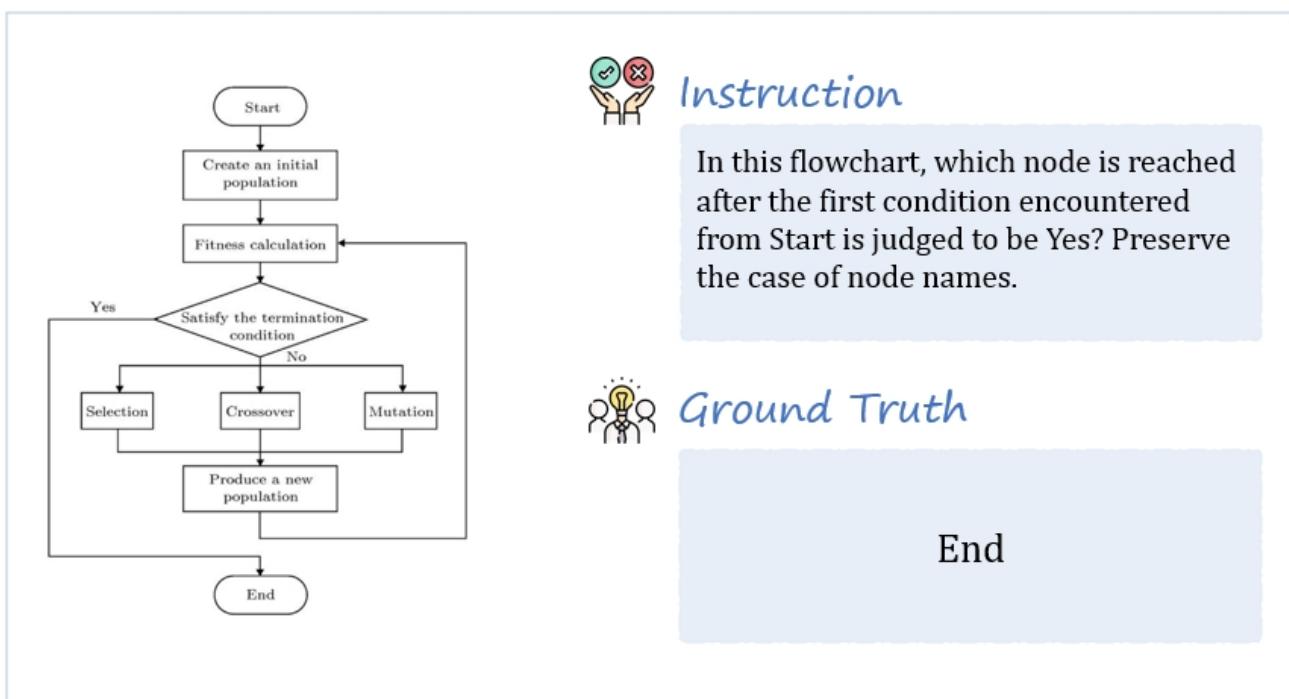
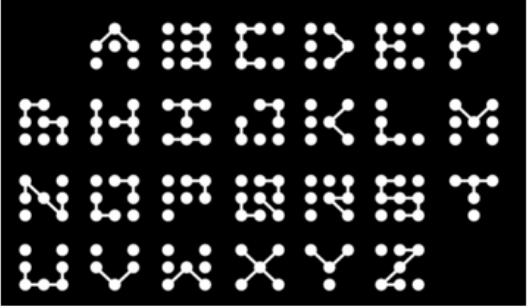


Figure 12. A perception-level problem example from the MM-IFEval benchmark in the diagram category.


Instruction



Observe the alphabet represented by white dots and line segments in the figure. Starting from 'A', what is the second letter composed of eight white dots? Output this letter in uppercase.


Ground Truth

G

Figure 13. A perception-level problem example from the MM-IFEval benchmark in the poster category.


Instruction



Sam and Tom used the red box and Tom used the blue box. They each gave three answers. Would you please judge which of the two boys found more differences? Print the name of the winning boy directly.


Ground Truth

Tom

Figure 14. A perception-level problem example from the MM-IFEval benchmark in the finding difference category.

Instruction generation prompt

You are an expert in generating concise instructions for images.

Task

Given the image, generate a list of appropriate instructions for it.

Your instructions should not be too long or overly detailed, and they should not include any specific details about the image.

On one hand, you can choose appropriate instructions cases for the provided image from the Examples and modify them naturally for the image.

On the other hand, you can generate new instructions, but only if these new instructions are relevant and appropriate for the image.

Examples

{original instructions list}

You output format should be in the following format:

{output format}

Figure 15. Prompt template for image generation instructions using a large language model in MM-IFEngine.

Constraint integration prompt

You are an expert in add appropriate constraints to the instruction for images.

Task

Given the original instruction, your task is to expand the instruction by adding constraints to it.

You can select **as many as possible** appropriate types of constraints from the given **Constraints List** below and modify them. However, ensure that the constraints you generate meet the following requirements:

- 1.Maintain the thematic consistency of the original instruction.
- 2.Be relevant and appropriate for the original instruction and be concise.
3. Do not conflict with the original instruction or with each other.

For example, if the original instruction is a simple query like "What color is the truck?" you should avoid adding constraints such as "Your response must be in three paragraphs," as such a requirement is unnecessary for a short and simple question. Moreover, if the original instruction is a question like "What is the object in the image?", you should avoid adding constraints such as "Respond in the second-person to directly address the reader," as it conflicts with the original instruction.

Original Instruction

{original_instruction}

Constraints List

Your added constraints can be from the following types:

{constraints_list_str}

Output Format Your output should follow the format below::

{output format}

Figure 16. prompt template for integrating constraints in MM-IFEngine.



Instruction

Write a short poem that captures the hustle and bustle of city life depicted in this image.



Constraints

1. Write each stanza so it includes a parenthetical phrase.
2. Each stanza should have 4 lines.
3. Use **bold** for every noun.
4. Start the poem with 'In the heart of the city...' and end with '...the city never sleeps.'
5. Use present tense throughout the poem.
6. Use the third-person perspective to maintain an objective viewpoint.



Answer

In the heart of the city, where the **streets** hum with life,

People rush (beneath the bright blue sky),
Buses pass (**tires** whispering on the road),
Every moment feels like a lively show.

The **signs** stand tall amidst the crowd,
An ambulance speeds by (with urgent sound),
Newspapers rest (in their quiet stack),
As footsteps weave through the urban track.

Buildings rise, their facades grand,
A man walks past (across the land),
Voices blend into ambient noise,
Life unfolds without any poise.

Shoppers carry bags (full of delight),
Sunlight dances (on the pavement's sight),
Each face tells tales of daily grind,
Yet spirits soar, unconfined....the city never sleeps.

Figure 17. A sample constructed by MM-IFEngine pipeline from cc3m dataset



Instruction

What treat can be baked in an oven?



Constraints

1. Write your answer for a general adult audience with limited baking experience, providing clear and simple explanations.
2. Write your answer in a positive and encouraging tone, making the baking process seem accessible and enjoyable.
3. Mention both 'oven' and 'baking' at least once in your response, ensuring they are used naturally and contextually.
4. Begin every sentence with an exclamation mark to convey excitement about baking.
5. Your response must consist of exactly 3 paragraphs, each focusing on a different type of treat that can be baked in an oven.



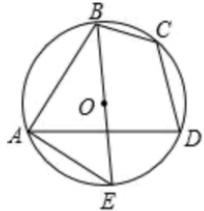
Answer

!One delightful treat you can bake in an oven is cookies! Whether you prefer classic chocolate chip or something more adventurous like oatmeal raisin, baking cookies is a fun and rewarding activity. The aroma of freshly baked cookies wafting through your home will make everyone excited!

! Another scrumptious option for oven baking is muffins! Muffins are perfect for breakfast or as a snack, and you can customize them with various ingredients such as blueberries, bananas, or even chocolate chips. Baking muffins allows you to experiment with flavors while enjoying the satisfaction of creating something delicious from scratch.

! Lastly, consider baking a cake! Cakes come in countless varieties, from simple vanilla to rich chocolate, and can be decorated to suit any occasion. Baking a cake not only provides a sweet treat but also enhances your skills in mixing, measuring, and decorating. So grab your apron and get ready to enjoy the magic of baking in your own kitchen!

Figure 18. A sample constructed by MM-IFEngine pipeline from Allava dataset



Instruction

In the diagram, quadrilateral ABCD is inscribed in circle O with diameter BE and line AE connected to it. Given that the measure of angle BCD is twice the measure of angle BAD, and that angle DAE is equal to angle CAD, develop a comprehensive geometric proof to determine the degree measure of angle DAE.



Constraints

1. Your response must consist of exactly 3 paragraphs, each focusing on a different aspect of the proof: the initial setup, the application of theorems, and the final calculation.
2. Each paragraph should contain between 4 and 6 sentences, ensuring a balanced and detailed explanation.
3. The response must be between 120 and 150 words in total, maintaining clarity and conciseness.
4. Write your answer in a clear and methodical tone, suitable for a high school geometry class.
5. Provide a concise title that summarizes the main idea of the proof, such as 'Geometric Proof for Angle DAE'.



Answer

Geometric Proof for Angle DAE

To begin our proof, we note that since quadrilateral ABCD is inscribed in circle O with BE as its diameter, angle ABE is a right angle (90 degrees) by the Inscribed Angle Theorem. This theorem states that an angle inscribed in a semicircle is always a right angle. Additionally, given that angle BCD is twice angle BAD, let's denote angle BAD as $\langle x \rangle$. Therefore, angle BCD measures $\langle 2x \rangle$.
Next, applying the properties of cyclic quadrilaterals, we know that opposite angles sum up to 180 degrees. Thus, angle BAD plus angle BCD equals 180 degrees. Substituting the values, we get $\langle x + 2x = 180^\circ \rangle$, leading to $\langle 3x = 180^\circ \rangle$ or $\langle x = 60^\circ \rangle$. Since angle DAE is equal to angle CAD, and considering triangle ADE where AD is common, angle DAE also measures 30 degrees because angle CAD is half of angle BAD due to symmetry and the given condition. Finally, calculating the exact measure of angle DAE involves recognizing that angle CAD is half of angle BAD, which was determined to be 60 degrees. Hence, angle CAD is 30 degrees. Consequently, angle DAE, being congruent to angle CAD, also measures 30 degrees. This completes our proof, demonstrating that angle DAE indeed measures 30 degrees based on the provided conditions and geometric principles applied throughout the process.

Figure 19. A sample constructed by MM-IFEngine pipeline from geo170k dataset



 Chosen Score:5/5

prompt

What might the seabird be doing flying alone over the ocean?

Respond in the third-person perspective to maintain an objective viewpoint.
Write the answer in the form of a brief nature observation log entry. Each paragraph should contain between 3 and 5 sentences. Write your answer for a general adult audience with an interest in wildlife and nature. Conclude with a single sentence summarizing the likely behavior of the seabird.

 Rejected Score:3/5

Nature Observation Log Entry
Date: [Insert Date]
Location: [Insert Location]

Observation: A solitary seabird, possibly a seagull or pelican, is seen soaring gracefully above the vast expanse of the ocean. The bird's wings are spread wide, indicating it is likely gliding or flying at a steady pace. The water below appears calm, with gentle ripples on its surface, suggesting a serene environment. The bird's position and posture suggest it may be searching for food, such as fish or crustaceans, that are abundant in the ocean's depths. The absence of other birds or marine life in the immediate vicinity could imply that this bird is either hunting alone or simply enjoying a solitary flight over the sea.

Analysis: Seabirds like seagulls and pelicans are known for their ability to glide and soar on the wind currents, conserving energy while searching for food. The bird's choice of location, over the ocean, indicates its reliance on marine resources for sustenance. The calmness of the water suggests favorable weather conditions, which are essential for successful hunting and gliding. This observation provides insight into the behavior and habitat preferences of seabirds, highlighting their adaptability to oceanic environments and their role in the marine ecosystem.

Figure 20. A DPO training set sample, where the rejected data is obtained by removing 33% of the constraints

Prompt to choose verified function and extract params

Please analyze the following constraint and select the most appropriate function from the given list to verify this constraint. Then extract the required parameters for the verification function from the constraint.

Constraint content: {constraint value}

Available verification functions:

{all candidate validation function names and parameters}

Please complete the analysis following these steps:

****Your task:****

1. Select the most appropriate verification function from the above list (return empty if none is suitable)
2. Extract the required parameters from the constraint based on the function description

****Please return the result in JSON format as follows:****

{output format}

Figure 21. Prompt template for automated verification function selection and parameter extraction

Compare Judge Prompt

You are an expert in judging whether the response follows the given constraint. Your task is to assess whether the model's response satisfies the given constraint and return True or False. I will provide you with the constraint and the model's response under this constraint. To assist with your evaluation, I will also provide you with the model's response to the same question without the constraint.

Constraint: {constraint}

Response under the constraint: {pred_with_constraint}

Response without the constraint: {pred_without_constraint}

Please follow the steps below to evaluate:

Step 1. Compare the model's response under the constraint with its response without the constraint. If you believe these two answers are very similar, it means the model has not fully considered the impact of the constraint on the answer. Please return False.

Step 2. Compare the model's response under the constraint with the content of the constraint. If you believe the model's response does not meet the requirements specified in the constraint, return False. Otherwise, if the response effectively satisfies the constraint, return True.

Response Format: Your answer should only include "True" or "False", and no additional text.

Figure 22. Prompt template for Compare Judge Method

Direct Judge Prompt

Your task is to evaluate whether the response from an AI assistant adheres to all of the given constraints.

Please follow the requirements below to make the judgment:

1. Be strict and consistent in your assessment.
2. You should refer to the content of image to make the judgment.
3. For one constraint, if the response fails to fully meet the constraint, give it a score of 0. Otherwise, give it a score of 1.

<start of response>

{prediction}

<end of response>

<start of constraint list>

{constraints_str}

<end of constraint list>

You should judge and explain for each constraint in the constraint list without omitting any constraint. Finally, list scores of all the constraints in one sentence.

You should strictly follow the format below:

Judgement: ...

Summary: Score of constraint_1: x/1, Score of constraint_2: x/1, Score of constraint_3: x/1, ..., Score of constraint_n: x/1.

Figure 23. Prompt template for Direct Judge Method

Main Class	Subclass	Evaluation	Description	Example
A. Rhetoric & Logic	A.1 Rhetoric requirements	Compare Judge	Constraint that requires the response to use a specific rhetorical technique.	"Your output should include a metaphor."
	A.2 Logical relation	Direct Judge	Constraint that ensures logical cohesion within the response by requiring specific logical connectors or structures.	"Each paragraph must contain at least one cause-and-effect relationship."
B. Format limit	B.1 Natural language	Direct Judge	Constraint specifying which natural language(s) should be used in the response.	"Please answer in Spanish."
	B.2 Part of speech	Direct Judge	Constraint that requires the response to use a specific part of speech.	"Use at least three adjectives in your response."
	B.3 Sentence structure	Direct Judge	Constraint that specifies special sentence structures to be used in the response.	"Write each sentence so it includes a parenthetical phrase."
	B.4 Tense requirements	Direct Judge	Constraint that specifies the use of multiple tenses within the response.	"In past tense totally."
	B.5 Punctuation	Rule-base	Constraint specifying unconventional yet feasible punctuation usage in the response.	"Replace all periods with semicolons."
	B.6 Highlight	Direct Judge	Constraint that specifies a unique but manageable method for highlighting text.	"Use **bold** for every noun."
	B.7 Title requirements	Direct Judge	Constraint that specifies how titles should be added to the response.	"Provide a concise title that summarizes the main idea."
	B.8 Style requirements	Compare Judge	Constraint that specifies an unconventional or distinctive writing style for the response.	"Write the answer in the form of a brief detective story."
	B.9 Case requirements	Direct Judge	Constraint specifying an unusual yet readable approach to letter case in the response.	"Write all nouns in UPPERCASE and all adjectives in lowercase."
	B.10 Unstrict format	Direct Judge	Constraint specifying a unique format for the output while keeping it approachable.	"Format your response as a short play script with speaker labels."
	B.11 Strict format	Direct Judge	Constraint that requires the response to follow a strictly defined format.	"Please provide the output as well-formed XML with custom tags."
	B.12 Number and List	Direct Judge	Constraint for using numbered or bulleted lists in the response.	"Present all key points as a numbered list with bulleted sub-lists."
	B.13 Wrap up	Direct Judge	Constraint that requires a concise, well-structured summary or conclusion.	"Provide a final paragraph summarizing the key arguments."
	B.14 First letter	Direct Judge	Constraint specifying a pattern for the first letters of sentences or paragraphs.	"Each sentence should begin with a letter that progresses through the alphabet."
C. Text Length limit	C.1 Paragraph limit	Rule-base	Constraint that specifies the number of paragraphs in the response.	"Your response must consist of exactly 4 paragraphs."
	C.2 Sentence limit	Rule-base	Constraint that specifies the number of sentences in each paragraph.	"Totally use 5 sentences in your response."
	C.3 Word limit	Rule-base	Constraint that specifies a small range for the total number of words in the text.	"Your response must be a single word or phrase."
D. Math limit	D.1 Precision	Rule-base	Constraint that specifies the level of precision required in mathematical calculations.	"Keep two decimal places for all numbers in the answer."
	D.2 Scientific notation	Rule-base	Constraint that requires the use of scientific notation for large or small numbers.	"Express all numbers greater than 1,000 in scientific notation."
E. Action limit	E.1 Role imitation	Compare Judge	Constraint requiring the response to imitate the tone and style of a specific role or public figure.	"Please answer in the style of a sports commentator."
	E.2 Prefix and Suffix	Rule-base	Constraint that requires the response to begin or end with a specific phrase or symbol.	"Please start your answer with 'Once upon a time...'"
	E.3 Tone requirement	Compare Judge	Constraint specifying an emotional tone for the response.	"Write your answer in a positive and encouraging tone."
	E.4 Perspective	Direct Judge	Constraint that specifies a narrative perspective for the response.	"Write your answer in the first-person singular as a personal account."
	E.5 Target audience	Compare Judge	Constraint requiring the response to be tailored for a specific audience.	"Craft your response as if explaining to high school students."
	E.6 Situation	Compare Judge	Constraint requiring the response to be set in a specific situation or scenario.	"Answer as if you are giving safety instructions before a flight."
	E.7 Prior condition	Direct Judge	Constraint stating that when a specific condition is met, the response must follow a particular process.	"If the user requests legal advice, begin with a disclaimer."
F. Keyword	F.1 Mention	Rule-base & Direct Judge	Constraint that requires including a specific keyword a certain number of times.	"Mention 'GreenTech' exactly three times throughout."
	F.2 Not mention	Rule-base & Direct Judge	Constraint that requires avoiding specific keywords or phrases.	"Do not mention the words 'budget' or 'investment'."
	F.3 Multiple mention	Rule-base & Direct Judge	Constraint requiring including multiple specified keywords in a balanced manner.	"Mention both 'sustainability' and 'renewable energy' at least twice."
	F.4 Keyword variation	Direct Judge	Constraint requiring the use of synonyms or variations of a given keyword.	"Use at least three synonyms for 'innovation' throughout your text."

Table 5. Constraint Categories and Evaluation Methods for MM-IFEval

Category	Instruction
Descriptive Analysis	Describe the animal's typical habitat, diet, and one unique behavioral trait.
	Provide a detailed analysis of the image, including the setting, characters, and notable objects.
	Explain the activity taking place in the image.
	Describe the activities of the person on the left in the image.
Emotional & Perspective	What emotions do you think the person in this image might be feeling?
	Imagine you are the person on the left in the scene depicted in this image, write a story about what you would do next.
	Personify the sign in the image and express its feelings about the rule it presents.
Creative Writing	Create a short conversation between any two individuals in the scene.
	Pretend this snapshot belongs to a larger story. Write a quick paragraph setting up the next plot twist.
	Use this picture as your muse. Craft a brief poem—any style—that captures the emotion you sense.
	Turn this scene into a short children's story focusing on wonder and curiosity.
Social Media & Content	Write a short poem with two stanzas, inspired by the emotion or content depicted in this image.
	Assume this is an image you are about to post on Twitter. Please provide a short, upbeat caption describing it.
	Assume you are creating a Pinterest pin with this image. Write a short inspirational or motivational caption to accompany it.
Role Play	If this image were promoting an upcoming event, compose a quick announcement with the date, a highlight of what to expect, and a call-to-action.
	Imagine you are the photographer who took this picture. Briefly explain why you chose to capture this particular moment and what story you hope it conveys.

Table 6. Task Pool for MM-IFEngine

Verified Function Name	Function Parameters	Constraint Example	Parameter Example
check_whether_response_paragraph_number_in_range	lower_bound:int, upper_bound:int	The number of text paragraphs be at least 3	[3, 10000]
check_whether_response_sentence_number_in_range	lower_bound:int, upper_bound:int	The number of sentences be exactly 3	[3, 3]
check_whether_each_paragraph_sentence_number_in_range	lower_bound:int, upper_bound:int	The number of sentences in each paragraph be less than 3	[0, 2]
check_whether_each_paragraph_sentence_number_in_range_list	ranges>List[tuple]	The number of sentences in the first paragraph be exactly 3, and in the second paragraph be at most 2	[[3, 3), (1, 2)]]
check_whether_each_paragraph_sentence_number_exceeds	exceed_num:int, upper_bound:int	Each new paragraph should have 1 sentence more than the previous one, no paragraph exceeds 7 sentences	[1, 7]
check_whether_response_word_count_in_range	lower_bound:int, upper_bound:int	The number of words should be between 50 and 80	[50, 80]
check_whether_each_paragraph_word_count_in_range	lower_bound:int, upper_bound:int	The number of words in each paragraph should be between 50 and 80	[50, 80]
check_whether_each_paragraph_word_count_in_range_list	ranges>List[tuple]	The number of words in the first paragraph be between 20 and 30, in the second between 50 and 80	[[20, 30), (50, 80)]]
check_whether_whole_response_not_contain_certain_substring	substring:str	The response should not contain the word "apple"	["apple"]
check_whether_whole_response_not_contain_certain_substrings	substrings>List[str]	The response should not contain the words "apple" and "banana"	[["apple", "banana"]]
check_whether_each_sentence_begin_with_certain_substring	substring:str	Each sentence should start with exclamation point	["!"]
check_whether_each_sentence_end_with_certain_substring	substring:str	Each sentence should end with "apple"	["apple"]
check_whether_whole_response_begin_with_certain_substring	substring:str	The response should start with "apple"	["apple"]
check_whether_whole_response_end_with_certain_substring	substring:str	The response should end with "apple"	["apple"]
check_whether_keywords_mentioned_in_range	keywords>List[str], lower_bound_times:int, upper_bound_times:int	The response should mention the word "apple" at least 3 times	[["apple"], 3, 10000]
check_number_precision_in_response	precision:int	The numbers in the response should have 2 decimal places	[2]
check_whether_has_no_number_in_response	-	The response should not contain any number	[]
check_scientific_notation_precision_in_response	significant_digits:int	The numbers in the response should have 3 significant digits	[3]

Table 7. Verification Functions for rule-based evaluation method in MM-IFEval