

FIRST THINGS FIRST: TEACHING MLLM AGENTS TO PRIORITIZE MUST-HAVES BEFORE NICE-TO-HAVES

Anonymous authors

Paper under double-blind review

ABSTRACT

Recent progress in multimodal large language models (MLLMs) has fueled significant enthusiasm in their potential to act as autonomous agents for real-world tasks. However, scenarios requiring agents to fulfill users’ complex, structured requirements remain largely underexplored. In this work, we examine reasoning tasks under three distinct requirement scenarios, each defined by the feasible solution set delineated by must-have and nice-to-have requirements: (i) Must-have requirements uniquely determine a unique feasible solution; (ii) Multiple candidate solutions satisfy the must-have requirements and are prioritized via the nice-to-have requirements; and (iii) No candidate solution satisfies the must-have requirements, in which case the agent should abstain from generating a response. We evaluate state-of-the-art MLLMs on 3,649 carefully constructed problems that reflect realistic service scenarios, including e-commerce platforms, booking systems, and map-based or ride-hailing applications. Our evaluation reveals that existing MLLMs exhibit catastrophic failures in all scenarios. Specifically, these models frequently misinterpret task requirements, violate must-have requirements, and produce invalid solutions. To address this critical gap, we propose First Things First Reinforcement Learning (FTF-RL) that explicitly optimizes reasoning over multi-priority user requirements. Experimental results show that our method substantially improves the task success rate compared to strong baselines. Moreover, FTF-RL yields general effectiveness on popular logical and mathematical reasoning tasks, including LogicVista, MathVision, and MathVista. Our finding suggests that enhancing requirement comprehension provides a simple yet effective pathway toward improving the broad generalization of MLLMs. Code and evaluation data are available at *anonymity*.

1 INTRODUCTION

Recent advances in multimodal agents have shown promising results in automating complex tasks such as booking flights, reserving hotels, or navigating graphical user interfaces (GUI) through natural language instructions (Xie et al., 2024a;b; Cheng et al., 2025; Guo et al., 2025). These systems are typically evaluated in settings where users provide unambiguous instructions, the agent simply needs to execute the command accurately (Jiang et al., 2024b; Rawles et al., 2025; Zhang et al., 2025c). While impressive, such scenarios represent an idealized setting: the intent is clear, the requirements are minimal, and there is little ambiguity about what constitutes a correct solution.

In real-world service scenarios, however, user requirements are rarely so simple (Wen et al., 2024; Ji et al., 2025; Liu et al., 2025). Requests are often multifaceted, containing both **must-have** requirements and **nice-to-have** requirements. Current agents often struggle in these settings because they lack a mechanism to prioritize hard requirements over soft requirements, and thus frequently overfit to all expressed conditions, returning either infeasible solutions or incorrect results. For example, when asked to “*book a non-smoking hotel for two people, preferably with breakfast included*”, a typical agent might return a hotel that includes breakfast but fails to ensure the non-smoking requirement, violating the user’s core intent. Conversely, when given conflicting requirements, such as “*book the cheapest hotel with two bedrooms, but also with a sea view if possible*”, agents often fail to resolve the trade-off and either refuse to answer or return an irrelevant option.

108 different sizes and observe consistent improvements on FTF-BENCH across single-answer, multiple-
109 answer, and unanswerable settings. Beyond FTF-BENCH, models fine-tuned with FTF-RL show
110 gains on other reasoning benchmarks, indicating that learning to parse and prioritize requirements
111 transfers to broader reasoning skills. An ablation that removes the requirement-classification term
112 reduces accuracy, especially in multiple-answer and unanswerable cases, confirming the value of
113 explicit supervision over necessity. We call for greater attention to the role of requirement-aware
114 reasoning (Section 5.3.1).

115 116 117 2 RELATED WORK

118 119 120 2.1 MULTIMODAL REASONING

121
122 Research on reasoning in MLLMs has advanced through Multimodal Chain-of-Thought (MCoT)
123 and RL approaches. To inject more procedural structure, early CoT-based approaches (Li et al.,
124 2025) decomposed reasoning into stages such as perceptual summarization, localized grounding,
125 and fine-grained verification. However, their rigid pipeline structure limited adaptability to diverse
126 tasks. Subsequent variants adopted more flexible decomposition of reasoning: for example, Can-
127 tor (Gao et al., 2024) explicitly partitions the model’s processing into perception and decision steps,
128 while TextCoT (Luan et al., 2024) refines reasoning by zooming from global descriptions to lo-
129 cal crop analyses. For specialized reasoning, Fei et al. (2025) proposed Video-of-Thought (VoT),
130 which breaks down video clips into temporally ordered segments to improve action prediction within
131 video-chat benchmarks.

132 Reinforcement learning has emerged to elicit deeper reflection and optimize reasoning quality.
133 RL frames reasoning as a Markov Decision Process (MDP), optimizing trajectories via rewards.
134 DeepSeek-AI et al. (2025) used verifiable rewards (e.g., math correctness), which Pan et al. (2025)
135 extended to MLLMs with MedVLM-R1. For spatial reasoning, Ouyang (2025) used spatial consis-
136 tency rewards to boost 6D reasoning accuracy. Recent RL advances like StepGRPO added interme-
137 diate rewards, enhancing logical consistency on R1-VL (Zhang et al., 2025a). However, most prior
138 RL approaches focus on pushing the upper bound of VLM reasoning performance, without explic-
139 itly ensuring that the model fully understands and decomposes the requirements of the question. To
140 address this gap, we design a requirement-aware reward, resulting in reasoning chains that are both
141 accurate and aligned with the task intent.

142 143 144 2.2 INSTRUCTION FOLLOWING BENCHMARKS IN MLLMS

145
146 A variety of benchmarks have been introduced to evaluate multimodal large language models
147 (MLLMs) across multiple facets. These encompass assessments of general knowledge (Liu et al.,
148 2024; Yue et al., 2024), document understanding (Yue et al., 2024; Fu et al., 2023), perceptual rea-
149 soning (Hu et al., 2020), multi-image comprehension (Yu et al., 2024), and instruction following
150 (IF) (Qian et al., 2025; Bitton et al., 2023).

151 Several benchmarks have been proposed to quantify the instruction-following capability of large
152 language models (Jiang et al., 2024a; Zhang et al., 2025b; Hida et al., 2024). For instance, LIF-
153 Bench (Wu et al., 2025) limits its scope to instructions whose fulfillment can be verified automati-
154 cally, which enhances reproducibility and model differentiation. InfoBench (Qin et al., 2024) de-
155 composes prompts into atomic sub-instructions and computes the Decomposed Requirements Fol-
156 lowing Ratio (DRFR), thereby granting partial credit for each satisfied micro-requirement. Bench-
157 marks like MIA-Bench (Qian et al., 2025) and VisIT-Bench (Bitton et al., 2023) adopt GPT-4
158 for both question generation and evaluation in instruction following settings. However, existing
159 instruction-following benchmarks implicitly treat all directives as equally important. In contrast,
160 our proposed FTF-BENCH introduces a necessity hierarchy to instruction-following tasks, evaluat-
161 ing whether models prioritize mandatory requirements before optional ones rather than treating all
directives as equally critical.

Table 1: Distribution of FTF-BENCH across different task scenarios.

Image Scene	Single-Answer	Multiple-Answer	Unanswerable	Total
E-commerce Platforms	585 (27.9%)	903 (43.0%)	610 (29.1%)	2098
Booking Services	286 (26.2%)	533 (48.8%)	274 (25.1%)	1093
Maps & Ride-Hailing	153 (33.4%)	152 (33.2%)	153 (33.4%)	458

3 INTRODUCING FIRST THINGS FIRST BENCHMARK

3.1 OVERVIEW

Real-world intelligent agents, such as customer-service chatbots and virtual assistants, must process user requests involving multiple requirements, where satisfying must-to-haves before nice-to-haves is critical for success. Existing multimodal benchmarks mainly assess generic instruction following and rarely measure whether models can identify and prioritize must-have requirements before optimizing for requirements. To address this gap, we introduce FTF-BENCH, a comprehensive evaluation suite for assessing MLLMs’ ability to understand and reason over complex, multi-priority user requirements.

FTF-BENCH covers three representative real-world domains: e-commerce platforms, booking services (e.g., hotel, flight, train), and maps & ride-hailing applications (Figure 1). These images span diverse interface layouts and interaction contexts. Tasks in FTF-BENCH are categorized based on the relationship between candidates and mandatory requirements. Single-answer tasks have a unique option that satisfies all mandatory conditions. Multi-answer tasks have multiple valid options, requiring models to consider optional requirements and trade-offs to select the best choice. Unanswerable tasks contain no valid options, testing whether models can detect infeasible requests and respond appropriately. Table 1 details the distribution of samples across these combinations. In total, the dataset comprises 3,649 samples, of which e-commerce platforms contribute 2,098 samples (57.5%), booking services 1,093 samples (30.0%), and maps & ride-hailing 458 samples (12.5%).

3.2 TASK FORMULATION

In this section, we formalize the definition of our benchmark tasks, which are designed to evaluate three representative scenarios *unique-answer*, *multiple-answer*, and *unanswerable*. Each sample is represented as:

$$\mathcal{D} = \{x_i\}_{i=1}^N, \quad x = (I, \mathcal{O}, \mathcal{R}^+, \mathcal{R}^-), \quad (1)$$

where I is the input image, $\mathcal{O} = \{o_1, \dots, o_M\}$ is the set of M candidate objects appearing in I , \mathcal{R}^+ is the set of must-have requirements, and \mathcal{R}^- is the set of nice-to-have requirements. Each requirement $r \in \mathcal{R}^+ \cup \mathcal{R}^-$ is a logical predicate evaluated on a candidate $o \in \mathcal{O}$. The satisfaction of a candidate o with respect to a requirement r is given by:

$$\text{sat}(o, r) \in \{0, 1\}, \quad \text{sat}(o, r) = 1 \Leftrightarrow o \text{ satisfies } r. \quad (2)$$

A candidate satisfies all must-have requirements if

$$\text{sat}^+(o) := \prod_{r \in \mathcal{R}^+} \text{sat}(o, r) = 1, \quad (3)$$

and the subset of candidates that meet all must-have requirements is:

$$\mathcal{S}^+ = \{o \in \mathcal{O} \mid \text{sat}^+(o) = 1\}. \quad (4)$$

The first scenario, single-answer, is defined by $|\mathcal{S}^+| = 1$, meaning exactly one candidate satisfies all must-have requirements.

The second scenario, multiple-answer, corresponds to $|\mathcal{S}^+| \geq 2$. In this case, candidates must be further distinguished using the nice-to-have requirements \mathcal{R}^- . Each optional requirement r_j^- has an associated priority $p(r_j^-) \in \mathbb{N}^+$. Let $\pi(\mathcal{R}^-) = (r_{(1)}^-, r_{(2)}^-, \dots, r_{(L)}^-)$ denote the priority-ordered sequence. Starting from $\mathcal{S}_0 = \mathcal{S}^+$, candidates are iteratively filtered according to $\mathcal{S}_k = \{o \in \mathcal{S}_{k-1} \mid$

216 $\text{sat}(o, r_{(k)}^-) = 1\}$. After applying all optional requirements, the remaining candidate set \mathcal{S}_L is used
 217 to determine the final answer.
 218

219 The third scenario, unanswerable, corresponds to $\mathcal{S}^+ = \emptyset$, meaning no candidate satisfies all must-
 220 have requirements. Any prediction of a candidate option in this case is counted as an error.
 221

222 3.3 BENCHMARK CONSTRUCTION

223 To construct a high-quality dataset for training and evaluation, we develop a large-model-driven
 224 pipeline that systematically generates image-requirement pairs. The pipeline comprises the follow-
 225 ing core stages: Image Collection, Requirement Generation, Colloquial Requirement Expression,
 226 and Human Verification. Several representative examples are presented in Appendix A.
 227

228 **Image Collection.** We first assemble a diverse image corpus to capture realistic service-oriented
 229 scenarios. For each domain, volunteers collect screenshots from multiple applications, sampling
 230 different interface states (e.g. homepages, product pages, shopping carts, search results). This design
 231 ensures broad coverage of everyday usage contexts, and yields a dataset reflecting high diversity and
 232 structural complexity.
 233

234 **Requirement Generation.** For each image, we leverage LLMs to generate both requirements
 235 and ground-truth answers. We design prompt templates to align precisely with three task types
 236 (single-answer, multi-answer, unanswerable), ensuring that each requirement is grounded in the
 237 image content. Specifically, we produce a labeled requirement set that separates must-have (\mathcal{R}^+)
 238 from nice-to-have (\mathcal{R}^-) requirements, and for multi-answer cases we assign an explicit priority
 239 order over \mathcal{R}^- to support ranking among candidates.

240 **Colloquial Requirement Expression.** We then convert structured requirements into natural, con-
 241 versational queries, better emulating real user behavior. In practice, users rarely frame needs in
 242 formal subclauses; rather, they pose a single coherent question that interleaves must-have and nice-
 243 to-have requirements. To simulate this, we prompt the model to rewrite requirements as context-
 244 aware, fluent utterances while preserving the priority hierarchy. This step preserves the explicit
 245 must-have/nice-to-have distinction in a single utterance so that downstream evaluation can still re-
 246 flect the necessity-first objective rather than generic instruction following.
 247

248 **Human Verification.** Finally, we apply human verification to secure benchmark integrity. Four
 249 trained annotators independently inspect each image–query pair, validating requirement correctness,
 250 answer consistency, and the separation between mandatory and optional constraints. Any disagree-
 251 ment is resolved via discussion, producing a high-quality, standardized dataset that supports robust
 252 evaluation. During checking, annotators ensure that each instance conforms to one of the three
 253 settings so that the final benchmark faithfully measures requirement-aware reasoning. We provide
 254 several prominent error types in Appendix D.

255 4 FIRST THINGS FIRST REINFORCEMENT LEARNING

257 4.1 MULTI-OBJECTIVE REWARD FUNCTIONS

259 Reward models play a crucial role in RL, as they directly determine the optimization signal that
 260 guides policy improvement. Recent advances, such as DeepSeek-R1 (DeepSeek-AI, 2025), have
 261 demonstrated that verifiable reward functions can substantially enhance the reasoning ability of
 262 MLLMs. Inspired by this success, we design a rule-based, multi-objective reward function to eval-
 263 uate both requirement understanding and visual reasoning capabilities. This design ensures that the
 264 model not only produces correct final answers but also generates interpretable intermediate steps,
 265 thereby improving generalization. Specifically, our reward framework assesses model outputs along
 266 four key dimensions: format compliance, final-answer correctness and requirement classification
 267 accuracy, which distinguishes between must-have and nice-to-have requirements (Figure 2).

268 **Format Reward.** The format reward R_{format} verifies whether the model output strictly follows
 269 the predefined XML-style schema. Specifically, we check three components: (i) whether the re-
 requirement classification is enclosed within `<requirements>...</requirements>` tags and

270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323

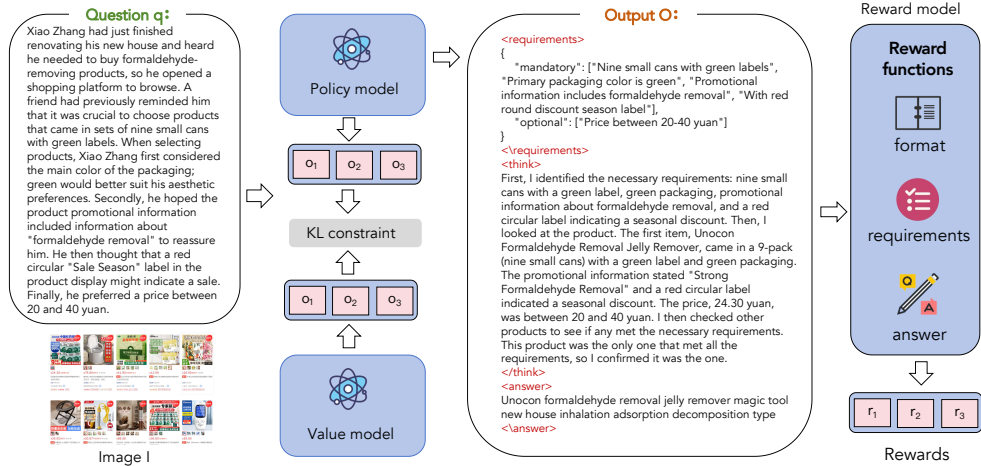


Figure 2: **FTF-RL framework.** Given an input image I and question q , we first sample G candidate outputs $\{o_i\}_{i=1}^G$ from the old policy model. Then we compute a reward r_i for each o_i using our proposed multi-objective reward function (see Section 4.1 for details). Finally, we optimize the current policy model by maximizing A_i , while regularizing the update using KL divergence between π_θ and the reference policy model π_{ref} to keep the updated policy close to the reference policy.

expressed as a valid JSON object separating `must_have` and `nice_to_have` fields; (ii) whether the intermediate reasoning process is included in `<think> ... </think>` tags; and (iii) whether the final answer is provided in `<answer> ... </answer>` tags. The reward is computed as a step-wise function:

$$R_{\text{format}}(y) = \begin{cases} 1, & \text{if all three components are valid and well-formed,} \\ 0, & \text{if none of the components follow the required format.} \end{cases} \quad (5)$$

This reward encourages the model to produce fully structured, machine-parsable, and human-readable outputs, facilitating both automatic evaluation and transparent error analysis.

Accuracy Reward. The accuracy reward R_{answer} measures whether the model’s final prediction matches the ground-truth answer. Since our tasks are formulated as fill-in-the-blank questions, we employ a powerful MLLM as a judging model to robustly compare the model-generated answer against the reference solution. This approach mitigates surface-form mismatch issues (e.g., synonyms, equivalent expressions). Formally, for a given question q , model output y , and ground-truth answer a^* , we define:

$$R_{\text{answer}}(y, a^*) = \begin{cases} 1, & \text{if } \text{Judger}(q, \text{Ans}(y), a^*) = \text{True}, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where $\text{Ans}(y)$ extracts the content between the `<answer> ... </answer>` tags from the model output, and $\text{Judger}(\cdot)$ is the MLLM-based evaluation function that returns `True` if the predicted answer is semantically equivalent to the ground-truth answer a^* .

Requirement Reward. The requirement reward $R_{\text{requirement}}$ evaluates the model’s ability to correctly identify and classify user requirements into `must_have` and `nice_to_have` categories. This component is motivated by the observation that many reasoning failures arise not from incorrect computation, but from misinterpretation of user intent: models often over-constrain by treating nice-to-have requirements as hard requirements, or under-constrain by ignoring must-have requirements, leading to invalid or suboptimal answers. By explicitly rewarding correct requirement classification, we encourage the model to faithfully represent user intent before performing reasoning, thereby reducing downstream reasoning errors and improving decision quality.

Formally, let J_{pred} denote the JSON object produced by the model within the `<requirements>` tags, and J_{gold} the ground-truth classification. We define a normalized F1-based reward:

$$R_{\text{requirement}}(y, J_{\text{gold}}) = \frac{1}{2} (\text{F1}(\mathcal{R}_{\text{pred}}^+, \mathcal{R}_{\text{gold}}^+) + \text{F1}(\mathcal{R}_{\text{pred}}^-, \mathcal{R}_{\text{gold}}^-)), \quad (7)$$

where $\text{F1}(\cdot, \cdot)$ computes the F1-score between the predicted and gold requirement sets for each category. This design provides a soft, differentiable reward signal that penalizes both false positives (e.g., over-constraining) and false negatives (e.g., ignoring essential requirements).

5 EXPERIMENTS

5.1 SETUP

We evaluate seven widely used MLLMs that represent the current frontier on FTF-BENCH. The proprietary group includes Gemini 2.5 Pro (Gemini, 2025), GPT-5 (OpenAI, 2025a), GPT-o3 (OpenAI, 2025b), and Doubao 1.6 Seed (ByteDance Seed, 2025). The open-source group covers Qwen2.5-VL (Bai et al., 2025), LLaMA-4 (Meta AI, 2025) and Claude-Sonnet-4.5 (Anthropic, 2025). For Qwen2.5-VL, we report results for the 7B, 32B, and 72B checkpoints to examine scaling effects. We also report results on four variants of LLaVA (Liu et al., 2023). To study the impact of our method, we apply FTF-RL to Qwen2.5-VL-3B, LLaVA-OneVision-7B and LLaVA-1.5-13B and evaluate the fine-tuned MLLMs alongside their instruction-tuned baselines on FTF-BENCH and additional reasoning suites.

For benchmark evaluation, we conduct experiments on all 3,649 instances of FTF-BENCH as illustrated in Figure 1. Detailed prompts used in both Direct and Upper settings are provided in Appendix B. For reinforcement learning, we train MLLMs using 90% of the benchmark data and reserve the remaining 10% for evaluation. We have verified that this 10% evaluation subset is representative of the full benchmark in Appendix K. All reinforcement learning experiments are conducted under a unified evaluation to achieve fair comparison. We adopt GRPO (Shao et al., 2024) with a KL-penalty coefficient of 10^{-2} to stabilize policy updates. Training is conducted with a global batch size of 128 and a learning rate of 10^{-6} , using AdamW optimization and full-shard FSDP for efficient distributed training. During rollout, we use a temperature of 1.0 and sample $n = 5$ candidate responses to encourage exploration.

5.2 EVALUATION RESULTS ON FTF-BENCH

We conduct a comprehensive evaluation of current MLLMs on FTF-BENCH under two input settings. In the Direct setting, MLLMs read the original colloquial user request and must infer both the requirement hierarchy and the final decision. In the Upper setting, we paraphrase the same request into gold requirement labels that separate must-haves from nice-to-haves and feed these labels to MLLMs as structured guidance, which serves as an empirical upper bound for performance with perfect requirement understanding. We provide the main results in Table 2.

Across most scenarios, Upper exceeds Direct, confirming that the primary source of error is not visual perception alone but the failure to correctly parse and prioritize requirements from natural language. Nearly all MLLMs show a noticeable decline when directly interpreting the original user text. The gap is particularly severe in the multiple-answer and unanswerable scenarios. For multiple-answer tasks, candidate options often contain several plausible distractors, and only by reasoning with nice-to-have conditions can the optimal choice be determined. Without explicit requirement decomposition, MLLMs struggle to resolve these fine-grained trade-offs. For unanswerable tasks, where no candidate satisfies the must-have requirements, MLLMs should reject all options. However, most MLLMs still attempt to produce an answer, revealing a tendency to over-accommodate user prompts even when abstention is the correct strategy. These observations expose a fundamental weakness in requirement-aware reasoning.

Another observation is the large gap between model performance and the upper bound. While proprietary MLLMs maintain relatively high accuracy under both settings, the difference between their direct and upper results remains non-trivial. For open-source MLLMs, the discrepancy is even

Table 2: Evaluation results of current MLLMs on FTF-BENCH across three task types. Upper denotes the performance when MLLMs are given golden requirements, while Direct reports accuracy when MLLMs directly answer from the original prompt without additional guidance.

Models	Single-Answer		Multiple-Answer		Unanswerable		Average	
	Upper	Direct	Upper	Direct	Upper	Direct	Upper	Direct
<i>Proprietary MLLMs</i>								
Gemini-2.5-pro [†]	88.89	86.91	84.20	82.26	81.72	78.55	84.26	81.75
GPT-5 [†]	83.57	85.21	80.73	78.10	82.95	80.24	82.49	80.86
GPT-o3 [†]	82.67	77.78	80.41	80.03	83.31	83.09	82.33	79.68
Doubao-1.6-seed [†]	80.09	78.66	81.70	76.99	83.80	84.05	82.24	80.59
Claude-Sonnet-4.5 [†]	77.52	75.37	73.89	71.94	80.98	82.49	78.01	77.51
<i>Open-Source MLLMs</i>								
LLaMA-4	58.36	55.48	55.93	54.45	44.43	36.94	55.03	51.99
Qwen2.5-VL-7B-Instruct	61.16	22.99	58.25	20.10	47.59	18.64	57.69	21.05
Qwen2.5-VL-32B-Instruct	70.66	69.25	70.55	70.17	53.08	50.58	67.72	66.57
Qwen2.5-VL-72B-Instruct	71.26	35.76	74.75	35.12	83.09	29.62	77.60	34.48
LLaVA-OneVision-7B	45.28	19.15	43.17	18.22	42.36	21.78	43.38	19.73
LLaVA-NEXT-34B	50.31	27.27	50.20	20.21	51.40	25.83	50.76	24.11
LLaVA-NeXT-13B	48.65	26.92	46.89	18.34	46.72	33.95	47.25	26.13
LLaVA-1.5-13B	48.70	26.98	46.94	16.39	42.77	34.00	45.72	25.28

more striking. In particular, Qwen2.5-VL and LLaMA-4 exhibit substantial improvements once gold requirements are provided, suggesting that their reasoning pipeline is hampered less by visual limitations than by misinterpretation of complex user intent.

Interestingly, scaling effects are not monotonic. Within Qwen2.5-VL, the 7B and 72B models fail catastrophically in direct reasoning, whereas the 32B variant demonstrates comparatively strong capability. We interpret this as evidence of two distinct failure modes. The smaller 7B model appears to lack the basic capacity to reliably distinguish must-haves from nice-to-haves, which leads to frequent violations of core requirements. The largest 72B model, while more powerful in general, shows an opposite failure pattern. Its tendency to overfit to the surface form of user prompts results in excessive alignment with every expressed condition, elevating optional requirements to mandatory status and thereby producing infeasible outputs. In contrast, the 32B model achieves a more balanced handling of requirement prioritization, revealing that scale alone does not guarantee progress and that training dynamics strongly influence requirement-aware reasoning. [To further characterize how MLLMs distinguish between must-have and nice-to-have requirements in the Direct setting, we also report requirement-classification confusion matrices for representative MLLMs in Appendix G.](#)

5.3 IMPROVE REQUIREMENT-AWARE REASONING BY FTF-RL

5.3.1 MAIN RESULTS

We further improve requirement-aware reasoning by applying FTF-RL to open-source Qwen2.5-VL. We randomly sample 90% of FTF-BENCH for training and hold out the remaining 10% for evaluation. Comparison results are presented in Table 3.

Both the 7B and 3B variants deliver strong gains, demonstrating that FTF-RL is highly effective at strengthening the requirement-aware reasoning capabilities of MLLMs. The most remarkable results appear in the multiple-answer scenario. Qwen2.5-VL-7B-Instruct improves by more than 26% after reinforcement learning, showing that the method enables MLLMs to reliably separate must-have conditions from nice-to-have preferences and make the correct choice even when several confusing candidates are present. The single-answer and unanswerable settings also benefit, with consistent improvements observed across scales.

Table 3: Impact of reinforcement learning on different MLLMs

Model	Single-Answer	Multiple-Answer	Unanswerable	Average
Qwen2.5-VL-3B-Instruct	48.55	33.13	32.79	38.95
+ FTF-RL	56.52 ↑7.97	41.10 ↑7.97	34.42 ↑1.63	45.85 ↑6.90
Qwen2.5-VL-7B-Instruct	46.38	32.52	44.26	39.78
+ FTF-RL	57.97 ↑11.59	58.90 ↑26.38	42.62↓1.64	55.80 ↑16.02
LLaVA-OneVision-7B	45.82	33.69	42.87	44.09
+ FTF-RL	52.51 ↑6.69	40.15 ↑6.46	48.79 ↑5.92	50.48 ↑6.39
LLaVA-1.5-13B	49.23	37.51	43.19	46.64
+ FTF-RL	57.81 ↑8.58	43.74 ↑6.23	49.05 ↑5.86	52.87 ↑7.23

Table 4: Effect of FTF-RL on model generalization across various reasoning benchmarks.

Model	LogicVista	MathVision	MathVista	InfoQA
Qwen2.5-VL-7B-Instruct	43.40	24.67	60.80	65.35
+ FTF-RL	47.43 ↑4.03	25.65 ↑0.98	61.30 ↑0.05	67.81 ↑2.46
Qwen2.5-VL-3B-Instruct	36.91	23.03	40.70	37.62
+ FTF-RL	40.49 ↑3.58	24.34 ↑1.31	39.50↓1.20	39.93 ↑2.31
LLaVA-1.5-13B	29.23	11.12	27.64	41.57
+ FTF-RL	35.61 ↑6.38	13.74 ↑2.62	29.05 ↑1.41	41.92 ↑0.35

5.3.2 GENERALIZATION ACROSS REASONING BENCHMARKS

To further investigate how requirement-aware reasoning contributes to the general capability of MLLMs, we compare MLLMs trained only on FTF-BENCH with their baselines on other logic and math reasoning benchmarks, including LogicVista (Xiao et al., 2024), MathVision (Wang et al., 2024), and MathVista (Lu et al., 2024). Table 4 presents the results of Qwen2.5-VL models before and after FTF-RL training. We are surprised to find that even without any explicit training on these reasoning benchmarks, the MLLMs exhibit consistent improvements after reinforcement learning on most reasoning benchmarks. This suggests that **requirement-aware reasoning not only strengthens the understanding of complex user intent but also stimulates the general reasoning ability of MLLMs, leading to clear gains across diverse and challenging tasks.** We present two case studies on LogicVista and MathVision in Appendix I.

6 CONCLUSION

In this paper, we investigate requirement-aware reasoning in realistic service settings where LLM-based agents are required to prioritize must-haves before nice-to-haves. We first introduce FTF-BENCH to evaluate the requirement-aware reasoning capability across various scenarios with 3,649 constructed instances. Experiments show that current MLLMs frequently misread requirements, violate hard requirements, and output invalid solutions, leading to catastrophic failures in prioritizing requirements. To improve the requirement-aware reasoning capability, we further present a multi-objective RL framework FTF-RL that rewards correct requirement identification, proper ordering of requirement satisfaction, and answer validity. After training, MLLMs perform substantially better and narrow the gap toward the upper bound obtained with requirement labels. We also observe consistent gains on other tasks that require complicated reasoning, which suggests that strengthening requirement comprehension yields broader generalization. We call for more attention to the pivotal role of requirement-aware reasoning in advancing the reasoning reliability of MLLMs.

ETHICAL CONSIDERATIONS

All authors of this work have read and agree to abide by the ICLR Code of Ethics. Our study focuses on evaluating and enhancing requirement-aware reasoning in MLLMs. We call for the widespread deficiencies of current models in prioritizing must-have over nice-to-have requirements and introduce reinforcement learning methods that substantially improve this capability. All experiments are conducted on data we collected through screenshots of publicly available applications, with requirements automatically generated and subsequently verified by trained human annotators. Annotators were compensated fairly at a rate of 1 RMB per data sample, and no personally identifiable or sensitive information is included in the dataset. We confirm that the benchmark does not introduce bias toward any social group, nor does it involve privacy, security, or harmful use concerns. For paper preparation, we only used LLMs for light editing after completing the writing, limited to correcting typos and grammar errors, and did not rely on LLMs for substantive content generation.

REPRODUCIBILITY STATEMENT

We fully commit to the reproducibility of all results reported in this paper. Section 5.1 introduces our experimental setup and evaluation protocols, while Section 3.3 provides the complete benchmark construction pipeline. Appendix B and C include the prompts used in both evaluation settings. We promise to release the complete benchmark and codebase for community use.

REFERENCES

- Anthropic. Introducing Claude 4, May 2025. URL <https://www.anthropic.com/news/claude-4>. Claude Sonnet 4 and Opus 4 announcement.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL <https://arxiv.org/abs/2502.13923>.
- Yonatan Bitton, Hritik Bansal, Jack Hessel, Rulin Shao, Wanrong Zhu, Anas Awadalla, Josh Gardner, Rohan Taori, and Ludwig Schmidt. Visit-bench: A dynamic benchmark for evaluating instruction-following vision-and-language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/5503389dbe070cdae9b48086c4996a59-Abstract-Datasets_and_Benchmarks.html.
- ByteDance Seed. Seed1.6: Multimodal General-Purpose Model Series, June 2025. URL https://seed.bytedance.com/en/seed1_6. Official model page.
- Pengzhou Cheng, Zheng Wu, Zongru Wu, Tianjie Ju, Aston Zhang, Zhuosheng Zhang, and Gongshen Liu. Os-kairos: Adaptive interaction for mllm-powered GUI agents. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 6701–6725. Association for Computational Linguistics, 2025. URL <https://aclanthology.org/2025.findings-acl.348/>.
- DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.12948>.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao,

- 540 Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding,
541 Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang
542 Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai
543 Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang,
544 Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang,
545 Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang,
546 Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang,
547 R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye,
548 Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing
549 reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.
550 48550/ARXIV.2501.12948. URL <https://doi.org/10.48550/arXiv.2501.12948>.
- 551 Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong-Li Lee, and Wynne
552 Hsu. Video-of-thought: Step-by-step video reasoning from perception to cognition. *CoRR*,
553 abs/2501.03230, 2025. doi: 10.48550/ARXIV.2501.03230. URL [https://doi.org/10.](https://doi.org/10.48550/arXiv.2501.03230)
554 [48550/arXiv.2501.03230](https://doi.org/10.48550/arXiv.2501.03230).
- 555 Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei
556 Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. MME: A comprehensive eval-
557 uation benchmark for multimodal large language models. *CoRR*, abs/2306.13394, 2023. doi: 10.
558 48550/ARXIV.2306.13394. URL <https://doi.org/10.48550/arXiv.2306.13394>.
- 560 Timin Gao, Peixian Chen, Mengdan Zhang, Chaoyou Fu, Yunhang Shen, Yan Zhang, Shengchuan
561 Zhang, Xiawu Zheng, Xing Sun, Liujuan Cao, and Rongrong Ji. Cantor: Inspiring multimodal
562 chain-of-thought of MLLM. In Jianfei Cai, Mohan S. Kankanhalli, Balakrishnan Prabhakaran,
563 Susanne Boll, Ramanathan Subramanian, Liang Zheng, Vivek K. Singh, Pablo César, Lexing
564 Xie, and Dong Xu (eds.), *Proceedings of the 32nd ACM International Conference on Multime-*
565 *dia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pp. 9096–
566 9105. ACM, 2024. doi: 10.1145/3664647.3681249. URL [https://doi.org/10.1145/](https://doi.org/10.1145/3664647.3681249)
567 [3664647.3681249](https://doi.org/10.1145/3664647.3681249).
- 568 Gemini. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and
569 next generation agentic capabilities, 2025. URL <https://arxiv.org/abs/2507.06261>.
- 570 Yuan Guo, Tingjia Miao, Zheng Wu, Pengzhou Cheng, Ming Zhou, and Zhuosheng Zhang. Atomic-
571 to-compositional generalization for mobile agents with a new benchmark and scheduling system,
572 2025. URL <https://arxiv.org/abs/2506.08972>.
- 574 Rem Hida, Junki Ohmura, and Toshiyuki Sekiya. Evaluation of instruction-following ability for
575 large language models on story-ending generation. *CoRR*, abs/2406.16356, 2024. doi: 10.48550/
576 ARXIV.2406.16356. URL <https://doi.org/10.48550/arXiv.2406.16356>.
- 577 Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction
578 with pointer-augmented multimodal transformers for textvqa. In *2020 IEEE/CVF Conference on*
579 *Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp.
580 9989–9999. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01001.
581 URL [https://openaccess.thecvf.com/content_CVPR_2020/html/Hu_](https://openaccess.thecvf.com/content_CVPR_2020/html/Hu_Iterative_Answer_Prediction_With_Pointer-Augmented_Multimodal_Transformers_for_TextVQA_CVPR_2020_paper.html)
582 [Iterative_Answer_Prediction_With_Pointer-Augmented_Multimodal_](https://openaccess.thecvf.com/content_CVPR_2020/html/Hu_Iterative_Answer_Prediction_With_Pointer-Augmented_Multimodal_Transformers_for_TextVQA_CVPR_2020_paper.html)
583 [Transformers_for_TextVQA_CVPR_2020_paper.html](https://openaccess.thecvf.com/content_CVPR_2020/html/Hu_Iterative_Answer_Prediction_With_Pointer-Augmented_Multimodal_Transformers_for_TextVQA_CVPR_2020_paper.html).
- 584 Yiyan Ji, Haoran Chen, Qiguang Chen, Chengyue Wu, Libo Qin, and Wanxiang Che. Mpcc: A
585 novel benchmark for multimodal planning with complex constraints in multimodal large language
586 models, 2025. URL <https://arxiv.org/abs/2507.23382>.
- 588 Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin
589 Jiang, Qun Liu, and Wei Wang. Followbench: A multi-level fine-grained constraints following
590 benchmark for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar
591 (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*
592 *(Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 4667–4688.
593 Association for Computational Linguistics, 2024a. doi: 10.18653/V1/2024.ACL-LONG.257.
URL <https://doi.org/10.18653/v1/2024.acl-long.257>.

- 594 Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin
595 Jiang, Qun Liu, and Wei Wang. Followbench: A multi-level fine-grained constraints following
596 benchmark for large language models. In *ACL (1)*, pp. 4667–4688, 2024b. URL <https://doi.org/10.18653/v1/2024.acl-long.257>.
597
598
- 599 Wei Li, William E Bishop, Alice Li, Christopher Rawles, Folawiyo Campbell-Ajala, Divya Tyam-
600 agundlu, and Oriana Riva. On the effects of data scale on UI control agents. In *The Thirty-eight
601 Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.
602 URL <https://openreview.net/forum?id=yUEBXN3cvX>.
- 603 Zejun Li, Ruipu Luo, Jiwen Zhang, Minghui Qiu, Xuanjing Huang, and Zhongyu Wei. Vo-
604 cot: Unleashing visually grounded multi-step reasoning in large multi-modal models. In Luis
605 Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Na-
606 tions of the Americas Chapter of the Association for Computational Linguistics: Human Lan-
607 guage Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA,
608 April 29 - May 4, 2025*, pp. 3769–3798. Association for Computational Linguistics, 2025. doi:
609 10.18653/V1/2025.NAACL-LONG.192. URL <https://doi.org/10.18653/v1/2025.naacl-long.192>.
610
- 611 Guangyi Liu, Pengxiang Zhao, Liang Liu, Zhiming Chen, Yuxiang Chai, Shuai Ren, Hao Wang,
612 Shibo He, and Wenchao Meng. Learnact: Few-shot mobile gui agent with a unified demonstration
613 benchmark, 2025. URL <https://arxiv.org/abs/2504.13805>.
614
- 615 Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In Al-
616 ice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine
617 (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural
618 Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 -
619 16, 2023*, 2023. URL [http://papers.nips.cc/paper_files/paper/2023/hash/
620 6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html).
- 621 Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi
622 Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model
623 an all-around player? In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten
624 Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan,
625 Italy, September 29-October 4, 2024, Proceedings, Part VI*, volume 15064 of *Lecture Notes in
626 Computer Science*, pp. 216–233. Springer, 2024. doi: 10.1007/978-3-031-72658-3_13. URL
627 https://doi.org/10.1007/978-3-031-72658-3_13.
- 628 Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu,
629 Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reason-
630 ing via thought chains for science question answering. In Sanmi Koyejo, S. Mo-
631 hamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural
632 Information Processing Systems 35: Annual Conference on Neural Information Process-
633 ing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9,
634 2022*, 2022. URL [http://papers.nips.cc/paper_files/paper/2022/hash/
635 11332b6b6cf4485b84afadb1352d3a9a-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2022/hash/11332b6b6cf4485b84afadb1352d3a9a-Abstract-Conference.html).
- 636 Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-
637 Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning
638 of foundation models in visual contexts. In *The Twelfth International Conference on Learning
639 Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL
640 <https://openreview.net/forum?id=KUNzEQMWU7>.
- 641 Bozhi Luan, Hao Feng, Hong Chen, Yonghui Wang, Wengang Zhou, and Houqiang Li. Textcot:
642 Zoom in for enhanced multimodal text-rich image understanding. *CoRR*, abs/2404.09797, 2024.
643 doi: 10.48550/ARXIV.2404.09797. URL [https://doi.org/10.48550/arXiv.2404.
644 09797](https://doi.org/10.48550/arXiv.2404.09797).
645
- 646 Meta AI. The Llama 4 Herd: The Beginning of a New Era of Natively Mul-
647 timodal Intelligence, April 2025. URL [https://ai.meta.com/blog/
llama-4-multimodal-intelligence/](https://ai.meta.com/blog/llama-4-multimodal-intelligence/). Official announcement.

- 648 OpenAI. Introducing GPT-5, August 2025a. URL [https://openai.com/index/
649 introducing-gpt-5/](https://openai.com/index/introducing-gpt-5/). Official announcement.
650
- 651 OpenAI. Introducing OpenAI o3 and o4-mini, April 2025b. URL [https://openai.com/
652 index/introducing-o3-and-o4-mini/](https://openai.com/index/introducing-o3-and-o4-mini/). Model family announcement; use with o3 or
653 o3-pro.
- 654 Kun Ouyang. Spatial-r1: Enhancing mllms in video spatial reasoning. *CoRR*, abs/2504.01805,
655 2025. doi: 10.48550/ARXIV.2504.01805. URL [https://doi.org/10.48550/arXiv.
656 2504.01805](https://doi.org/10.48550/arXiv.2504.01805).
- 657 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong
658 Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kel-
659 ton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike,
660 and Ryan Lowe. Training language models to follow instructions with human feedback. In
661 Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neu-
662 ral Information Processing Systems*, 2022. URL [https://openreview.net/forum?id=
663 TG8KACxEON](https://openreview.net/forum?id=TG8KACxEON).
- 664 Jiazhen Pan, Che Liu, Junde Wu, Fenglin Liu, Jiayuan Zhu, Hongwei Bran Li, Chen Chen, Cheng
665 Ouyang, and Daniel Rueckert. Medvlm-r1: Incentivizing medical reasoning capability of vision-
666 language models (vlms) via reinforcement learning. *CoRR*, abs/2502.19634, 2025. doi: 10.48550/
667 ARXIV.2502.19634. URL <https://doi.org/10.48550/arXiv.2502.19634>.
- 668 Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning
669 with gpt-4, 2023. URL <https://arxiv.org/abs/2304.03277>.
- 670 Yusu Qian, Hanrong Ye, Jean-Philippe Fauconnier, Peter Grasch, Yinfei Yang, and Zhe Gan. Mia-
671 bench: Towards better instruction following evaluation of multimodal llms. In *The Thirteenth
672 International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*.
673 OpenReview.net, 2025. URL <https://openreview.net/forum?id=7EhS3YBxjY>.
- 674 Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng
675 Wu, Fei Liu, Pengfei Liu, and Dong Yu. Infobench: Evaluating instruction following ability in
676 large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of
677 the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meet-
678 ing, August 11-16, 2024*, pp. 13025–13048. Association for Computational Linguistics, 2024.
679 doi: 10.18653/V1/2024.FINDINGS-ACL.772. URL [https://doi.org/10.18653/v1/
680 2024.findings-acl.772](https://doi.org/10.18653/v1/2024.findings-acl.772).
- 681 Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Mary-
682 beth Fair, Alice Li, William E Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Kenji Toyama,
683 Robert James Berry, Divya Tyamagundlu, Timothy P Lillicrap, and Oriana Riva. Androidworld:
684 A dynamic benchmarking environment for autonomous agents. In *The Thirteenth International
685 Conference on Learning Representations*, 2025. URL [https://openreview.net/forum?
686 id=il5yUQsrjC](https://openreview.net/forum?id=il5yUQsrjC).
- 687 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
688 Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathe-
689 matical reasoning in open language models, 2024. URL [https://arxiv.org/abs/2402.
690 03300](https://arxiv.org/abs/2402.03300).
- 691 Ke Wang, Juntong Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and
692 Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. In *The
693 Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks
694 Track*, 2024. URL <https://openreview.net/forum?id=QWTCcxMpPA>.
- 695 Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and
696 Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions.
697 In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual
698 Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–
699 13508, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/
700 v1/2023.acl-long.754. URL <https://aclanthology.org/2023.acl-long.754/>.
- 701

- 702 Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu,
703 Wendy Gao, Jiaying Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. Benchmark-
704 ing complex instruction-following with multiple constraints composition. In *The Thirty-eight*
705 *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.
706 URL <https://openreview.net/forum?id=U2aVNDrZGx>.
- 707 Xiaodong Wu, Minhao Wang, Yichen Liu, Xiaoming Shi, He Yan, Xiangju Li, Junmin Zhu, and
708 Wei Zhang. Lifbench: Evaluating the instruction following performance and stability of large
709 language models in long-context scenarios. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova,
710 and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association*
711 *for Computational Linguistics (Volume 1: Long Papers)*, ACL 2025, Vienna, Austria, July 27
712 - August 1, 2025, pp. 16445–16468. Association for Computational Linguistics, 2025. URL
713 <https://aclanthology.org/2025.acl-long.803/>.
- 714 Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal LLM logical reasoning
715 benchmark in visual contexts. *CoRR*, abs/2407.04973, 2024. doi: 10.48550/ARXIV.2407.04973.
716 URL <https://doi.org/10.48550/arXiv.2407.04973>.
- 717 Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and
718 Yu Su. Travelplanner: A benchmark for real-world planning with language agents. In *Forty-first*
719 *International Conference on Machine Learning*, 2024a. URL [https://openreview.net/](https://openreview.net/forum?id=15XQzNkAOe)
720 [forum?id=15XQzNkAOe](https://openreview.net/forum?id=15XQzNkAOe).
- 721 Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing
722 Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio
723 Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal
724 agents for open-ended tasks in real computer environments. In Amir Globersons, Lester
725 Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang
726 (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural*
727 *Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10*
728 *- 15, 2024*, 2024b. URL [http://papers.nips.cc/paper_files/paper/2024/](http://papers.nips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html)
729 [hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_](http://papers.nips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html)
730 [Benchmarks_Track.html](http://papers.nips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html).
- 731 Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang,
732 and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities.
733 In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria,*
734 *July 21-27, 2024*. OpenReview.net, 2024. URL [https://openreview.net/forum?id=](https://openreview.net/forum?id=KOTutrSR2y)
735 [KOTutrSR2y](https://openreview.net/forum?id=KOTutrSR2y).
- 736 Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. RRHF:
737 Rank responses to align language models with human feedback. In *Thirty-seventh Conference on*
738 *Neural Information Processing Systems*, 2023. URL [https://openreview.net/forum?](https://openreview.net/forum?id=EdIGMCHk4l)
739 [id=EdIGMCHk4l](https://openreview.net/forum?id=EdIGMCHk4l).
- 740 Xiang Yue, Yuansheng Ni, Tianyu Zheng, Kai Zhang, Ruoqi Liu, Ge Zhang, Samuel Stevens,
741 Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun,
742 Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and
743 Wenhui Chen. MMMU: A massive multi-discipline multimodal understanding and reason-
744 ing benchmark for expert AGI. In *IEEE/CVF Conference on Computer Vision and Pattern*
745 *Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 9556–9567. IEEE, 2024.
746 doi: 10.1109/CVPR52733.2024.00913. URL [https://doi.org/10.1109/CVPR52733.](https://doi.org/10.1109/CVPR52733.2024.00913)
747 [2024.00913](https://doi.org/10.1109/CVPR52733.2024.00913).
- 748 Jingyi Zhang, Jiaying Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng
749 Tao. R1-VL: learning to reason with multimodal large language models via step-wise group
750 relative policy optimization. *CoRR*, abs/2503.12937, 2025a. doi: 10.48550/ARXIV.2503.12937.
751 URL <https://doi.org/10.48550/arXiv.2503.12937>.
- 752 Tao Zhang, Chenglin Zhu, Yanjun Shen, Wenjing Luo, Yan Zhang, Hao Liang, Fan Yang, Mingan
753 Lin, Yujing Qiao, Weipeng Chen, Bin Cui, Wentao Zhang, and Zenan Zhou. Cfbench: A compre-
754 hensive constraints-following benchmark for llms. In Wanxiang Che, Joyce Nabende, Ekaterina
755

756 Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the*
757 *Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2025, Vienna, Austria,
758 July 27 - August 1, 2025, pp. 32926–32944. Association for Computational Linguistics, 2025b.
759 URL <https://aclanthology.org/2025.acl-long.1581/>.

760
761 Tao Zhang, Chenglin Zhu, Yanjun Shen, Wenjing Luo, Yan Zhang, Hao Liang, Tao Zhang, Fan
762 Yang, Mingan Lin, Yujing Qiao, Weipeng Chen, Bin Cui, Wentao Zhang, and Zenan Zhou.
763 Cfbench: A comprehensive constraints-following benchmark for llms, 2025c. URL <https://arxiv.org/abs/2408.01122>.
764

765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

Table 5: Comparison of two error examples, presented in a column-wise layout for side-by-side inspection.

Item	Example 1	Example 2
Image		
Error Type	Must-have Conflict, No Refusal	Misclassification of Nice-to-have as Must-have
User Requirements	<p>Must-have: 1 King Bed required.</p> <p>Must-have: Must accommodate 4 guests.</p> <p>Nice-to-have: Prefer balcony and partial ocean view.</p>	<p>Must-have: Price per night \leq \$300.</p> <p>Must-have: Must accommodate 2 guests.</p> <p>Nice-to-have: Prefer rating \geq 8.5.</p>
Model Output (Incorrect)	Recommended “Room, 2 Double Beds, Balcony, Partial Ocean View”, 377 sq ft, accommodates 4 guests, includes balcony and partial ocean view — claimed to fully meet the request.	“Sorry, no available rooms meet the requirements.”
Expected Output	The system should reject or indicate no available room meets all must-have requirements. Example: “No room satisfies both the ‘1 King Bed’ and ‘4 guests’ requirements.”	The system should treat rating criterion as optional and list all rooms satisfying the price and capacity requirements, possibly ranked by rating. Example: “3 rooms found under \$300 for 2 guests, sorted by rating.”

A REPRESENTATIVE EXAMPLES FROM FTF-BENCH

Table 5 presents representative instances from FTF-BENCH, illustrating typical error types made by MLLMs. It shows how models often misclassify must-have versus nice-to-have requirements or fail to properly reject infeasible requests.

B EVALUATION PROMPTS

The detailed prompt for evaluating FTF-BENCH in Section 5.2 is shown below:

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

Prompts for Evaluating FTF-BENCH

Given the following task requirements, strictly select the product from the image that best satisfies the criteria. If no product satisfies all the "must-have requirements," explicitly return a refusal. Selection must be based solely on the information provided in the task and visible in the image; no additional reasoning or assumptions are allowed. The rules are as follows:

1. Requirement types:

- "Must-have requirements"
- "Nice-to-have requirements"

2. Must-have requirements:

- A product must satisfy all must-have requirements simultaneously; missing any requirement disqualifies it.
- Any product failing to meet even one must-have requirement is considered ineligible and cannot proceed to further comparison.

3. Priority-based selection of nice-to-have requirements (applied only if more than 2 candidate products remain):

- Only when multiple products satisfy all must-have requirements should nice-to-have requirements be used for further selection.
- Compare products sequentially according to the priority of nice-to-have requirements, from highest to lowest (priority 1 being the highest).
- Select the product that satisfies more of the higher-priority requirements. If there is a tie at one priority level, continue comparing the next lower priority until a unique best product is identified.
- Example: If priority is ordered as 1>2>3..., and Product A satisfies priorities 1, 2, 3 while Product B satisfies priorities 1, 2, 4, 5, Product A should be chosen because priority 3 is higher than 4 or 5.

Output rules:

1. If at least one product satisfies all must-have requirements:
 - If only one product qualifies, directly output its name.
 - If multiple products qualify, apply the nice-to-have priority rules to select the unique best product and output its name.
2. If no product satisfies all must-have requirements, output the refusal text: "No product satisfies all must-have requirements."

Output format:

- Must output strictly in JSON format and only output the JSON.
- Format:


```
{"Answer": ""}
```
- "Answer": Fill in the name of the selected product. If no product qualifies, set the value to "No product satisfies all must-have requirements" (including quotes).

The question is as follows:

<question>

Strictly follow the above rules and base the selection solely on the information provided in the task and visible in the image.

C PROMPT DESIGN FOR BENCHMARK CONSTRUCTION

To systematically construct our benchmark, we designed a set of task-oriented prompts that guide the generation of evaluation questions based on the information present in images. For each task scenario, the prompts explicitly define the core objectives, requirement categories, and output format to ensure consistency and rigor in the generated data. For instance, in the *Single-Answer* scenario, the prompt instructs the model to select one product as the correct answer while providing must-have and nice-to-have requirements with explicit priority levels. Similarly, for *Unanswerable* scenarios, the prompt is carefully constructed to ensure that no product in the image can satisfy all must-have requirements, while optional requirements create a controlled level of confusion.

918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

Prompt for Generating Requirements in Single-Answer Tasks for FTF-BENCH

Given an image displaying products or services, generate a question that asks users to select a product or service based on their needs. Both the question and its answer must be strictly grounded in the information presented in the image; no content should be fabricated. The requirements are as follows:

1. Determining the correct answer

- Before generating the question, select one product or service from the candidates in the image as the correct answer.
- The question must be designed around this selected item, while one or more other items serve as distracting/confusing options.
- All content must be strictly based on the image; no invented details are allowed.
- **Core principle:** The question should test the ability to choose the correct item based on user requirements using the visual and contextual information in the image.

2. Requirement categories and design

- **Must-have requirements:**
 - These requirements must uniquely identify the correct answer, forming a minimal necessary set.
 - Each requirement in this set is indispensable—removing any single one may result in multiple items satisfying the criteria.
 - The set must not contain redundant requirements—if removing a requirement still uniquely identifies the correct answer, that requirement is considered redundant and should be omitted.
- **Nice-to-have requirements:**
 - These are optional requirements that may or may not be satisfied.
 - They must be ordered by priority from highest to lowest.
 - High-priority nice-to-have requirements are intended to make confusing options more appealing, while the correct answer must satisfy all must-have requirements and may satisfy only a subset of nice-to-have requirements.

3. Design for confusion

- The question should be challenging and complex.
- Distractor options should satisfy more of the high-priority nice-to-have requirements, whereas the correct answer fully satisfies must-have requirements but only partially satisfies nice-to-have requirements.
- This encourages users to overemphasize nice-to-have requirements and potentially overlook the must-have requirements, which are the decisive criteria.

4. Output format

- The output should be a JSON object with the following fields:
 - "Answer": the correct answer
 - "confusing_answer": the distracting option(s)
 - "must_have_requirements": a list of must-have requirements
 - "nice_to_have_requirements": a list of nice-to-have requirements ordered from highest to lowest priority

972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

Prompt for Generating Requirements in Multiple-Answer Tasks for FTF-BENCH

Based on the products or services shown in the provided image, generate a question that asks users to select a product or service according to specific requirements. Both the question and the answer must be strictly grounded in the information visible in the image; no content should be fabricated. The requirements are as follows:

1. Determining the correct answer

- Before generating the question, select one product or service from the image as the correct answer.
- All questions must be designed around this selected product, while one or more other products serve as distracting/confusing options.
- Core principle: The question should reflect how the correct answer satisfies the requirements relative to the other candidates in the image.

2. Requirement categories and design

- **Must-have requirements:**
 - These requirements should form a candidate pool containing multiple products from the image, including the correct answer.
- **Nice-to-have requirements:**
 - Organize these into a list with explicit priority, from highest to lowest (smaller priority value indicates higher importance).
 - By sequentially matching according to priority, the correct answer should be uniquely determined from the candidate pool.
 - Requirement content should be realistic and contextually appropriate.

3. Difficulty and distractor design

- The question should be challenging, with complex and diverse requirements.
- High-priority nice-to-have requirements may conflict with must-have requirements to increase difficulty.
- Distractor options should satisfy more of the high-priority nice-to-have requirements but fail to meet must-have requirements; the correct answer may satisfy fewer high-priority nice-to-have requirements but must fully satisfy all must-have requirements.

4. Output format

- The output must be strictly in JSON format and contain only the JSON object, with the following fields:
 - "mandatory_requirements": a list of must-have requirements
 - "optional_requirements": a list of nice-to-have requirements with priorities, each represented as an object containing "priority" and "requirement"
 - "confusing_answer": the distracting option
 - "final_selected_product": the product ultimately selected after applying the optional requirements

Strictly follow the above rules and base the selection solely on the information provided in the task and visible in the image.

1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079

Prompt for Generating Requirements in Unanswerable Tasks for FTF-BENCH

Based on the products or services shown in the provided image, generate a question that asks users to select a product or service according to specific requirements. The question must be designed so that no product or service in the image can satisfy all core requirements. Both the question and its answer must be strictly grounded in the information visible in the image; no content should be fabricated. The requirements are as follows:

1. Core objective

- The generated question must ensure that all products or services in the image cannot become the correct answer; that is, no product/service can fully satisfy the task’s core requirements.

2. Requirement categories and design

- Must-have requirements:

- Must satisfy two conditions:

1. Each individual requirement should be satisfied by at least one product or service in the image.
2. When all must-have requirements are combined, no single product or service should satisfy them all (i.e., each requirement has a corresponding “matching product,” but no product matches all requirements).

- Nice-to-have requirements (optional/distractor requirements):

- Organize as a list with explicit priority, from highest to lowest (smaller priority value indicates higher importance).

- Some optional requirements should be satisfied by most candidate products to create the illusion that a perfect product exists.

- Each requirement should be fluent, reasonable, and reflect common, real-world user needs.

3. Difficulty and distractor design

- The question should be complex and challenging, emphasizing confusion.

- Each product/service in the image should satisfy most of the must-have requirements, failing only on 1–2 critical requirements.

- Each product/service should satisfy some high-priority nice-to-have requirements, creating a “each has advantages” scenario, making it difficult to realize that no product fully satisfies all must-have requirements.

4. Output format

- The output must be strictly in JSON format and contain only the JSON object, with the following fields:

- "mandatory_requirements": list of selected must-have requirements

- "optional_requirements": list of selected nice-to-have requirements with priority, each represented as an object containing "priority" and "requirement"

- "candidate_products": list of products that satisfy some, but not all, must-have requirements (serving as candidate options)

Strictly follow the above rules and base the selection solely on the information provided in the task and visible in the image. Ensure that no single product fully satisfies all must-have requirements.

D DETAILS OF HUMAN VERIFICATION

To ensure the quality and reliability of our dataset, we implemented a rigorous human verification process. Specifically, we developed a custom annotation script that streamlined the verification workflow, as shown in figure 3, allowing annotators to efficiently inspect each instance. The script presents the input data, associated candidate options, and relevant contextual information in a user-friendly interface, while logging the annotators’ selections and comments for further analysis.

Annotators were instructed to evaluate multiple aspects of each item. These include the reasonableness of the question, the accuracy of the answer, the correctness of requirement categorization and requirement descriptions. Each instance was independently reviewed by at least two annotators. Disagreements were resolved through discussion or adjudication by a senior annotator. This procedure not only guarantees high annotation accuracy but also establishes a transparent and reproducible verification pipeline.

During the human verification process, we systematically examined the model’s erroneous predictions and identified several prominent error types, as shown in Figure 4. The model often failed to follow *must-have requirements*, and sometimes treated *nice-to-have requirements* as mandatory and

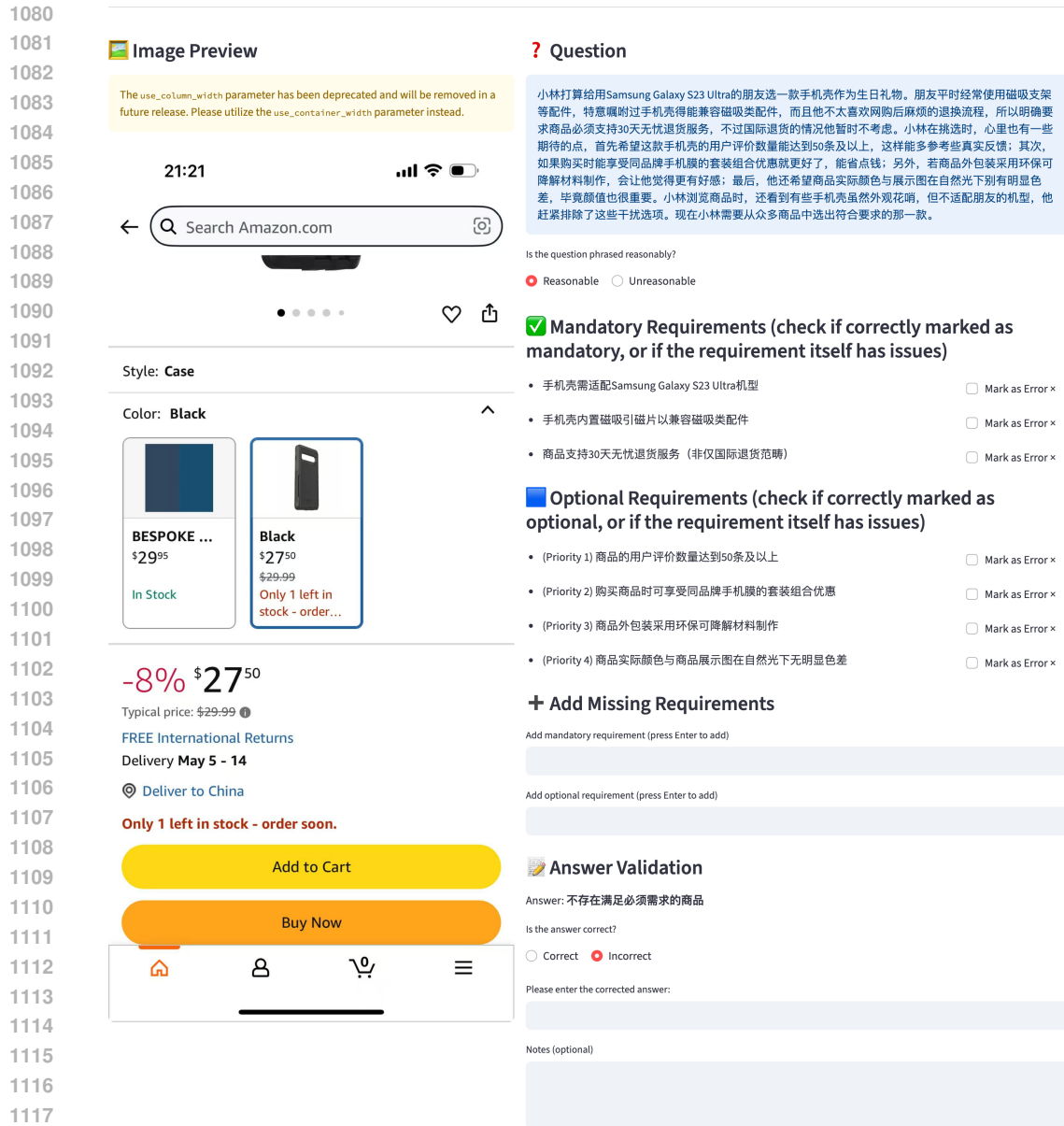


Figure 3: Screenshot of the human verification annotation interface. Annotators are asked to evaluate question reasonableness, answer accuracy, requirement categorization, and description correctness.

occasionally misordered them, reflecting challenges in understanding and prioritizing user preferences.

E ABLATION STUDY OF REWARD FUNCTIONS

To evaluate the contribution of requirement classification supervision, we conduct an ablation study by removing $R_{\text{requirement}}$, R_{format} , R_{answer} from the overall objective and retraining the model under identical settings, respectively. The results across three benchmarks are summarized in Table 6. We observe a consistent performance drop across all datasets. The performance drop indicates that all reward components contribute to the final performance. This finding confirms that multi-objective reinforcement learning, which jointly optimizes for format compliance, answer accuracy,

1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187

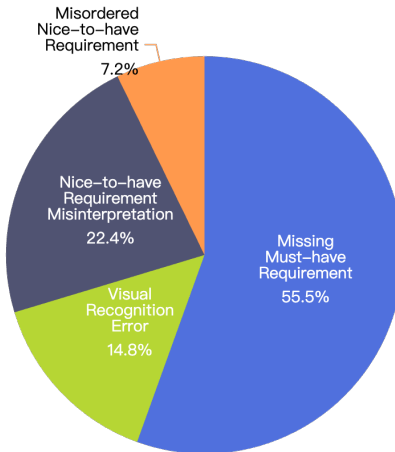


Figure 4: Distribution of model error types identified during human verification. Models frequently failed to follow *must-have requirements*, sometimes treated *nice-to-have requirements* as mandatory, and occasionally misordered them, highlighting challenges in understanding and prioritizing user preferences.

Table 6: Ablation study of the different rewards on Qwen2.5-VL-7B-Instruct.

Setting	FTF-BENCH _{val}	LogicVista	MathVision	InfoQA
Full FTF-RL	55.8	47.4	25.7	67.8
Ablated (w/o $R_{\text{requirement}}$)	52.4↓3.4	44.7↓2.7	24.7↓1.0	60.3↓7.5
Ablated (w/o R_{format})	54.5↓1.3	45.8↓1.6	25.1↓0.6	64.5↓3.3
Ablated (w/o R_{answer})	48.3↓7.5	43.7↓3.7	24.5↓1.2	61.2↓6.6

requirement understanding, and reasoning quality, is crucial for improving accuracy in multimodal document understanding tasks.

F MODEL-DEPENDENT BIAS IN DATA CREATION AND EVALUATION

In our pipeline, we use `Doubao-Seed-1.6-250615` to generate the initial requirement sets and candidate answers, and also as the judge. All items are then checked by human annotators, who verify both the realism of the user requirements and the correctness of the final answer. Table 7 summarizes the joint distribution of human judgments on question reasonableness and answer correctness.

Most discrepancies come from prompts that annotators consider unrealistic in real life, while the majority of retained items have both reasonable requirements and correct answers. This indicates that `Doubao-Seed-1.6-250615` is capable of serving as a generator/judge, and that the subsequent human pass substantially reduces residual errors and potential same-source bias in the benchmark.

To further address the concern regarding model-dependent bias, we also substituted the judge model with GPT-5 while keeping the same evaluation protocol in Table 8. We observe very high cross-model agreement across all three scenarios, which suggests that label reliability does not depend on a single model family.

G REQUIREMENT CLASSIFICATION CONFUSION MATRICES

To better understand how MLLMs distinguish between must-have and nice-to-have requirements, we further evaluate their performance on a binary requirement-classification task over FTF-BENCH. For each requirement, we compare the gold label (must-have and nice-to-have) with the label predicted by the MLLM in the *Direct* setting and aggregate the counts into a confusion matrix.

Table 7: Joint distribution of human judgments on question reasonableness (question_reasonable) and answer correctness (has_error).

Human judgment	Answer has error	Answer correct	Total
Question unreasonable	683 (18.72%)	51 (1.41%)	734 (20.13%)
Question reasonable	361 (9.90%)	2554 (69.98%)	2915 (79.87%)
Total	1044 (28.62%)	2605 (71.38%)	3649 (100%)

Table 8: Agreement between Doubao-Seed-1.6-250615 and GPT-5.

Setting	Total	Num of Agree	Num of Disagree	Agree Rate
Single Answer	3111	3093	17	99.42%
Multiple Answer	3072	3057	14	99.51%
Unanswerable	4764	4749	15	99.69%

Table 9: Confusion matrix for requirement classification on FTF-BENCH (Direct setting) using Doubao-1.6-seed.

Ground truth / Prediction	Predicted must-have	Predicted nice-to-have	Total
Must-have	5221 (64.74%)	2844 (35.26%)	8065
Nice-to-have	2068 (16.79%)	10246 (83.21%)	12314
Total	7289	13090	20379

Table 10: Confusion matrix for requirement classification on FTF-BENCH (Direct setting) using GPT-5.

Ground truth / Prediction	Predicted must-have	Predicted nice-to-have	Total
Must-have	6425 (79.67%)	1640 (20.33%)	8065
Nice-to-have	1418 (11.52%)	10896 (88.48%)	12314
Total	7843	12486	20379

Table 11: Confusion matrix for requirement classification on FTF-BENCH (Direct setting) using Qwen2.5-VL-7B-Instruct.

Ground truth / Prediction	Predicted must-have	Predicted nice-to-have	Total
Must-have	4073 (50.52%)	3992 (49.48%)	8065
Nice-to-have	4229 (34.34%)	8085 (65.66%)	12314
Total	7843	12486	20379

Tables 9, 10, and 11 summarize the confusion matrices for Doubao-1.6-seed, GPT-5, and Qwen2.5-VL-7B-Instruct.

It can be seen that GPT-5 behaves in a relatively balanced way for both must-have and nice-to-have requirements. Doubao-1.6-seed shows strong recognition of nice-to-have requirements, but it also tends to classify more must-have requirements as nice-to-have. Qwen2.5-VL-7B-Instruct, as a representative open-source LLM, has a generally weaker capability to correctly separate the two types of requirements.

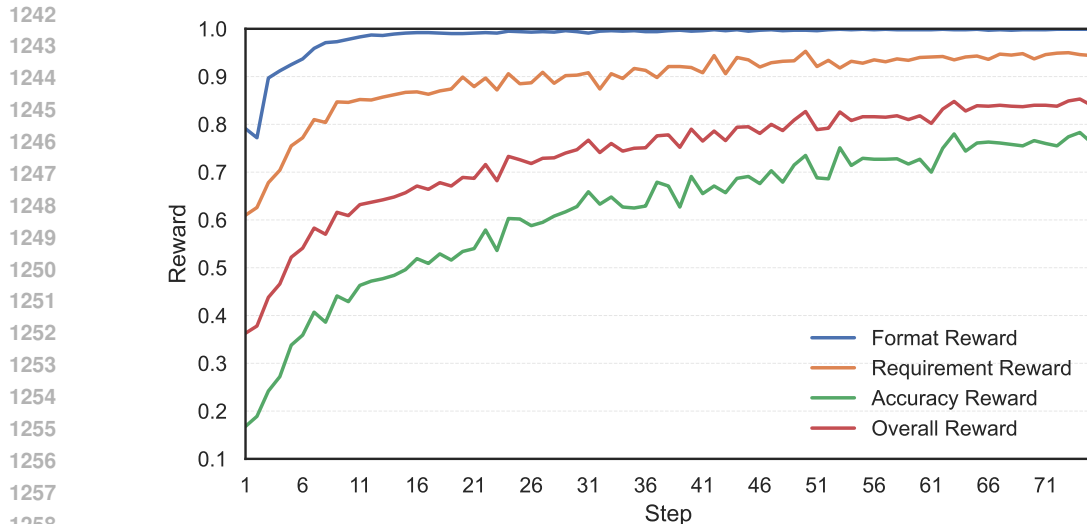


Figure 5: Training reward curves of Qwen2.5-VL-7B-Instruct under FTF-RL.

H TRAINING DYNAMICS OF FTF-RL ON QWEN2.5-VL-7B-INSTRUCT

We further analyze the behavior of FTF-RL by tracking the evolution of different rewards over the whole training process on Qwen2.5-VL-7B-Instruct. We monitor the format reward, the requirement classification reward, the answer correctness reward, and the aggregated overall reward at multiple training steps.

As shown in Figure 5, all four rewards improve monotonically during training, leading to a steady rise in the overall reward.

I QUALITATIVE CASE STUDIES ON GENERAL REASONING BENCHMARKS

To illustrate how requirement-aware reasoning learned from FTF-BENCH transfers to other tasks, we present two short case studies on LogicVista and MathVision using Qwen2.5-VL-7B-Instruct trained with FTF-RL. In both examples, the model only sees the original benchmark input and no additional supervision.

We provide the two cases in Table 12 and 13. In both cases, the trained model follows a consistent pattern. The first case is a MathVision geometry problem about a circular carpet on a tiled floor. The model must decide which grey tile pattern cannot come from any circle. After FTF-RL, Qwen2.5-VL-7B-Instruct does not jump directly to an option. It first summarizes the task requirement that grey tiles must be exactly those intersected by a single convex circle, so they should form one connected region with a smooth boundary. It then explicitly plans to check each candidate against this requirement and finally concludes that the option with a disconnected grey region is impossible. This shows that the model uses a requirement-driven plan rather than local pattern matching.

The second case is a LogicVista problem that involves inferring the meaning of two symbolic operations applied to shapes. The model must fill in a missing output shape and a missing operation symbol. After FTF-RL, Qwen2.5-VL-7B-Instruct begins by restating the subgoals, then applies these inferred rules to the two queries. It then follows this plan step by step and correctly selects the option. The model learns to organize the task into requirement extraction and execution.

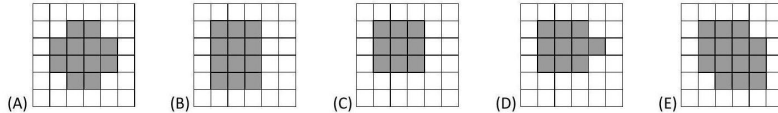
J ADDITIONAL EXPERIMENTS ON COMPLEX AGENT TASKS

To further assess the generality of requirement-aware reasoning beyond the three service-oriented domains in our main experiments, we conduct additional evaluations on complex agent tasks. We select two representative benchmarks.

Table 12: Qualitative case study on MathVision after training on Qwen2.5-VL-7B-Instruct.

Question. A circular carpet is placed on a floor which is covered by equally big, square tiles. All tiles that have at least one point in common with the carpet are coloured in grey. Which of the following cannot be a result of this? <image>

Image.



Output.

<requirements>"mandatory": ["Identify which grey tile pattern CANNOT result from coloring all tiles touching a circular carpet (tiles with at least one point in common with the circle)],"optional": ["Explain why the pattern is invalid"]</requirements>

<think>

To solve this problem, we need to understand how placing a circular carpet would affect the surrounding squares if they touch it anywhere.

A circle is a convex, smooth, and symmetric shape. Tiles colored by a circle (all tiles intersecting the circle) must form a connected, "rounded" boundary (no sharp corners or disconnected "extensions" that cannot be touched by a circle).

Let's analyze each option:

A: The shape is connected with smooth edges. A circle could touch these tiles.

B: A tall, narrow shape—still connected and smooth; a circle could align with this.

C: A compact, connected shape; a circle could fit this.

D: The shape has a small extra tile on the right side but remains connected and smooth. A circle could touch this.

E: The the bottom-right tiles are separated from the main block by an empty tile. A circle is convex and continuous—it cannot touch tiles that are disconnected from the main group. The empty tile between the main block and the lower-right tiles means the circle cannot reach both regions without also touching the empty tile.

Thus, E cannot be the result.

</think>

<answer>E</answer>

AndroidControl (Li et al., 2024) evaluates mobile GUI control in realistic Android environments. It consists of human demonstrations of everyday tasks across diverse apps, where each trajectory records the screen observations, natural language instructions, and the corresponding low-level actions executed by the user. We evaluate performance on AndroidControl using Task Match Rate (TMR) and Action Match Rate (AMR) as our main metrics.

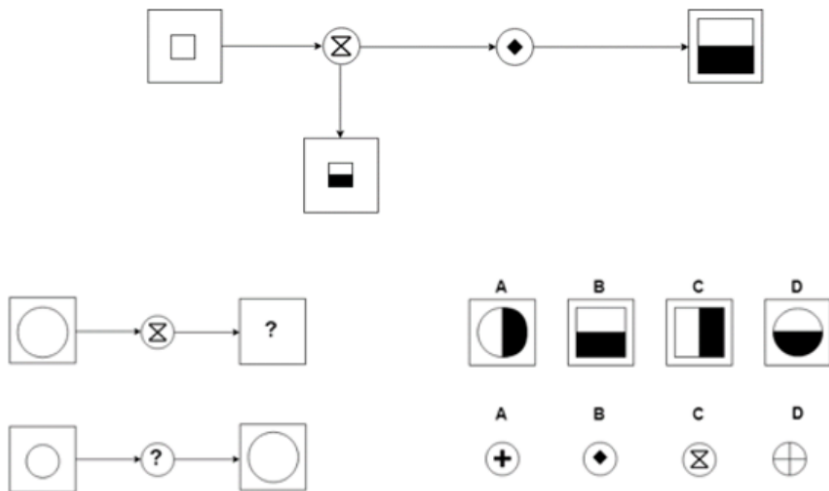
ScienceQA (Lu et al., 2022) captures multi-step scientific reasoning, which serves as a comprehensive testbed for multimodal reasoning across diverse science topics. It features a diverse collection of science questions covering 26 topics and 127 categories, where each example is annotated with CoT explanations and lectures to assess the model's ability to perform complex scientific reasoning and explanation generation. The metric is accuracy, measuring the number of corrected answers provided by models.

Table 14 and 15 report the performance before and after FTF-RL training on ScienceQA and AndroidControl, respectively. Across both benchmarks, we observe consistent improvements after FTF-RL, mirroring the gains we reported previously on LogicVista, MathVision, and MathVista. This shows that learning to prioritize requirements on FTF-BENCH transfers to more complex agent behaviors.

Table 13: Qualitative case study on LogicVista after training on Qwen2.5-VL-7B-Instruct.

Question. Based on the diagram at the top of the page, which two of the proposals A, B, C or D completes the diagrams at the bottom of the page? <image>

Image.



Output.

<requirements>"mandatory": ["Analyze the top diagram to determine the transformation rules of operations for shapes within square frames", "Apply the rules to select the correct shape (from A/B/C/D) for the '?' in the first bottom diagram", "Apply the rules to select the correct operation (from A/B/C/D) for the '?' in the second bottom diagram"], "optional": ["Explain each operation's function with reference to the top diagram"]</requirements>

<think>

First, I need to re-extract the operation rules from the top diagram. The top diagram has two inputs: a square with a small white square, and a square with a small square.

Operation cross symbol combines the "small inner shapes" of the two inputs.

Operation the diamond symbol then scales the combined small shape to fill the entire outer square frame.

Next, for the first bottom diagram: Input is a square with a small white circle.

Operation cross symbol should combine this small circle with the "bottom half black" fill.

So the result should be a square frame containing a small circle (bottom half black) — which is option D.

For the second bottom diagram:

Input is a square with a small white circle; output is a square with a large circle.

This scaling function matches operation the diamond symbol from the top diagram — which is option B.

</think>

<answer>D, B</answer>

Table 14: Performance on ScienceQA after training with FTF-RL on FTF-BENCH.

Model	ScienceQA
Qwen2.5-VL-7B-Instruct	40.06
+FTF-RL	40.28 ↑0.22

Table 15: Performance on AndroidControl after training with FTF-RL on FTF-BENCH.

Model	Metric	CLICK	TYPE	SCROLL	OPENAPP	WAIT	COMPLETE	PRESS
Qwen2.5-VL-7B-Instruct	TMR	0.9538	0.8880	0.8571	0.0000	0.0459	0.9339	0.2857
	AMR	0.3317	0.6800	0.0603	0.0000	0.0459	0.9339	0.2857
+FTF-RL	TMR	0.9608 \uparrow 0.0071	0.9216 \uparrow 0.0336	0.8026 \downarrow 0.0545	0.0000 \uparrow 0.0000	0.0529 \uparrow 0.0071	0.9533 \uparrow 0.0194	0.5364 \uparrow 0.2507
	AMR	0.3240 \downarrow 0.0077	0.7184 \uparrow 0.0384	0.0446 \downarrow 0.0157	0.0000 \uparrow 0.0000	0.0529 \uparrow 0.0071	0.9533 \uparrow 0.0194	0.5364 \uparrow 0.2507

Table 16: Performance of all models on the full FTF-BENCH and on the 10% evaluation subset.

Model	Split	Sin. Upper	Sin. Direct	Mul. Upper	Mul. Direct	Unans. Upper	Unans. Direct	Avg. Upper	Avg. Direct
Gemini-2.5-pro	Full	88.89	86.91	84.20	82.26	81.72	78.55	84.26	81.75
	Subset	79.61	81.55	72.28	69.31	63.29	58.23	69.16	66.57
GPT-o3	Full	82.67	77.78	80.41	80.03	83.31	83.09	82.33	79.68
	Subset	75.73	72.15	69.31	75.25	81.01	73.79	75.22	72.33
Doubao-1.6-seed	Full	80.09	78.66	81.70	76.99	83.80	84.05	82.24	80.59
	Subset	72.82	66.99	63.37	58.42	74.68	68.99	69.74	63.98
LLaMA-4	Full	58.36	55.48	55.93	54.45	44.43	36.94	55.03	51.99
	Subset	54.37	49.51	41.58	25.32	30.38	40.59	37.75	33.72
Qwen2.5-VL-7B-Instruct	Full	61.16	22.99	58.25	20.10	47.59	18.64	57.69	21.05
	Subset	57.28	20.39	39.60	11.88	37.34	9.49	41.21	9.51
Qwen2.5-VL-32B-Instruct	Full	70.66	69.25	70.55	70.17	53.08	50.58	67.72	66.57
	Subset	66.99	67.96	42.41	34.18	54.46	58.42	50.72	48.41

Table 17: DOUBAO-1.6-SEED accuracy on FTF-BENCH under different decoding temperatures.

Temperature	Single-Answer	Multiple-Answer	Unanswerable	Average
0.0	78.66	76.99	84.05	80.59
0.2	88.16	80.57	73.61	81.70
0.4	79.41	80.96	62.28	72.21
0.6	81.10	79.10	61.37	71.92
0.8	90.24	82.89	77.55	82.65
1.0	89.47	82.29	77.81	82.38

K ANALYSIS OF THE 10% EVALUATION SUBSET

In the RL experiments, we randomly sample 90% of FTF-Bench for training and hold out the remaining 10% for evaluation, and the same subset is used for all models. To check whether this subset introduces bias, we evaluate all baseline models on both the full benchmark and this 10% subset and compare the results in Table 16.

L ADDITIONAL ANALYSIS ON DECODING TEMPERATURE

We ran an additional experiment on DOUBAO-1.6-SEED with different decoding temperatures, including a fully deterministic setting with temperature 0.0. Table 17 reports the accuracy on FTF-BENCH for temperatures between 0.0 and 1.0.

The overall accuracy varies only moderately across temperatures, indicating that the tendency to refuse is stable.

We further measure the agreement between outputs at temperature 0 and outputs at other temperatures on an instance-by-instance basis. Results are shown in Table 18.

This high consistency shows that the refusal behavior and the error patterns we analyze are stable. We have included the results above in Appendix L of the revised version of our paper.

1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511

Table 18: Instance-level agreement between temperature 0.0 and higher temperatures for DOUBAO-1.6-SEED.

Temperature	Single-Answer	Multiple-Answer	Unanswerable	Average
0.2	79.19	75.24	80.89	77.79
0.4	74.00	73.33	67.16	70.72
0.6	73.55	73.47	66.55	70.47
0.8	79.11	75.22	78.34	77.74
1.0	78.84	75.54	79.21	78.13