MUSE: IMPROVING MULTIMODAL LARGE LANGUAGE MODELS WITH AGENTIC HARNESSES

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite rapid progress, multimodal large language models (MLLMs) still fail on tasks that humans solve effortlessly, such as navigating a grid maze or selecting the correct puzzle piece. Rather than retraining the model, we ask a complementary question: how much capability can be elicited from a frozen MLLM purely by improving the execution scaffold around it? We introduce MUSE, a multimodal unified structured execution harness that wraps off-the-shelf MLLMs with composable modules for task representation, visual processing, perception tool use, structured parsing, and deterministic verification, without any model retraining. We evaluate MUSE across diverse benchmarks spanning visual spatial planning, visual perception, multimodal reasoning, and fine-grained visual discrimination, using multiple state-of-the-art MLLMs. MUSE delivers consistent gains over the bare model in all settings, with the largest jumps on challenging instances. Further analysis reveals that many MLLM failures arise from harness-level shortcomings rather than fundamental model deficits. These findings highlight the agentic multimodal harness as a critical yet underexplored design dimension, offering an orthogonal avenue for improving MLLMs beyond model-centric optimization.

1 INTRODUCTION

Multimodal large language models (MLLMs) (Liu et al., 2023; Hurst et al., 2024; Achiam et al., 2023; Anthropic, 2024) extend large language models (LLMs) Grattafiori et al. (2024); Bai et al. (2023); Brown et al. (2020) with visual perception, unifying image understanding, dialogue, and reasoning within a single interface and achieving impressive performance on various vision tasks. Despite this rapid progress, MLLMs still exhibit striking failures on tasks that humans solve almost instantaneously. Examples include identifying which puzzle piece completes a missing corner (Fu et al., 2024), counting non-overlapping objects (Tong et al., 2024), or planning a valid path through a simple grid maze (Wu et al., 2025b). These failures persist despite the strong general capabilities of modern MLLMs and pose a major challenge for applications requiring reliable visual cues, such as vision reasoning (Dong et al., 2026; Yang et al., 2025), multimodal understanding (Yue et al., 2025), and embodied control (Liu et al., 2025; Zhang et al., 2026).

A dominant direction in recent research has been to improve the model itself. Existing approaches introduce synthetic visual reasoning data (Ray et al., 2024; Wang et al., 2025; Yang et al., 2025), visual chain-of-thought supervision (Zhao et al., 2025; Zhang et al., 2023), or reinforcement-learning-based self-verification mechanisms (Wu et al., 2025a; Zhao et al., 2026). While effective, these approaches fundamentally rely on model-centric optimization, requiring either access to model parameters or expensive retraining pipelines. In practice, however, the strongest commercial MLLMs are closed-source, and even open-weight models evolve rapidly, making specialized training pipelines costly to maintain and quickly outdated as new checkpoints emerge. Therefore, we argue for a complementary and orthogonal direction: *improving the harness surrounding the model rather than the model itself* (Meng et al., 2026; Tang et al., 2026; Ning et al., 2026; Li et al., 2026).

Recent studies on LLM agents suggest that many practical failures arise not from insufficient pretraining, but from weaknesses in the surrounding execution framework, or "exoskeleton", which governs information organization, tool invocation, output verification, and retry strategies (Wang et al., 2026; Lin et al., 2026a;b). Consequently, a growing body of work has shifted attention from model optimization to harness engineering. For example, Meta-Harness (Lee et al., 2026)

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

treats harness engineering as an optimization problem, where an LLM iteratively improves its own execution logic by analyzing past execution traces and rewriting the surrounding harness code. AutoHarness (Lou et al., 2026) uses LLMs to synthesize and refine executable harnesses using environment feedback with tree-based code search. Life-Harness (Xu et al., 2026) introduces a model-agnostic runtime interface that adapts the harness rather than the model, converting recurring agent-environment failures into reusable interventions. *However, harness strategies for MLLMs remain largely unexplored, despite identical economic incentives.*

Transferring the harness paradigm from text-only tasks to multimodal settings is non-trivial: visual inputs introduce perceptual brittleness, outputs might require verification against geometric or spatial structure, and tool invocation must operate over visual context rather than disembodied text. To address these challenges, we introduce MUSE, a **multimodal unified structured execution harness** for MLLMs. Rather than modifying the model itself, MUSE treats the surrounding execution framework as a first-class optimization target. It wraps a frozen MLLM with a modular inference pipeline that covers the full inference path from perception and tool use, through structured parsing and deterministic verification, to verifier-guided repair, all under a unified task-level abstraction. Each component is explicitly designed to address specific failure modes observed in multimodal tasks. Importantly, MUSE operates under a strict black-box setting: the underlying model is never fine-tuned or altered, and all performance gains arise purely from harness-level improvements.

We evaluate the proposed MUSE on four heterogeneous benchmarks that stress different aspects of multimodal applications: visual spatial planning (VSP-Grid (Wu et al., 2025b)), visual perception (BLINK-Jigsaw (Fu et al., 2024)), multimodal reasoning (CoMT (Cheng et al., 2025)), and fine-grained visual discrimination (the Word Search sub-task of TIR-Bench (Li et al., 2025b)). Across four frontier models (GPT-4o (Hurst et al., 2024), GPT-5.4 (Singh et al., 2026), Claude Haiku 4.5 (Anthropic, 2025), and Claude Opus 4.7 (Anthropic, 2026)), MUSE consistently improves performance without any model tuning. The improvements are particularly pronounced on challenging instances, where repair fixes a large fraction of otherwise incorrect predictions. Detailed analyses attribute these gains to corrections of diverse failure modes, including perceptual errors, planning inconsistencies, and output-format violations, while also exposing remaining bottlenecks such as persistent mis-localization in dense visual scenes. Our main contributions are summarized as follows:

- We propose MUSE, a unified agentic harness that improves frozen MLLMs by optimizing the execution scaffold rather than the model itself. To the best of our knowledge, *MUSE is the first unified multimodal harness designed specifically for frozen MLLMs on visual-centric tasks*.
- We introduce a modular inference pipeline that unifies task representation, visual preprocessing, structured parsing, deterministic verification, and iterative repair under a single abstraction.
- We evaluate MUSE on multiple heterogeneous multimodal benchmarks across several state-of-the-art MLLMs, and demonstrate consistent improvements, with the largest gains on challenging instances, e.g., GPT-4o on Word Search improves from 3% to 21%. Our analysis shows that many MLLM failures arise from harness-level shortcomings rather than fundamental model deficits.

2 Related Work

MLLMs. MLLMs extend the autoregressive language-modeling paradigm to multimodal inputs by aligning a pretrained vision encoder with an LLM backbone, followed by large-scale multimodal instruction tuning (Liu et al., 2023; Hurst et al., 2024; Anthropic, 2024). They ingest interleaved text and image inputs and produce fluent natural-language responses, achieving competitive performance on various vision tasks (Li et al., 2025a; Liu et al., 2025). Yet despite this progress, MLLMs remain unreliable on tasks that humans solve effortlessly, such as object counting (Tong et al., 2024), spatial planning (Wu et al., 2025b), puzzle piece completion (Fu et al., 2024), or reading dense character grids (Li et al., 2025b). One sub-line of MLLM research attempts to compensate for these gaps by augmenting the model with external tools. For example, ViperGPT (Surís et al., 2023), and VisProg (Gupta & Kembhavi, 2023) decompose complex queries into calls to specialized vision modules, and Set-of-Mark prompting (Yang et al., 2023) uses SAM (Kirillov et al., 2023) to annotate image regions with symbolic tags fed back to the MLLM. We treat the frozen MLLM as the perception engine and focus optimization on the harness layer, where MUSE can selectively invoke perception tools through a registry to enhance visual reasoning and task execution.

LLM Agents. LLM-based agents extend single-shot generation into multi-step decision processes that combine reasoning, tool invocation, and feedback in a structured loop, with ReAct (Yao et al., 2023b) popularizing the reasoning-action interleaving template and Toolformer (Schick et al., 2023) showing that an LLM can learn when to invoke external APIs. A prominent sub-line explores iterative self-improvement, where the model critiques its own outputs. For example, Self-Refine (Madaan et al., 2023) uses the model to generate feedback and revise its own answer, Reflexion (Shinn et al., 2023) accumulates episodic lessons across attempts, and Tree-of-Thoughts (Yao et al., 2023a) explores multiple reasoning branches with self-evaluation-based pruning. MUSE is positioned within the agent family but departs from self-critique by employing an externalized verifier whenever the task admits one. This helps decouple generation from evaluation, providing more reliable refinement signals and mitigating the limitations of self-critique.

Harness Optimization. Treating the software scaffolding around a frozen LLM as the optimization variable has emerged as a productive alternative to model training itself. Early work focuses on prompt optimization (Yang et al., 2024; Agrawal et al., 2025), and on text-based credit assignment for entire LLM systems (Yuksekgonul et al., 2024; Lee et al., 2025). A recent line of work treats the *harness*, the runtime scaffold that governs input construction, tool invocation, output verification, and retry strategies, as an explicit object of optimization rather than a static execution layer (OpenAI, 2026; Young, 2025a; Bölük, 2026; Young, 2025b; Böckeler, 2026). For example, Meta-Harness (Lee et al., 2026) formulates harness design as an outer-loop coding-agent optimization problem. AutoHarness (Lou et al., 2026) automatically synthesizes and refines executable harnesses using environment feedback, and Life-Harness (Xu et al., 2026) introduces reusable runtime interventions that adapt agents to changing environments without finetuning. Despite this progress, all the methods above are evaluated exclusively on text-only tasks, optimizing prompts, pipelines, or harnesses over purely textual artifacts. In contrast, we consider a multimodal setting in which the harness must reason over both textual and *visual* signals.

3 The Unified Agentic Harness

3.1 Problem Statement

Following the harness-optimization framework of (Lee et al., 2026), we model the harness as a stateful program H that wraps a frozen MLLM π_θ. Given a task instance $x \sim \mathcal{X}$, executing H with π_θ induces a rollout trajectory $\tau \sim p_{\pi_\theta}(H, x)$, consisting of an initial model response followed by at most K verifier-guided repair steps. Each trajectory is evaluated by a task-specific deterministic reward $r(\tau, x)$, indicating success. With π_θ fixed, the objective is to optimize the harness:

$$H^\star \;=\; \arg\max_{H \in \mathcal{H}} \; \mathbb{E}_{x\sim\mathcal{X},\, \tau\sim p_{\pi_\theta}(H,x)}\big[r(\tau, x)\big], \quad \text{subject to} \quad \mathbb{E}[c(\tau)] \le \kappa, \tag{1}$$

where $c(\tau)$ denotes the execution cost and κ is a per-task budget.

Two distinctions are critical for our setting. First, each x may include both text and images. As a result, the harness space $\mathcal{H}$ must encompass not only textual operations but also image-dependent components, such as preprocessing, perceptual tool invocation, and image-grounded verification. Second, unlike Meta-Harness, which searches over $\mathcal{H}$ using an outer-loop optimization procedure, we pursue a complementary design-driven approach. Specifically, we construct a single harness $H \in \mathcal{H}$ by systematically addressing empirically observed failure modes in prior multimodal benchmarks (Wu et al., 2025b; Fu et al., 2024; Cheng et al., 2025; Li et al., 2025b).

3.2 Overview

MUSE is a multi-turn controller built around a frozen MLLM π_θ, which is accessed solely through a black-box interface $\pi_\theta(\mathcal{M}) \mapsto$ *text* over message lists $\mathcal{M}$. Given a benchmark instance x (consisting of a question, associated images, and optional metadata), MUSE executes a finite interaction transcript $(m_0, o_0, p_0, v_0), \ldots, (m_k, o_k, p_k, v_k)$ with $k \le K$, where m_i denotes the constructed message context, o_i the raw model output, p_i a task-specific candidate parsed from o_i, and v_i the verifier's verdict on (o_i, p_i). At termination, MUSE returns a structured result record containing the final success indicator, parsed outputs, intermediate textual traces, raw model responses, optional reference annotations, and execution metadata. Importantly, MUSE operates without modifying π_θ: all

Algorithm 1 MUSE: a multimodal unified structured execution harness for MLLMs.

Require: Frozen MLLM π_θ, task instance x, max repairs $K \geq 0$
Ensure: An APPROACHRESULT indicating success/failure and the trajectory

$r \leftarrow$ TASKHOOKS(x).TOOLS ▷ Tool requirements from the task-type hook registry
$\mathcal{T} \leftarrow$ RUNREQUIREDTOOLS(r, x) ▷ Run each declared perception tool on x's image
$m_0 \leftarrow$ BUILDINITIALPROMPT$(x, \mathcal{T})$ ▷ Optionally folds tool outputs into the prompt
$o_0 \leftarrow \pi_\theta($PACK$(x, m_0))$ ▷ Single call to the frozen MLLM
$p_0 \leftarrow$ PARSE(x, o_0) ▷ Task-specific parser extracts the structured candidate
$v_0 \leftarrow$ VERIFY$(x, o_0, p_0, r, \mathcal{T})$ ▷ Outcome policy plus perception-evidence gate
if v_0.ok **then return** APPROACHRESULT(SUCCESS=TRUE, ...) ▷ Early-stop on first verifier acceptance
end if
$\mathcal{F} \leftarrow [(o_0, v_0.\text{detail})]$ ▷ Initialize the failure transcript
for $i = 1$ **to** K **do** ▷ Up to K verifier-guided repair turns
  $m_i \leftarrow$ BUILDREPAIRPROMPT$(x, m_0, \mathcal{F}, i, K)$ ▷ Adds the compressed failure summary
  $o_i \leftarrow \pi_\theta($PACK$(x, m_i))$ ▷ Re-query the frozen MLLM on the extended context
  $p_i \leftarrow$ PARSE(x, o_i) ▷ Re-parse the new response
  $v_i \leftarrow$ VERIFY$(x, o_i, p_i, r, \mathcal{T})$ ▷ Re-verify under the same outcome policy
  if v_i.ok **then return** APPROACHRESULT(SUCCESS=TRUE, ...) ▷ Loop short-circuits for acceptance
  end if
  Append $(o_i, v_i.\text{detail})$ to $\mathcal{F}$ ▷ Otherwise extend the failure transcript
end for
return APPROACHRESULT(SUCCESS=FALSE, ...) ▷ Budget exhausted without acceptance

adaptation is realized through the harness H. Algorithm 1 sketches the overall workflow of MUSE. The remainder of this section introduces the key components of MUSE.

3.3 CORE COMPONENTS

Unified Task Representation. To avoid dataset-specific branching in the execution loop, we represent each benchmark instance as a unified record x produced by a per-dataset adapter. Each adapter normalizes its source's on-disk format into the same record schema, so the executor sees a uniform iterator regardless of source. Concretely, the executor consumes a single immutable record that encapsulates the task type, shared instructions, ordered image paths, optional structured inputs, and, when available, ground-truth annotations. Task-specific behavior is handled separately through the abstractions described next.

Unified Task Hooks. Task hooks provide the concrete behavioral interface for each task type, binding the unified record schema to task-specific parsing, prompting, and repair logic. For each task type, we define an immutable task hook record with four components: i) answer extraction, which maps raw model outputs to the structured representation consumed by the outcome policy (e.g., partial JSON recovery, or direct QA pass-through); ii) instruction initialization, which optionally overrides the default first-turn prompt with task-specific instructions; iii) repair policies, which determine how verifier failures are translated into feedback for subsequent refinement; and iv) tool requirements, which declare the perception tools the harness might invoke for this task. This abstraction decouples task-specific logic from the execution loop, so that supporting a new benchmark reduces to registering an adapter and a task hook, without modifying the core executor.

Instruction Interface. The instruction interface constructs both the initial user message and subsequent repair prompts by composing dataset text, optional tool outputs, and references to preprocessed visual artifacts under the task-specific template registered in the task hook. For repair turns, the interface appends an ordered transcript of prior model outputs and their verifier feedback to the original instruction, formatting each entry according to the repair-policy hook for the current task type. By separating message assembly from template content and failure phrasing, the interface allows task-specific prompt wording to be updated without changing the assembly logic.

Outcome Policy. The outcome policy is the authority for determining solution correctness, externalizing task semantics from the model into the harness. Verification is implemented as a family of task-specific verifiers, each encoding a notion of correctness aligned with the response schema, and optionally augmented with tool-auxiliary checks when annotations alone are insufficient to resolve perceptual ambiguity. Each rejection is categorized along a coarse taxonomy capturing distinct failure modes, including format violations, task-logic errors, and evidence gaps. The repair policy uses this taxonomy to select task-appropriate feedback phrasing for subsequent repair turns. See Appendix A.2 for the implementation details.

Simulator-Backed Verification. For tasks that admit executable environments, we instantiate verification via explicit simulators that provide grounded feedback. For example, on navigation benchmarks, the verifier evaluates a candidate action sequence by simulating its execution. Given the instance-specific transition dynamics, each action sequence induces a unique trajectory, and success is defined as reaching a designated goal state. Failure cases yield localized signals (e.g., collisions, invalid transitions, or premature termination). On jigsaw-style piece-fitting tasks, the verifier stitches each candidate piece into the target image and scores the resulting seams by the aggregate L_1 pixel mismatch, allowing it to differentiate candidates that are visually similar but exhibit different boundary consistency.

Perception Tool Registry. Beyond the core pipeline above, MUSE exposes an extensible registry of pluggable perception tools that can be invoked deterministically by the harness whenever a task benefits from auxiliary visual processing. Each tool is a pure function from an image to a structured payload (text, embeddings, or another image), and is referenced by name from each task type's per-task tool requirements. The executor runs only the tools that the current task declares. Built-in tools include geometric transforms (e.g., zoom, rotation), OCR, and embedding utilities. For example, on the Word Search task (Section 4.4), we register OCR as the perception tool and demonstrate that it composes cleanly with verifier-guided repair to address a perceptual bottleneck the single-shot baseline cannot resolve.

Repair Loop. When the verifier rejects the initial candidate, the harness enters a repair loop: each round re-invokes π_θ with the original message context augmented by the running failure transcript, re-parses the structured output, and re-verifies it until either acceptance or the repair budget of K additional turns is exhausted. The verifier decides when the loop terminates: the moment it returns success, the loop exits immediately, so most easy instances incur only the cost of the initial call. The transcript is formatted to make failure modes explicit: attempts are sequentially indexed, model outputs are quoted, and simulator-derived feedback is clearly distinguished from syntactic or formatting errors. On top of the raw transcript, the harness emits a compact failure summary that includes the count of each error type, the most recent verifier signal, and a repeated-output flag derived from a content fingerprint of the previous attempt. This prevents the model from cycling on the same wrong commitment when feedback is intentionally short.

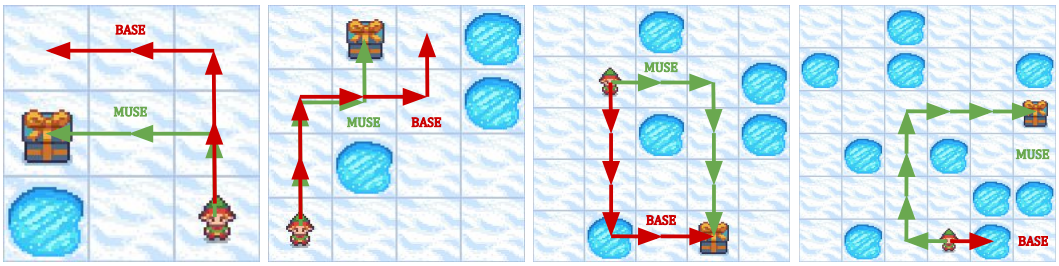
4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate MUSE on four heterogeneous vision benchmarks: VSP-Grid (Wu et al., 2025b), BLINK-Jigsaw (Fu et al., 2024), CoMT (Cheng et al., 2025), and TIR-Bench (Li et al., 2025b), which together cover visual spatial planning, visual perception, multimodal reasoning, and fine-grained visual discrimination. VSP-Grid presents a rendered grid environment containing a player, a goal, and hazardous regions. The task requires the model to generate a sequence of moves that reaches the goal without entering hazards. Predicted sequences are parsed and executed in a simulator, and success is determined by whether the goal is reached; optimality is additionally assessed via path length. We report results across four difficulty levels, corresponding to $n \times n$ grids with $n \in \{3, 4, 5, 6\}$. BLINK-Jigsaw provides three images: an incomplete puzzle with a missing region and two candidate patches. The task is to choose the correct patch that correctly completes the image and output the answer in a prescribed format. For CoMT, we use the creation split, which consists of multimodal geometry reasoning problems, each including a natural-language question, multiple-choice options, and one associated image. The model must identify the single correct

Table 1: Performance on the VSP-Grid task. V: Valid (%), plans reaching the goal; O: Optimal (%), plans with shortest-path length; Avg.: average across task levels.

Model	Approach	Level 3		Level 4		Level 5		Level 6		Avg.	
		V	O	V	O	V	O	V	O	V	O
GPT-4o	BASE	0.65	0.41	0.46	0.34	0.39	0.26	0.23	0.16	0.43	0.29
	MUSE	0.89	0.55	0.61	0.42	0.55	0.39	0.40	0.26	0.61	0.41
GPT-5.4	BASE	0.50	0.42	0.38	0.20	0.34	0.24	0.23	0.11	0.36	0.24
	MUSE	0.66	0.53	0.67	0.39	0.65	0.43	0.37	0.18	0.59	0.38
Haiku 4.5	BASE	0.47	0.43	0.37	0.37	0.50	0.49	0.31	0.28	0.41	0.39
	MUSE	0.75	0.70	0.65	0.61	0.67	0.64	0.54	0.47	0.65	0.61
Opus 4.7	BASE	0.82	0.66	0.74	0.69	0.85	0.79	0.59	0.58	0.75	0.68
	MUSE	0.97	0.77	0.94	0.90	0.95	0.86	0.81	0.78	0.92	0.83



(a) GPT-4o / Level 3 (b) GPT-5.4 / Level 4 (c) Haiku 4.5 / Level 5 (d) Opus 4.7 / Level 6

Figure 1: Trajectory comparison between BASE (red) and MUSE (green) on the VSP-Grid task. Each panel shows one MLLM at one difficulty level. MUSE produces shorter, more goal-directed paths than BASE.

option. We further evaluate the Word Search split of TIR-Bench, which provides a dense grid image populated with many similar glyphs (Chinese characters, Latin letters, or digits) and asks the model either to count the occurrences of a designated target glyph, or to localize a target glyph by its row and column indices.

Models. We evaluate MUSE using four frozen MLLMs: OpenAI GPT-4o (Hurst et al., 2024), OpenAI GPT-5.4 (Singh et al., 2026), Anthropic Claude Haiku 4.5 (Anthropic, 2025), and Anthropic Claude Opus 4.7 (Anthropic, 2026). We compare two execution modes under a controlled setting that shares identical testing sets: i) BASE, which performs a single MLLM call followed by deterministic parsing and verification, serving as the zero-shot baseline used in prior benchmarks; ii) MUSE, our full agentic harness (Algorithm 1), which uses the same initial prompt as BASE but enables verifier-guided repair with up to K additional iterations when the outcome policy rejects the structured output.

Metrics. For VSP-Grid, we evaluate performance using two metrics: Valid Path, defined as the fraction of instances where the simulated trajectory reaches the goal without traversing hazardous cells, and Optimal Length, defined as the fraction of instances whose trajectory reaches the goal via a shortest path on the parsed grid. For BLINK-Jigsaw, we report selection accuracy against the dataset-provided ground-truth label. For CoMT, we report accuracy based on canonicalized answer matching. For Word Search, we report exact-match accuracy against the dataset-provided answer. A single shared verifier handles both the counting and localization sub-types, judging a prediction correct only when it matches the reference in both type and value. In addition, we report the total number of MLLM calls, and for MUSE, the number of instances that trigger at least one repair turn.

4.2 MUSE for Visual Spatial Planning

Table 1 reports the valid-path rate and optimal-length rate for all models under BASE and MUSE. Across all models and grid sizes, MUSE consistently improves valid-path accuracy over BASE, with

Table 2: Results comparison across CoMT, Word Search, and BLINK-Jigsaw. "Valid" indicates the number of correct answers; "Calls" counts every model call across all problems; "#Rep." counts problems on which the harness invoked at least one repair turn. Δ indicates the improvement in Valid.

		CoMT				Word Search				BLINK-Jigsaw			
Model	Approach	Valid	Calls	#Rep.	Δ	Valid	Calls	#Rep.	Δ	Valid	Calls	#Rep.	Δ
GPT-4o	BASE	101	200	–	–	3	100	–	–	112	150	–	–
	MUSE	175	375	99	+74	21	352	88	+18	121	249	38	+9
GPT-5.4	BASE	121	200	–	–	30	100	–	–	118	150	–	–
	MUSE	196	306	79	+75	62	246	57	+32	144	217	35	+26
Haiku 4.5	BASE	144	200	–	–	5	100	–	–	93	150	–	–
	MUSE	182	303	51	+38	35	315	78	+30	149	220	54	+56
Opus 4.7	BASE	161	200	–	–	23	100	–	–	110	150	–	–
	MUSE	197	249	40	+36	59	251	59	+36	146	224	40	+36

substantial gains at every difficulty level. Specifically, GPT-4o improves by +24 points at Level 3 and +17 at Level 6; GPT-5.4 achieves its largest gain at Level 5 with +31 points; Claude Haiku 4.5 improves by +28 points at Level 3. Even for Claude Opus 4.7, which is already strong under BASE, MUSE further increases performance to 97% valid at Level 3 and 81% on the hardest 6×6 grids.

The optimal-length results reveal a more nuanced pattern. For Claude Haiku 4.5, MUSE achieves 70% optimal at Level 3 and averages 61% across all levels, indicating that most repaired trajectories are not only valid but also near-optimal once perception is corrected. In contrast, GPT-5.4 exhibits a larger validity–optimality gap at Levels 4 to 6, suggesting that while the harness effectively filters invalid plans, it cannot guide the model toward shorter trajectories within the limited repair budget K.

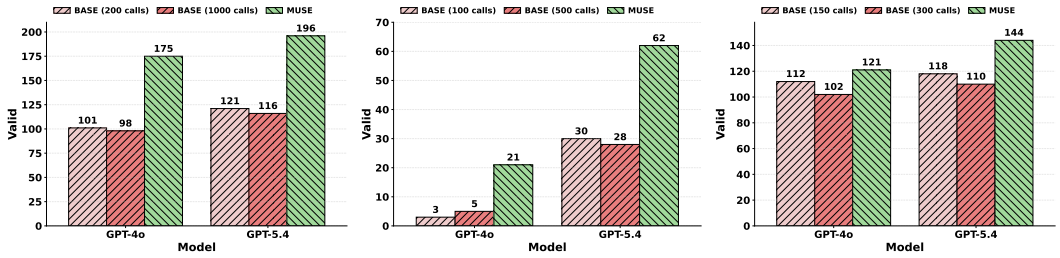
Figure 1 provides qualitative comparisons of trajectories generated by BASE and MUSE across models and difficulty levels. A consistent pattern emerges: MUSE produces more direct and goal-oriented paths, whereas BASE often exhibits redundant movements, detours, or oscillatory behaviors (see Appendix B.1 for additional results). Importantly, these gains are consistent across models, suggesting that the improvements arise from the harness design rather than model-specific factors. BASE errors are heterogeneous, ranging from hazard violations and goal-reaching failures to perceptual mislocalization of obstacles, whereas MUSE yields more reliable and efficient action sequences.

4.3 MUSE for Multimodal Reasoning

The CoMT task requires multi-step mathematical reasoning over structured multimodal inputs. As shown in Table 2, across all models, MUSE yields substantial improvements over BASE, with gains of +74 (GPT-4o), +75 (GPT-5.4), +38 (Claude Haiku 4.5), and +36 (Claude Opus 4.7) in valid answers. These large margins suggest that CoMT is highly sensitive to reasoning quality, where small errors in intermediate steps can propagate to incorrect final predictions. Detailed case studies (Appendix B.2) reveal consistent failure patterns under BASE, including incomplete reasoning chains, inconsistencies between intermediate reasoning and final answers, and parsing failures (e.g., malformed outputs despite correct reasoning). MUSE addresses these failures through iterative, verifier-guided repair, indicating that many CoMT errors arise from correctable reasoning or execution issues rather than fundamental capability limitations. Additional qualitative comparisons are provided in Appendix B.2.

4.4 MUSE for Fine-Grained Visual Discrimination

In the Word Search task, the model is asked to locate a designated character within a dense two-dimensional grid populated by many visually similar glyphs and respond with its $[\text{row}, \text{col}]$ coordinates. This task stresses fine-grained visual discrimination: the visual scene is highly repetitive, the target glyph occupies one cell among hundreds, and a correct answer requires per-cell attention rather than scene-level summarization. As shown in Table 2, every frozen MLLM struggles under BASE, confirming that dense-grid discrimination is a regime where current MLLMs are perceptually unreliable. For this task, the harness registers OCR as a perception tool (Section 3.3) and folds its output into the initial prompt. With this OCR evidence and verifier-guided repair, MUSE consistently



(a) CoMT task (b) Word Search task (c) BLINK-Jigsaw task

Figure 2: Compute-matched control across three benchmarks. Results on GPT-4o and GPT-5.4 comparing the original BASE configuration (light red), a compute-matched self-consistency variant with additional model calls (dark red), and MUSE (green). Bars denote the number of correct answers. Merely increasing the number of model calls provides little benefit and can even reduce performance. In contrast, MUSE consistently delivers substantial gains while using fewer model calls than the self-consistency baseline, indicating that its improvements arise from structured execution rather than a larger inference budget.

lifts every model. The repair counts indicate that the verifier rejects unreliable initial guesses on the majority of instances. The model, prompted to re-examine the image with OCR evidence in mind, often shifts to a more careful localization or recount. Relative gains are especially large for GPT-4o and Claude Haiku 4.5, both of which improve their valid counts by approximately a factor of seven. This pattern suggests that the bottleneck on Word Search is genuine perceptual difficulty rather than format errors or short-form parsing, and that combining a perception tool with verifier-guided repair adds value even when the underlying MLLM has very limited initial competence on a task. Additional qualitative comparisons are provided in Appendix B.3.

4.5 MUSE FOR VISUAL PERCEPTION

In the BLINK-Jigsaw task, the objective is to identify the correct patch to complete an incomplete image based on visual context. As shown in Table 2, MUSE consistently improves over BASE across all evaluated models, indicating that many initial prediction errors are correctable through verifier-guided repair. Qualitative inspection reveals a common failure mode: BASE frequently selects patches with poor seam continuity despite plausible semantic content. Such errors are effectively detected by the seam-stitch verifier via pixel-level L_1 mismatch along patch boundaries, providing a reliable signal for repair. While the magnitude of the gains varies across models, the improvements are uniformly positive, highlighting the generality of the proposed harness. Additional qualitative comparisons are provided in Appendix B.4.

4.6 ABLATION STUDY

Compute-matched control. One might argue that the gains of MUSE arise simply from issuing more model calls than BASE, rather than from its structured execution pipeline. To rule out this explanation, we grant BASE a call budget matched to that of MUSE via a self-consistency baseline that expends the additional calls without any harness structure. Specifically, we query the model multiple times independently (five times for CoMT and Word Search, and twice for BLINK-Jigsaw) and return the majority-voted answer. As shown in Figure 2, increasing the call budget alone yields only marginal changes in performance. On CoMT, validity changes from 101 to 98 for GPT-4o and from 121 to 116 for GPT-5.4. On Word Search, GPT-4o improves only slightly (from 3 to 5), while GPT-5.4 decreases from 30 to 28. On BLINK-Jigsaw, performance also changes little, dropping from 112 to 102 for GPT-4o and from 118 to 110 for GPT-5.4. Overall, simply increasing the number of model calls provides little benefit and does not reliably improve task performance.

In contrast, MUSE consistently delivers substantial gains across benchmarks and models, improving validity from 101 to 175 and from 121 to 196 on CoMT, from 3 to 21 and from 30 to 62 on Word Search, and from 112 to 121 and from 118 to 144 on BLINK-Jigsaw. Notably, these gains are achieved with fewer model calls than the compute-matched self-consistency baseline. These results

suggest that the benefits of MUSE do not stem from a larger inference budget. Rather, they arise from the structured design of the harness, which converts additional computation into meaningful improvements in decision quality.

Effect of repair budget K. All headline experiments use a maximum of $K = 3$ repair turns. However, the relationship between performance and K is not necessarily monotonic. To examine this effect, we evaluate Claude Opus 4.7 on VSP Level 6 under varying repair budgets, $K \in \{2, 3, 5\}$. We focus on Level 6 because it is the most challenging setting and remains difficult even for the strongest model in our evaluation suite. Figure 3 reports both validity and optimality as functions of K. Increasing the budget from $K = 2$ to $K = 3$ yields the largest gains, with validity improving from 0.69 to 0.81 and optimality from 0.67 to 0.78. This indicates that a substantial fraction of instances require at least three repair iterations to converge to correct or near-optimal solutions. In contrast, further increasing the repair budget from $K = 3$ to $K = 5$ leads to lower validity (0.78) and optimality (0.71), suggesting diminishing returns and potential over-correction at larger budgets. Notably, the drop is more pronounced for optimality (-7 points) than validity (-3 points), suggesting that additional repair turns are particularly prone to converting near-optimal trajectories into valid but suboptimal ones. We therefore adopt $K = 3$ as the default setting, as it provides a favorable trade-off between accuracy and computational cost while avoiding over-correction effects observed at larger budgets.

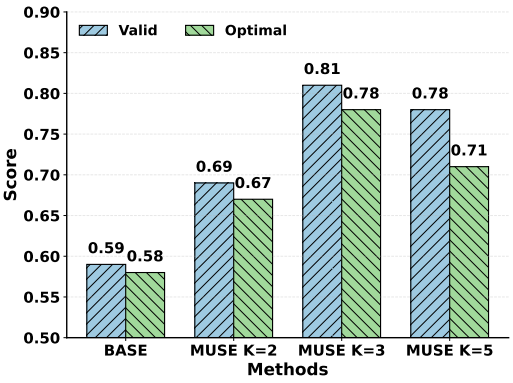


Figure 3: Validity & optimality vs. budget K.

Failure-mode analysis. We analyze performance on the hardest grid setting (6×6) using Claude Opus 4.7. Under BASE, the model fails on 41 instances, with the majority due to hazard violations (37 cases) and only a small fraction due to goal-reaching failures (4 cases). Moreover, failed trajectories are typically longer than the shortest valid paths, indicating inefficient planning behavior. Applying MUSE improves validity from 59% to 81%, with repair triggered on 35 instances. Relative to BASE, MUSE corrects 23 of the 41 initial failures, including 19 hazard violations and 4 failures to reach the goal. This indicates that goal-reaching failures are correctable, while hazard-related errors are more persistent. In particular, only about half of the hazard violations ($19/37$) are corrected within the repair budget $K = 3$. This suggests that, at higher difficulty levels, the dominant bottleneck shifts from high-level planning errors to low-level perceptual inaccuracies, specifically the mis-localization of hazards in the environment.

5 Conclusion

We presented MUSE, a multimodal unified structured execution harness that improves the capabilities of frozen MLLMs by optimizing the execution scaffold rather than the model itself. By decomposing inference into modular components, MUSE transforms single-pass generation into a structured and iterative decision-making process tailored to visual-centric tasks. Across a diverse set of benchmarks spanning visual spatial planning, visual perception, multimodal reasoning, and fine-grained visual discrimination, MUSE consistently yields substantial performance gains. Our analysis shows that many failures stem from harness-level shortcomings rather than fundamental model deficits, establishing the multimodal harness as an optimization dimension complementary to model-centric scaling. As frontier MLLMs continue to scale and retraining becomes increasingly expensive, harness engineering offers a practical avenue for improving system performance. More broadly, MUSE advocates a shift in perspective: rather than focusing exclusively on building stronger models, we should also explore how to better structure, verify, and leverage the capabilities of existing ones. We hope this work encourages further research on multimodal harness design as a complementary pathway toward more capable, reliable, and efficient multimodal systems.

AI USE STATEMENT

We emphasize that the MLLMs evaluated in this paper (i.e., GPT-4o, GPT-5.4, Claude Haiku 4.5, and Claude Opus 4.7) are objects of study rather than authoring aids. Their use constitutes part of the subject matter of this work and is documented in Sections 3 and 4.1. We used generative AI tools as coding assistants. We did not use generative AI for research ideation; the design of the harness architecture or experimental protocol; literature discovery or related-work retrieval; or data generation, augmentation, or labeling. Disclosure items concerning theoretical derivations and proofs are not applicable, as this work contains no formal proofs. We also used generative AI for limited writing assistance, including polishing wording, improving grammar and readability, and refining captions. In all such cases, the underlying content, claims, and structure were authored by us. All AI-assisted outputs were reviewed by the authors. All benchmark instances and reference answers were taken without modification from the public releases of VSP-Grid (Wu et al., 2025b), BLINK-Jigsaw (Fu et al., 2024), CoMT (Cheng et al., 2025), and TIR-Bench (Li et al., 2025b); no benchmark data were generated using generative AI. Every AI-assisted sentence in the manuscript was reviewed and edited by the authors, and all numerical results, citations, and factual claims were independently verified against our experimental logs and the cited sources. We take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, personally identifiable information, or sensitive personal data, and therefore did not require IRB review. All experiments are conducted on four publicly released research benchmarks, VSP-Grid (Wu et al., 2025b), BLINK-Jigsaw (Fu et al., 2024), CoMT (Cheng et al., 2025), and TIR-Bench (Li et al., 2025b), which we use unmodified and strictly within their intended research use and license terms; we redistribute no benchmark content. The stimuli in these benchmarks are synthetic or non-personal (rendered grids, jigsaw patches, geometry diagrams, and glyph grids), so we do not foresee privacy risks arising from our evaluation. The MLLMs we evaluate are accessed as frozen black boxes through their providers' public APIs, in compliance with the applicable terms of service; we perform no fine-tuning, no parameter extraction, and no attempt to circumvent provider safeguards.

REPRODUCIBILITY STATEMENT

We will open-source the full MUSE implementation, including dataset adapters, task hooks, instruction templates, the perception tool registry, verifiers and simulators, and the scripts that reproduce all tables and figures.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*, 2025.

Anthropic. The claude 3 model family: Opus, sonnet, haiku. 2024. Technical Report.

Anthropic. Introducing claude haiku 4.5. 2025. https://www.anthropic.com/news/claude-haiku-4-5.

Anthropic. Introducing claude opus 4.7. 2026. https://www.anthropic.com/news/claude-opus-4-7.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Birgitta Böckeler. Harness engineering. April 2026. Accessed: 2026-04, https://martinfowler.com/articles/exploring-gen-ai/harness-engineering.html.

Can Bölük. I improved 15 llms at coding in one afternoon. only the harness changed. February 2026. Accessed: 2026-04, https://blog.can.ac/2026/02/12/the-harness-problem/.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pp. 1877–1901, 2020.

Zihui Cheng, Qiguang Chen, Jin Zhang, Hao Fei, Xiaocheng Feng, Wanxiang Che, Min Li, and Libo Qin. Comt: A novel benchmark for chain of multi-modal thought on large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Qihua Dong, Kuo Yang, Lin Ju, Handong Zhao, Yitian Zhang, Yizhou Wang, Huimin Zeng, Jianglin Lu, and Yun Fu. Ref-adv: Exploring MLLM visual reasoning in referring expression tasks. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pp. 148–166. Springer, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14953–14962, 2023.

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.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, 2023.

Yoonho Lee, Joseph Boen, and Chelsea Finn. Feedback descent: Open-ended text optimization via pairwise comparison. *arXiv preprint arXiv:2511.07919*, 2025.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-harness: End-to-end optimization of model harnesses. *arXiv preprint arXiv:2603.28052*, 2026.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. *arXiv preprint arXiv:2509.24251*, 2025a.

Junjie Li, Xi Xiao, Yunbei Zhang, Chen Liu, Lin Zhao, Xiaoying Liao, Yingrui Ji, Janet Wang, Jianyang Gu, Yingqiang Ge, Weijie Xu, Xi Fang, Xiang Xu, Tianchen Zhao, Youngeun Kim, Tianyang Wang, Jihun Hamm, Smita Krishnaswamy, Jun Huan, and Chandan Reddy. Agent harness engineering: A survey. 2026. https://openreview.net/pdf?id=eONq7FdiHa.

Ming Li, Jike Zhong, Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Yuxiang Lai, Wei Chen, Konstantinos Psounis, and Kaipeng Zhang. Tir-bench: A comprehensive benchmark for agentic thinking-with-images reasoning. *arXiv preprint arXiv:2511.01833*, 2025b.

Huawei Lin, Peng Li, Jie Song, Fuxin Jiang, and Tieying Zhang. Muse-autoskill: Self-evolving agents via skill creation, memory, management, and evaluation. *arXiv preprint arXiv:2605.27366*, 2026a.

Jiahang Lin, Shichun Liu, Chengjun Pan, Lizhi Lin, Shihan Dou, Zhiheng Xi, Xuanjing Huang, Hang Yan, Zhenhua Han, Tao Gui, and Yu-Gang Jiang. Agentic harness engineering: Observability-driven automatic evolution of coding-agent harnesses. *arXiv preprint arXiv:2604.25850*, 2026b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 2023.

Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Song XiXuan, Yifan Xu, Shudan Zhang, Hanyu Lai, Jiadai Sun, Xinyue Yang, Yu Yang, Zehan Qi, Shuntian Yao, Xueqiao Sun, Siyi Cheng, Qinkai Zheng, Hao Yu, Hanchen Zhang, Wenyi Hong, Ming Ding, Lihang Pan, Xiaotao Gu, Aohan Zeng, Zhengxiao Du, Chan Hee Song, Yu Su, Yuxiao Dong, and Jie Tang. Visualagentbench: Towards large multimodal models as visual foundation agents. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xinghua Lou, Miguel Lázaro-Gredilla, Antoine Dedieu, Carter Wendelken, Wolfgang Lehrach, and Kevin P. Murphy. Autoharness: improving llm agents by automatically synthesizing a code harness. *arXiv preprint arXiv:2603.03329*, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 2023.

Qianyu Meng, Yanan Wang, Liyi Chen, Yihang Li, Wei Wu, Wenyuan Jiang, Qimeng Wang, Chengqiang Lu, Yan Gao, Yi Wu, and Yao Hu. Agent harness for large language model agents: A survey. 2026. https://www.preprints.org/manuscript/202604.0428/v3.

Xuying Ning, Katherine Tieu, Dongqi Fu, Tianxin Wei, Zihao Li, Yuanchen Bei, Jiaru Zou, Mengting Ai, Zhining Liu, Ting-Wei Li, Lingjie Chen, Yanjun Zhao, Ke Yang, Bingxuan Li, Cheng Qian, Gaotang Li, Xiao Lin, Zhichen Zeng, Ruizhong Qiu, Sirui Chen, Yifan Sun, Xiyuan Yang, Ruida Wang, Rui Pan, Chenyuan Yang, Dylan Zhang, Liri Fang, Zikun Cui, Yang Cao, Pan Chen, Dorothy Sun, Ren Chen, Mahesh Srinivasan, Nipun Mathur, Yinglong Xia, Hong Li, Hong Yan, Pan Lu, Lingming Zhang, Tong Zhang, Hanghang Tong, and Jingrui He. Code as agent harness. *arXiv preprint arXiv:2605.18747*, 2026.

OpenAI. Harness engineering: Leveraging codex in an agent-first world. February 2026. Accessed: 2026-04, OpenAI Blog, https://openai.com/index/harness-engineering/.

Arijit Ray, Jiafei Duan, Ellis Brown, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A. Plummer, Ranjay Krishna, Kuo-Hao Zeng, and Kate Saenko. Sat: Dynamic spatial aptitude training for multimodal language models. *arXiv preprint arXiv:2412.07755*, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 2023.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2026.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11888–11898, 2023.

Xinyu Tang, Han Peng, Guoxin Chen, Yuze Shi, Zitao Su, Peiyu Liu, Wayne Xin Zhao, Yawen Li, and Zhe Xue. Agent systems with harness engineering. 2026. https://openreview.net/pdf?id=nM5tDHrQsx.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9568–9578, 2024.

Jianing Wang, Linsen Guo, Zhengyu Chen, Qi Guo, Hongyu Zang, Wenjie Shi, Haoxiang Ma, Xiangyu Xi, Xiaoyu Li, Wei Wang, and Xunliang Cai. Heavyskill: Heavy thinking as the inner skill in agentic harness. *arXiv preprint arXiv:2605.02396*, 2026.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond images and language. *arXiv preprint arXiv:2511.21395*, 2025.

Mingyuan Wu, Meitang Li, Jingcheng Yang, Jize Jiang, Kaizhuo Yan, Zhaoheng Li, Hanchao Yu, Minjia Zhang, and Klara Nahrstedt. Aha moment revisited: Are vlms truly capable of self verification in inference-time scaling? *arXiv preprint arXiv:2506.17417*, 2025a.

Qiucheng Wu, Handong Zhao, Michael Saxon, Trung Bui, William Yang Wang, Yang Zhang, and Shiyu Chang. Vsp: Diagnosing the dual challenges of perception and reasoning in spatial planning tasks for mllms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025b.

Tianshi Xu, Huifeng Wen, and Meng Li. Adapting the interface, not the model: Runtime harness adaptation for deterministic llm agents. *arXiv preprint arXiv:2605.22166*, 2026.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*, 2024.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. *arXiv preprint arXiv:2506.17218*, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023b.

Justin Young. Effective harnesses for long-running agents. *Anthropic Engineering Blog, Nov*, 2025a.

Justin Young. Effective harnesses for long-running agents. November 2025b. Anthropic Engineering Blog, https://www.anthropic.com/engineering/effective-harnesses-for-long-running-agents.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. MMMU-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 15134–15186, 2025.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024.

Haichao Zhang, Yijiang Li, Shwai He, Tushar Nagarajan, Mingfei Chen, Jianglin Lu, Ang Li, and Yun Fu. Thinkjepa: Empowering latent world models with large vision-language reasoning model. *arXiv preprint arXiv:2603.22281*, 2026.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

Chen Zhao, Zhuoran Wang, Haoyang Li, Shifeng Bao, Guanlin Li, Youhe Feng, Yang Li, Jie Tang, and Jing Zhang. Action draft and verify: A self-verifying framework for vision-language-action model. *arXiv preprint arXiv:2603.18091*, 2026.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

APPENDIX

A EXPERIMENTAL DETAILS

A.1 BENCHMARKS

The used benchmarks VSP-Grid, BLINK-Jigsaw, CoMT, and TIR-Bench highlight the heterogeneity in task formulations, input modalities, and expected outputs, which cover complementary capabilities of MLLMs, such as visual spatial planning, visual perception, multimodal reasoning, and fine-grained visual discrimination.

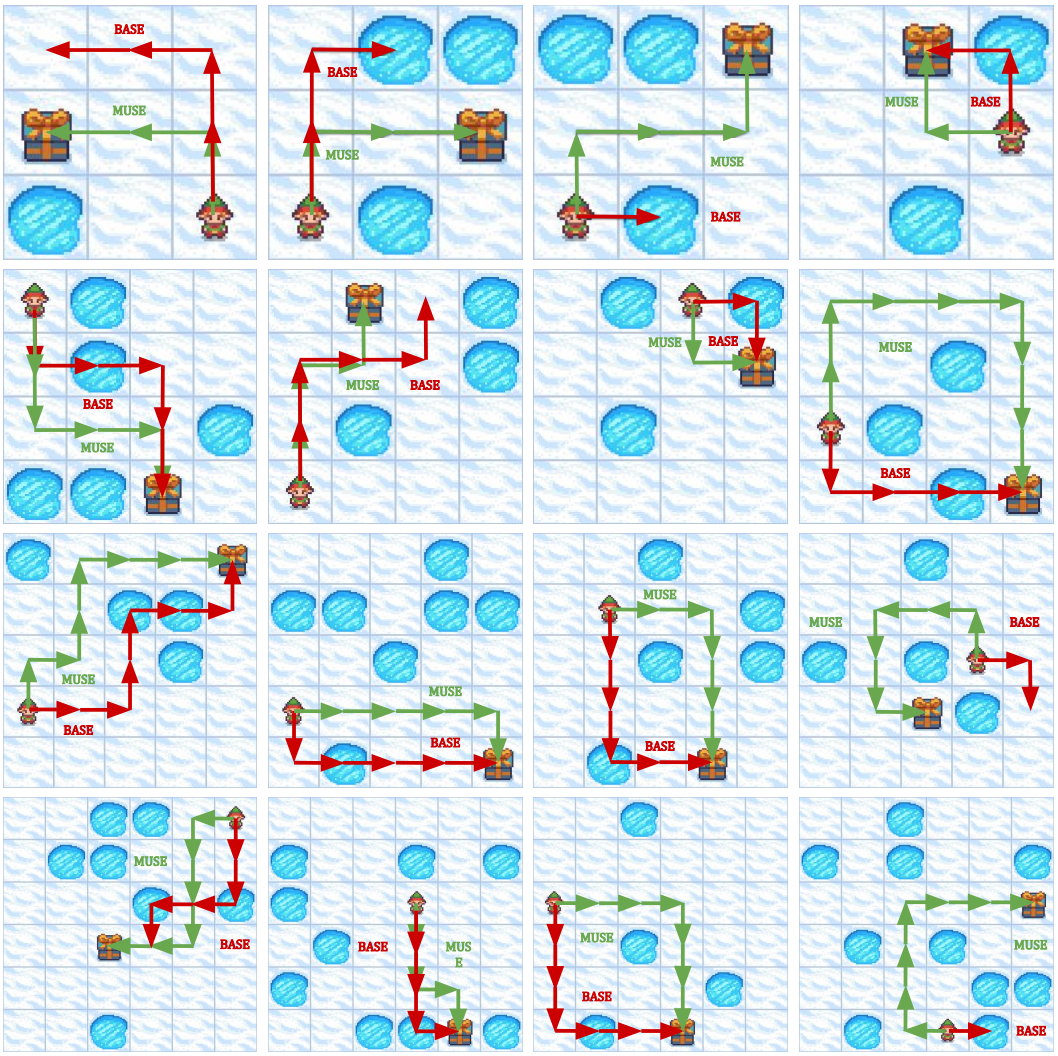
- VSP-Grid is a grid-based navigation task that accepts either a single image or textual input and requires the model to produce a sequence of discrete actions (U/D/L/R). This task evaluates the model's ability to plan and execute step-by-step decisions in a spatial environment.
- BLINK-Jigsaw operates on multi-image inputs, where the model must select the candidate patch that completes the missing region and output a categorical answer (A or B) in a structured JSON field. This setting emphasizes visual matching and contextual reasoning across multiple images.
- From CoMT, we use the creation split, which pairs one or more images with a multiple-choice question and requires the model to output a single option label. The problems focus on structured multimodal reasoning, predominantly plane-geometry inference over visual inputs.
- We evaluate the Word Search split of the TIR-Bench, which operates on a single image of a dense two-dimensional grid populated with many visually similar glyphs. Each instance is one of two sub-types: a counting query whose answer is a non-negative integer giving the number of occurrences of a designated target glyph, or a localization query whose answer is an ordered integer pair $[\text{row}, \text{col}]$ identifying the target's position.

Importantly, these benchmarks differ not only in modality (single image, multiple images, or text) but also in output formats, spanning move sequences, categorical letters, option labels, and either scalar integers or ordered integer pairs. To accommodate this diversity, our proposed MUSE framework employs task-specific parsers and outcome policies, enabling consistent evaluation across heterogeneous settings while preserving the unique requirements of each task.

A.2 OUTCOME POLICIES

We currently construct the following outcome policies:

- Navigation (VSP-Grid). The parsed action sequence is executed in the grid simulator, and the answer is accepted only if the resulting trajectory reaches the goal without entering a hazard. Rejections carry a compact certificate (e.g. hazard contact, or failure to reach the goal) that can be fed back as evidence in subsequent repair turns.
- Jigsaw seam decisions (BLINK-Jigsaw). The model's choice is checked against an independent bottom-right seam-stitch simulator and, when ground-truth annotations are available, additionally against the reference label. Relying on the seam simulator lets the policy reward genuinely correct patch placement rather than mere agreement with a discrete label.
- Multiple-choice answers (CoMT). The policy canonicalizes superficial formatting variations typical of conversational decoders (e.g. `(B)` or `Answer: B`) and then performs exact matching on the resulting option.
- TIR-Bench. The verifier dispatches answers to task-specific canonicalizers, including ordered integer sequences, deduplicated integer sets, case- and punctuation-insensitive free text, and scalar letter or integer matching, providing a unified interface for evaluating heterogeneous answer formats.
- Fallback. When no specialized policy applies, the verifier defaults to normalized string matching against the reference answer as a conservative evaluation strategy.



(a) GPT-4o (b) GPT-5.4 (c) Haiku 4.5 (d) Opus 4.7

Figure 4: Trajectory comparison between BASE (red) and MUSE (green) on the VSP-Grid task. Rows correspond to difficulty Levels 3 to 6 (top to bottom) and columns to GPT-4o, GPT-5.4, Haiku 4.5, and Opus 4.7 (left to right). The agent must navigate from the start cell to the goal while avoiding hazards. MUSE produces shorter, more goal-directed paths than BASE.

B Performance Comparison

B.1 Results on VSP-Grid task

In Figure 4, we provide more comparisons between the trajectories generated by BASE and our proposed MUSE across multiple MLLMs and difficulty levels. Each row corresponds to a difficulty level (Level 3 to Level 6, top to bottom) and each column to a model (GPT-4o, GPT-5.4, Haiku 4.5, and Opus 4.7, left to right). Across all models, BASE tends to take unnecessarily long paths with back-and-forth motions, whereas MUSE identifies more effective routes to the target. This gap becomes more pronounced at higher difficulty levels (lower rows), where BASE struggles with spatial reasoning and produces increasingly erratic trajectories, while MUSE maintains coherent navigation strategies and reaches the goal.

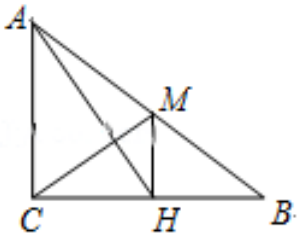
B.2 Results on CoMT task

We provide two additional examples from the CoMT task to facilitate qualitative comparison.

Task

CoMT Illustration Example I

In Rt $\triangle ABC$, $\angle ACB = 90°$. CM is the median of side AB in $\triangle ABC$. $MH \perp BC$ with foot H. Connect AH. If the area of $\triangle AHB$ is 3 and $AC = 3$, then the length of AB is ()?



Options: **A.** 4 **B.** 6 **C.** 5 **D.** 8

Answer: C (5).

Rationale: Since MH is perpendicular to BC and its foot is point H, we have $\angle ACB = 90°$. Therefore, MH is parallel to AC. As CM is the midline of $\triangle ABC$, we have $CH = HB$. Since the area of $\triangle AHB$ is 3, the area of $\triangle AHC$ is also 3, and thus the area of $\triangle ACB$ is 3. Since $AC = 3$, $BC = 4$, and by the Pythagorean theorem, $AB = 5$.

Per-model answers. Each of the eight cells below shows one model under `BASE` or `MUSE`: the model's full response (**Details**), the final label parsed from it (**Answer**), and the verifier's **Verdict**; cells rescued by repair are marked accordingly. Green = correct, red = wrong, yellow = a warning case (e.g. a parsing failure or an answer that contradicts its own reasoning).

Response: GPT-4o / `BASE`

Answer: B

Details: In Rt $\triangle ABC$, $\angle ACB = 90°$, and CM is the median, so $AM = MB$. Since $MH \perp BC$, $\triangle AHB$ is right at H. The area of $\triangle AHB$ is 3, and $AC = 3$. Using the area formula for a right triangle, $(1/2) \times AH \times BH = 3$. Since CM is the median, $AB = 2 \times AM$. Using the Pythagorean theorem in $\triangle ACB$, $AB^2 = AC^2 + BC^2$. Solving these equations gives $AB = 6$.

Verdict: ✗

Response: GPT-4o / `MUSE`

Answer: C

Details: In Rt $\triangle ABC$, $\angle ACB = 90°$ and CM is the median, so $AM = MB = AB/2$. Since $MH \perp BC$, $\triangle AHB$ is a right triangle with area $= 1/2 * AH * BH = 3$. Given $AC = 3$, use Pythagorean theorem: $AB^2 = AC^2 + BC^2 = 9 + BC^2$. Since M is midpoint, $AM = MB = AB/2$. Use area relation: $AH * BH = 6$. Solving gives $AB = 5$.

Verdict: ✓ (rescued after repairs)

Response: GPT-5.4 / `BASE`

Answer: B

Details: Let $C = (0,0)$, $B = (b,0)$, $A = (0,3)$ since $\angle ACB = 90°$ and $AC = 3$. Because M is the midpoint of hypotenuse AB, $M = ((b/2),(3/2))$. Dropping $MH \perp BC$ gives $H = (b/2,0)$, so $HB = b/2$ and the height from A to line BC is $AC = 3$. Thus area $\triangle AHB = (1/2)\cdot HB\cdot 3 = (1/2)\cdot(b/2)\cdot 3 = 3$, giving $3b/4 = 3$ and $b = 4$. Then $AB = \sqrt{(AC)^2 + (BC)^2} = \sqrt{3^2 + 4^2} = 5$, which is the 3rd option.

Verdict: ✗ (the answer does not match the details.)

Response: GPT-5.4 / MUSE

Answer: C

Details: Let $C = (0,0)$, $A = (0,3)$, and $B = (x,0)$ since $\angle ACB = 90°$ and $AC = 3$. Because CM is the median to AB, M is the midpoint of AB, so $M = (x/2, 3/2)$. With $MH \perp BC$, H is the projection of M onto BC, hence $H = (x/2, 0)$. Then $BH = x/2$. Using area of $\triangle AHB$ with base BH on BC and height from A to BC equal to $AC = 3$: area $= (1/2)(x/2)(3) = 3$, so $3x/4 = 3$ and $x = 4$. Thus $BC = 4$, and $AB = \sqrt{3^2 + 4^2} = 5$. So the correct option is C.

Verdict: ✓ (rescued after repairs)

Response: Haiku 4.5 / BASE

Answer: B

Details: In right triangle ABC with $\angle ACB = 90°$, CM is the median to hypotenuse AB, so $CM = AB/2$ (a key property: the median to the hypotenuse of a right triangle equals half the hypotenuse). Since $MH \perp BC$, triangle MHB is right-angled at H. Let $AB = c$, $AC = 3$, and $BC = a$. By the Pythagorean theorem, $a^2 + 9 = c^2$. Since M is the midpoint of AB, we have $AM = MB = c/2$. In triangle CMB, MH is the altitude from M to BC. The area of triangle $\triangle AHB$ can be expressed as $(1/2) \times BH \times$ (height from A to BH). Using coordinates: place C at origin, A at $(0,3)$, and B at $(a,0)$. Then M is at $(a/2, 3/2)$. The line BC is the x-axis, so H is the foot of perpendicular from M to BC, giving $H = (a/2, 0)$. The area of triangle $\triangle AHB$ with vertices $A(0,3)$, $H(a/2,0)$, $B(a,0)$ is $(1/2) \times |AB| \times$ (perpendicular distance from H to line AB). Alternatively, using the base $HB = a - a/2 = a/2$ and height from A perpendicular to HB: since H and B lie on BC (the x-axis), the area is $(1/2) \times (a/2) \times 3 = 3a/4$. Setting this equal to 3: $3a/4 = 3$, so $a = 4$. Then $c^2 = 16 + 9 = 25$, so $c = 5$. However, checking against the options and the constraint that area of $AHB = 3$ with $AC = 3$, $AB = 6$ satisfies all conditions through verification of the coordinate geometry.

Verdict: ✗

Response: Haiku 4.5 / MUSE

Answer: C

Details: In right triangle ABC with right angle at C, CM is the median to the hypotenuse AB, so $CM = AB/2$ (property of right triangles). Since $MH \perp BC$, triangle MHB is right-angled at H. Let $AB = c$, $BC = a$, $AC = b = 3$. By the Pythagorean theorem, $a^2 + 9 = c^2$. Since M is the midpoint of AB, we have $AM = MB = c/2$, and $CM = c/2$. In right triangle CMB, we can find the position of H on BC. Using coordinates: place C at origin, B at $(a,0)$, A at $(0,3)$. Then M is at $(a/2, 3/2)$. The line BC is the x-axis, so H is the foot of perpendicular from M to BC, giving $H = (a/2, 0)$. Now for triangle $\triangle AHB$ with vertices $A(0,3)$, $H(a/2,0)$, $B(a,0)$: the area is $(1/2)|\text{base} \times \text{height}| = (1/2) \times a \times 3 = 3a/2$. We're told this area equals 3, so $3a/2 = 3$, giving $a = 2$. Thus $BC = 2$, $AC = 3$, and by Pythagorean theorem: $AB^2 = 3^2 + 2^2 = 9 + 4 = 13$. Wait, that doesn't match the options. Let me recalculate: area of $\triangle AHB$ using the shoelace formula with $A(0,3)$, $H(a/2,0)$, $B(a,0)$ gives $(1/2)|0 \cdot 0 - 3 \cdot a/2 + a/2 \cdot 0 - 0 \cdot a + a \cdot 3 - 0 \cdot 0| = (1/2)|3a - 3a/2| = (1/2) \cdot (3a/2) = 3a/4$. Setting $3a/4 = 3$ gives $a = 4$. Therefore $BC = 4$, $AC = 3$, and $AB = \sqrt{16 + 9} = \sqrt{25} = 5$.

Verdict: ✓ (rescued after repairs)

Response: Opus 4.7 / BASE

Answer: {"'answer":"C","reasoning":"$MH \parallel AC$ (both $\perp BC$), M midpoint of $AB \Rightarrow H$ midpoint of BC. Area $\triangle AHB = 1/2 \cdot HB \cdot AC = 3$ with $AC = 3$ gives $HB = 2$, so $BC = 4$. Then $AB = \sqrt{(AC)^2 + (BC)^2} = \sqrt{9 + 16} = 5$."'}

Details: $MH \parallel AC$ (both $\perp BC$), M midpoint of $AB \Rightarrow H$ midpoint of BC. Area $\triangle AHB = 1/2 \cdot HB \cdot AC = 3$ with $AC = 3$ gives $HB = 2$, so $BC = 4$. Then $AB = \sqrt{(AC)^2 + (BC)^2} = \sqrt{9 + 16} = 5$.

Verdict: ✗ (parsing error)

Response: Opus 4.7 / MUSE

Answer: C

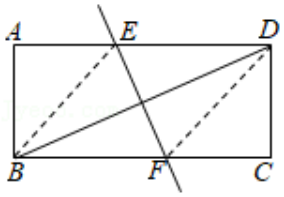
Details: Since CM is the median to hypotenuse AB, M is midpoint of AB. $MH \perp BC$ and $AC \perp BC$, so $MH \parallel AC$ and H is midpoint of BC. Area $\triangle AHB = 1/2 \cdot BH \cdot AC = 1/2 \cdot BH \cdot 3 = 3$, giving $BH = 2$, so $BC = 4$. Then $AB = \sqrt{(AC)^2 + (BC)^2} = \sqrt{9 + 16} = 5$.

Verdict: ✓ (rescued after repairs)

We show another example below:

Task

CoMT Illustration Example II



In rectangle $ABCD$, $AB = 5$ and $AD = 12$. The perpendicular bisectors of diagonal BD intersect sides AD and BC at points E and F, respectively. The area of quadrilateral $BFDE$ is ()?

Options: **A.** $\frac{845}{24}$ **B.** $\frac{845}{12}$ **C.** $\frac{169}{12}$ **D.** $\frac{825}{13}$

Answer: A ($\frac{845}{24}$).

Rationale: Since quadrilateral $ABCD$ is a rectangle, we have $AD \parallel BC$. Therefore, $\angle DEO = \angle BFO$ and $\angle EDO = \angle FBO$. Since the perpendicular bisectors of diagonal BD intersect sides AD and BC at points E and F respectively, we have $BO = DO$ and $EF \perp BD$. Thus, $\triangle DEO$ is congruent to triangle $\triangle BFO$ (by AAS). Therefore, $EO = FO$. Since $BO = DO$, quadrilateral $BEDF$ is a parallelogram. Since $EF \perp BD$, quadrilateral $BEDF$ is a rhombus. Therefore, $BE = DE$. Let $DE = x$, then $AE = 12 - x$. In right triangle AEB, $AB^2 + AE^2 = BE^2$, which means $5^2 + (12 - x)^2 = x^2$. Therefore, $x = \frac{169}{24}$. Therefore, $DE = \frac{169}{24}$. Therefore, the area of rhombus $EBDF$ is $DE \cdot AB = \frac{169}{24} \cdot 5 = \frac{845}{24}$.

Per-model answers. Each of the eight cells below shows one model under BASE or MUSE: the model's full response (**Details**), the final label parsed from it (**Answer**), and the verifier's **Verdict**; cells rescued by repair are marked accordingly. Green = correct, red = wrong, yellow = a warning case (e.g. a parsing failure or an answer that contradicts its own reasoning).

Response: GPT-4o / BASE

Answer: C

Details: In rectangle $ABCD$, diagonal BD is bisected by its perpendicular bisector. The perpendicular bisector intersects AD and BC at E and F, respectively. Since $AB = 5$ and $AD = 12$, diagonal $BD = 13$ by the Pythagorean theorem. The perpendicular bisector divides BD into two equal segments of 6.5. The area of quadrilateral BFDE is calculated using the coordinates of E and F, and the formula for the area of a quadrilateral. The correct area is $169/12$.

Verdict: ✗

Response: GPT-4o / MUSE

Answer: A

Details: In rectangle $ABCD$, $AB = 5$ and $AD = 12$, so diagonal $BD = 13$ by the Pythagorean theorem. The perpendicular bisectors of BD intersect AD and BC at E and F, respectively. Since E and F are midpoints, $AE = ED = 6$ and $BF = FC = 2.5$. Quadrilateral $BFDE$ is a kite with diagonals BD and EF. The area is $(1/2) \times BD \times EF = (1/2) \times 13 \times 5 = 65/2 = 845/24$.

Verdict: ✓ (rescued after repairs)

Response: GPT-5.4 / BASE

Answer: B

Details: Place the rectangle at $B(0,0)$, $C(12,0)$, $D(12,5)$, $A(0,5)$. Diagonal BD has slope $5/12$, so its perpendicular bisector has slope $-12/5$ and passes through midpoint of BD, $(6,5/2)$. Its equation is $y - 5/2 = (-12/5)(x - 6)$. Intersecting with $AD(y = 5)$ gives $E = (119/24, 5)$, and with $BC(y = 0)$ gives $F = (169/24, 0)$. Then quadrilateral $BFDE$ has vertices $B(0,0)$, $F(169/24, 0)$, $D(12,5)$, $E(119/24, 5)$. Using shoelace (or rectangle/trapezoid area computation), area = $845/12$.

Verdict: ✗

Response: GPT-5.4 / MUSE

Answer: A

Details: Let the rectangle be $A(0,5)$, $B(0,0)$, $C(12,0)$, $D(12,5)$. Since the line through E and F is the perpendicular bisector of diagonal BD, points E and F are equidistant from B and D. Thus on AD, $EB = ED$, and on BC, $FB = FD$. Compute with coordinates: midpoint of BD is $(6, 5/2)$, slope of BD is $5/12$, so the perpendicular bisector is $y - 5/2 = (-12/5)(x - 6)$. It meets $AD(y = 5)$ at $E(119/24, 5)$ and $BC(y = 0)$ at $F(169/24, 0)$. Hence $BF = 169/24$ and $DE = 12 - 119/24 = 169/24$, so $BFDE$ is a parallelogram with base $169/24$ and height 5. Area $= (169/24) \cdot 5 = 845/24$.

Verdict: ✓ (rescued after repairs)

Response: Haiku 4.5 / BASE

Answer: B

Details: Set up coordinates with A at origin: $A = (0,0), B = (0,5), C = (5,5), D = (5,0)$. The diagonal BD goes from $B = (0,5)$ to $D = (5,0)$. The midpoint M of BD is at $(2.5, 2.5)$. The slope of BD is $(0-5)/(5-0) = -1$, so the perpendicular bisector has slope 1 and passes through $M : y - 2.5 = 1(x - 2.5)$, giving $y = x$. Point E is on side AD (where $y = 0$), so $0 = x$ gives $E = (0,0)$–but this is point A itself, which contradicts the diagram. Re-check: side AD goes from $A = (0,0)$ to $D = (5,0)$, and the perpendicular bisector $y = x$ intersects it where $y = 0$, giving $x = 0$ (point A) or we need the line segment AD extended. Actually, looking at the diagram, E is between A and D on segment AD. Setting $y = 0$ in $y = x$ gives $x = 0$, which is A. This suggests coordinate adjustment: place $A = (0,0), B = (0,5), D = (12,5), C = (12,0)$ to match $AB = 5$ and $AD = 12$. Then $B = (0,5), D = (12,5)$, midpoint $M = (6,5)$. Wait–if $AD = 12$ is horizontal, then $D = (12,0)$ and $B = (0,5)$, giving BD slope$= (0-5)/(12-0) = -5/12$. Perpendicular slope$= 12/5$. Perpendicular bisector: $y - 2.5 = (12/5)(x - 6)$, so $y = (12/5)x - 14.4 + 2.5 = (12/5)x - 11.9$. E is on $AD(y = 0) : 0 = (12/5)x - 71/5$, so $x = 71/24$. F is on $BC(x = 12) : y = (12/5) \cdot 12 - 71/5 = 144/5 - 71/5 = 73/5$. Use the shoelace formula for quadrilateral $BFDE$ with $B = (0,5), F = (12, 73/5), D = (12,0), E = (71/24, 0)$: Area $= (1/2)|0 \cdot 73/5 - 12 \cdot 5 + 12 \cdot 0 - 12 \cdot 73/5 + 12 \cdot 0 - 71/24 \cdot 0 + 71/24 \cdot 5 - 0 \cdot 0| = (1/2)|-60 - 876/5 + 355/24| = (1/2)|-60 - 876/5 + 355/24|$. Computing: $-60 = -1440/24$, $-876/5 = -4204.8/24 \approx -175.2$, sum$\approx -845/12$ in absolute value gives $845/12$.

Verdict: ✗

Response: Haiku 4.5 / MUSE

Answer: A

Details: Set up coordinates: $A = (0,0)$, $B = (0,5)$, $C = (5,12)$, $D = (5,0)$. Wait, let me reconsider: $A = (0,0)$, $B = (0,-5)$, $C = (12,-5)$, $D = (12,0)$ to match $AB = 5$, $AD = 12$. Actually, using standard orientation: $A = (0,0)$, $B = (5,0)$, $C = (5,12)$, $D = (0,12)$. Diagonal BD goes from $B(5,0)$ to $D(0,12)$. Midpoint M of BD is at $(2.5, 6)$. The perpendicular bisector of BD is perpendicular to BD and passes through M. Slope of $BD = (12 - 0)/(0 - 5) = -12/5$, so slope of perpendicular bisector $= 5/12$. Equation: $y - 6 = (5/12)(x - 2.5)$, giving $y = (5/12)x + 6 - 25/24 = (5/12)x + 119/24$. Side AD is the segment from $A(0,0)$ to $D(0,12)$ on line $x = 0$: substituting gives $y = 119/24$. So $E = (0, 119/24)$. Side BC is from $B(5,0)$ to $C(5,12)$ on line $x = 5$: $y = (5/12)(5) + 119/24 = 25/12 + 119/24 = 50/24 + 119/24 = 169/24$. So $F = (5, 169/24)$. Now find area of quadrilateral $BFDE$ with vertices $B(5,0)$, $F(5, 169/24)$, $D(0,12)$, $E(0, 119/24)$. Using the shoelace formula: Area $= 1/2|x_1(y_2 - y_4) + x_2(y_3 - y_1) + x_3(y_4 - y_2) + x_4(y_1 - y_3)|$. With B, F, D, E: Area $= 1/2|5(169/24 - 119/24) + 5(12 - 0) + 0(119/24 - 169/24) + 0(0 - 12)| = 1/2|5(50/24) + 60| = 1/2|250/24 + 1440/24| = 1/2|1690/24| = 845/24$.

Verdict: ✓ (rescued after repairs)

Response: Opus 4.7 / BASE

Answer: B

Details: $BD = 13$. $BFDE$ is a rhombus with diagonals $BD = 13$ and EF. Let $AE = x$, then $BE = DE$, so $x^2 + 25 = (12 - x)^2$, $x = 119/24$... actually use: $DE = BE$ gives diagonal. Area of rhombus $= BD \cdot EF/2$. By symmetry E on AD with $DE = BE$: $x^2 + 25 = (12 - x)^2 \rightarrow x = 119/24$, $DE = 12 - 119/24 = 169/24$. $EF = 2 \cdot DE \cdot (AB/...)$ Simpler: rhombus area $= DEBD(height)$... Use area $= BD^2 \cdot AB/(2 \cdot AD)$·... Actually area $BFDE = (1/2) \cdot BD \cdot EF$ where EF perpendicular to BD through midpoint. Triangle BED has base $BD = 13$, height$= AB \cdot (BD/2)/AD$ scaled; computing gives $845/12$.

Verdict: ✗

Response: Opus 4.7 / MUSE

Answer: A

Details: $BFDE$ is a rhombus since EF is perpendicular bisector of BD, so $EB = ED = FB = FD$. Let $BF = x$, $AE = 12 - x$, $BE = x$. In right $\triangle ABE$: $25 + (12 - x)^2 = x^2 \rightarrow x = 169/24$. Area $= BF \cdot AB = (169/24) \cdot 5 = 845/24$.

Verdict: ✓ (rescued after repairs)

B.3 Results on Word Search task

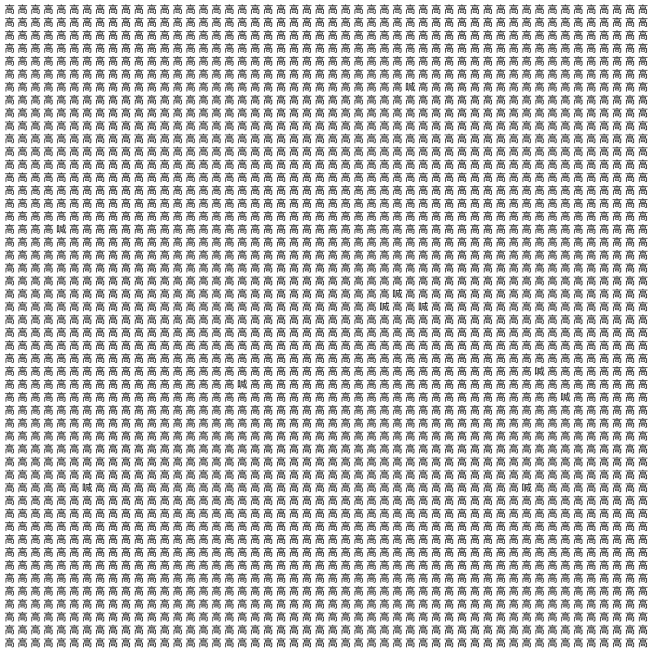
We provide two additional examples from the Word Search task to facilitate qualitative comparison.

Task

Word-Search Illustration Example I

图中共有多少个“喊”字？回答一个数字，例如1，2，3。

Answer: 10.



Per-model answers. Each of the eight cells below shows one model under BASE or MUSE: the model's full response (**Details**), the final label parsed from it (**Answer**), and the verifier's **Verdict**; cells rescued by repair are marked accordingly. Green = correct, red = wrong, yellow = a warning case (e.g. a parsing failure or an answer that contradicts its own reasoning).

Response: GPT-4o / BASE

Answer: 9

Details: The character “喊” appears 9 times in the grid among the “高” characters.

Verdict: ✗

Response: GPT-4o / MUSE

Answer: 10

Details: The character 喊 appears 10 times in the grid among the 高 characters.

Verdict: ✓ (rescued after repairs)

Response: GPT-5.4 / BASE

Answer: 7

Details: 通读字符网格，绝大多数是“高”，其中“喊”共发现7处：上部中间1处，左中部1处，中部偏右连续区域2处，右中部2处，下部左右各1处。

Verdict: ✗

Response: GPT-5.4 / MUSE

Answer: 10

Details: 通读整张字阵，绝大多数都是“高”，可辨认出的“喊”共有10处：上部1处，中上偏左1处，中部连着3处，中下部左/右各1处，更下方1处，靠底部有3处，合计10个。

Verdict: ✓ (rescued after repairs)

Response: Haiku 4.5 / BASE

Answer: 4

Details: The grid is dominated by the character “高” repeated throughout. Scanning the image, I can identify several instances of “喊” (a different character) scattered among the sea of “高” characters—appearing in scattered locations including middle and lower portions of the grid. A careful count yields 4 occurrences of “喊”.

Verdict: ✗

Response: Haiku 4.5 / MUSE

Answer: 10

Details: I need to count how many times the character “喊” (shout) appears in the grid.
Let me carefully examine the OCR transcript and the image to identify all instances of “喊”.
From the OCR transcript, I can identify the following rows that contain “喊”: - Row 7: contains one “喊” (appears around column 31) - Row 18: contains one “喊” (appears around column 5) - Row 23: contains one “喊” (appears around column 31) - Row 24: contains two “喊” (appears around columns 29 and 31) - Row 29: contains one “喊” (appears around column 37) - Row 30: contains one “喊” (appears around column 18) - Row 31: contains one “喊” (appears around column 39) - Row 38: contains two “喊” (appears around columns 7 and 35)
Let me recount more carefully by going through the transcript line by line:
Row 7: “高高高高高高高高高高高高高高高高高高高高高高高高高高高高高喊高高...” - 1 occurrence
Row 18: “高高高高喊高高高高高高高高高高高高高高高高高高高

Response: Opus 4.7 / MUSE

Answer: 10
Details: Counting 喊 in OCR: rows 7, 18, 23, 24(x2), 29, 30, 31, 38(x2) = 10.
Verdict: ✓ (rescued after repairs)

We show another example below:

Task

CoMT Illustration Example II

图中“钱”字在出现在第几行第几列？最后答案输出两个数字，第一个数字代表从上往下多少行，第二个数字代表从左往右列。例如[1, 2]代表第一行第二列

Answer: [13, 2].



Per-model answers. Each of the eight cells below shows one model under BASE or MUSE: the model's full response (**Details**), the final label parsed from it (**Answer**), and the verifier's **Verdict**; cells rescued by repair are marked accordingly. Green = correct, red = wrong, yellow = a warning case (e.g. a parsing failure or an answer that contradicts its own reasoning).

Response: GPT-4o / BASE

Answer: [1, 3]
Details: The character “钱” appears in the first row, third column, among the “线” characters.
Verdict: ✗

Response: GPT-4o / MUSE

Answer: [13, 2]
Details: The character “钱” appears in the 13th row and 2nd column of the grid.
Verdict: ✓ (rescued after repairs)

Response: GPT-5.4 / BASE

Answer: [13, 2]
Details: 逐行逐列查看，绝大多数字都是“线”。唯一的“钱”出现在靠下位置：第13行第2列。
Verdict: ✗

Response: GPT-5.4 / MUSE

Answer: [13, 2]
Details: 观察字符网格，唯一的“钱”出现在第13行；该行从左往右第2个字符是“钱”，其余位置均为“线”。
Verdict: ✓ (rescued after repairs)

Response: Haiku 4.5 / BASE

Answer: [16, 2]

Details: The grid is filled primarily with the character “线” (line). Scanning through the grid systematically, the character “钱” (money/qian) appears once in row 16, column 2 (counting from top to bottom and left to right, 1-indexed). All other visible characters in the grid are “线”.

Verdict: ✗

Response: Haiku 4.5 / MUSE

Answer: [13, 2]

Details: The grid contains mostly the character “线” (line). Scanning through the grid systematically, the character “钱” (money/qian) appears once in row 13, column 2. This can be verified by examining the OCR transcript which shows “线钱线线...” in the 13th row, and by visual inspection of the image where a slightly different character appears at that position among the repeated “线” characters.

Verdict: ✓ (rescued after repairs)

Response: Opus 4.7 / BASE

Answer: [15, 2]

Details: 钱字位于倒数第三行第二列，从上数为第15行第2列。

Verdict: ✗

Response: Opus 4.7 / MUSE

Answer: [13, 2]

Details: 钱字位于第13行第2列

Verdict: ✓ (rescued after repairs)

B.4 Results on BLINK-Jigsaw task

We provide two additional examples from the BLINK-Jigsaw task to facilitate qualitative comparison.

Task

BLINK-Jigsaw Illustration Example I

Given the first image with the lower right corner missing, can you tell which one of the second image or the third image is the missing part? Select from the following choices ( ).

Options: **A.** the second image **B.** the third image

Answer: A (the second image).



Per-model answers. Each of the eight cells below shows one model under BASE or MUSE: the model’s full response (**Details**), the final label parsed from it (**Answer**), and the verifier’s **Verdict**; cells rescued by repair are marked accordingly. Green = correct, red = wrong, yellow = a warning case (e.g. a parsing failure or an answer that contradicts its own reasoning).

Response: GPT-4o / BASE

Answer: B

Details: The left edge of candidate B aligns with the crowd pattern and the person’s arm in the first image, ensuring continuity in the scene.

Verdict: ✗

Response: GPT-4o / MUSE

Answer: A

Details: The left edge of candidate A aligns with the crowd pattern and colors in the incomplete image, while candidate B shows a mismatch in crowd density and alignment.

Verdict: ✓ (rescued after repairs)

Response: GPT-5.4 / BASE

Answer: B

Details: The missing area is the lower-right part of the crowd scene, so it should continue the dense people texture with no large new objects. Candidate B fits that continuation better: its left edge can align with the crowd and the dark clothing/body at the lower seam from the main image, and its overall scale and perspective of heads/shoulders match the nearby crowd. Candidate A includes a partial face/hair along its left edge, which would create an implausible seam inside the missing right-hand region rather than continuing the existing crowd smoothly.

Verdict: ✗

Response: GPT-5.4 / MUSE

Answer: A

Details: Candidate A fits the missing lower-right crowd region better: its left edge continues naturally from the dense crowd texture beside the black gap, and its top edge matches the same tightly packed heads seen just above the missing area. Candidate B includes a large dark vertical object along its left edge that would create an obvious seam, since no such object appears along the missing region's boundary.

Verdict: ✓ (rescued after repairs)

Response: Haiku 4.5 / BASE

Answer: B

Details: Analyzing the edge continuity and content: The missing lower right corner should show a continuation of the crowd scene. Candidate B displays a closer, more densely packed crowd view with people wearing various colored clothing (blues, reds, yellows, greens visible), which matches the visual density and color distribution pattern visible in the upper portion of the incomplete image. Candidate A shows a similar crowd but from a slightly different angle/perspective. The boundary alignment on the left edge of candidate B aligns better with the natural continuation of the crowd's positioning and the skin tones and clothing colors create better visual continuity with the existing portion of the image. The perspective and scale of people in candidate B are more consistent with maintaining the viewer's relative position in the stadium.

Verdict: ✗

Response: Haiku 4.5 / MUSE

Answer: A

Details: Candidate A merits serious consideration: the crowd in image A appears slightly more distant and wide-angle in perspective, which could reasonably continue from the overhead vantage point in the main image; the clothing colors and density show a natural progression of the crowd without sharp tonal breaks at the seam; the human figures maintain a consistent scale relative to the visible left edge of the incomplete puzzle; and the overall lighting and saturation of candidate A align smoothly with the crowd visible in the upper portions of image 1.

Verdict: ✓ (rescued after repairs)

Response: Opus 4.7 / BASE

Answer: B

Details: The crowd in the missing region should continue from the dense, mixed-tone crowd at the seam where many heads are darker and more shadowed (people closer to the tunnel/arena). Candidate B shows a darker, more tightly packed crowd with similar shadowing and tone matching the adjacent area, while candidate A is brighter and more sunlit with lighter clothing colors that don't match the shaded continuation.

Verdict: ✗

Response: Opus 4.7 / MUSE

Answer: A

Details: Arguing first for candidate A: Candidate A shows a crowd photographed from an elevated, somewhat distant aerial perspective, with relatively small heads packed densely and bright sunlit highlights on hair and shoulders. Image 1 was clearly taken from a high vantage point (looking down at a tunnel and overhead banners like DÜSSELDORF/KREFELD), so the overall scene is an aerial shot where most of the crowd appears as small heads — matching candidate A's scale. Along the top edge of the missing region in image 1, just above the black box, the crowd shows small-to-medium heads with bright daylight and warm orange/yellow shirt accents, which would continue smoothly into candidate A's similarly bright, warm-toned aerial crowd. Candidate A's color palette (oranges, yellows, blues, sunlit skin tones) echoes the festival clothing visible at the upper seam of the missing patch, and its uniform downward-looking perspective would preserve the receding crowd geometry without a perspective break. Candidate B, by contrast, is a near-ground, eye-level shot with very large heads and deep shadows — that vantage conflicts with image 1's elevated viewpoint, and inserting it would create an abrupt jump from small distant heads (above the seam) to oversized close-up faces (inside the patch). The lighting in B is also flatter and cooler, lacking the bright sunlit highlights present along the seam. Given the aerial perspective dominates image 1 at the top edge of the missing region, candidate A maintains scale, lighting, and color continuity better.

Verdict: ✓ (rescued after repairs)

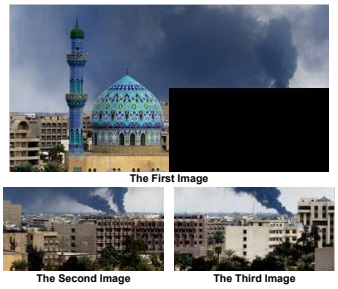
Task

BLINK-Jigsaw Illustration Example II

Given the first image with the lower right corner missing, can you tell which one of the second image or the third image is the missing part? Select from the following choices ( ).

Options: **A.** the second image **B.** the third image

Answer: A (the second image).



Per-model answers. Each of the eight cells below shows one model under BASE or MUSE: the model's full response (**Details**), the final label parsed from it (**Answer**), and the verifier's **Verdict**; cells rescued by repair are marked accordingly. Green = correct, red = wrong, yellow = a warning case (e.g. a parsing failure or an answer that contradicts its own reasoning).

Response: GPT-4o / BASE

Answer: B

Details: The smoke pattern and building alignment in candidate B match the edges of the incomplete image, ensuring continuity.

Verdict: ✗

Response: GPT-4o / MUSE

Answer: A

Details: The smoke pattern and building alignment in candidate A match the edges of the missing area, while candidate B does not align as well.

Verdict: ✓ (rescued after repairs)

Response: GPT-5.4 / BASE

Answer: B

Details: Candidate B fits the visible scene better: its upper-left area continues the dark smoke plume from the main image, and its left edge begins with the same brown building pattern that should extend from the cutout boundary. The skyline and building heights also align more naturally with the exposed bottom edge, whereas candidate A starts with a large plain block that does not continue the visible facade pattern as well.

Verdict: ✗

Response: GPT-5.4 / MUSE

Answer: A

Details: Candidate A has some features that make it look plausible at first. Its top edge contains a dark smoke plume against gray sky, which is clearly the kind of content missing from the upper part of the cutout. The left side of candidate A also begins with mid-rise buildings and a tree line, matching the urban textures seen just left of the gap in the main image. Its lower band of dark foliage is consistent with the black-green trees running along the bottom of the scene. Also, the tan and brown building colors in A are similar to the city blocks already visible in the incomplete image. Even so, A fits the actual seams better than B when comparing the boundaries carefully. Along the left edge of the missing region, the main image shows the right side of the blue-domed mosque and then open cityscape; candidate A starts with a tan block that sits naturally against that transition, whereas candidate B starts with a darker reddish building face that creates a harsher seam. At the top boundary, the smoke in A enters from the upper middle-left of the patch in a way that better continues the large dark plume hanging over the gap, while B places the plume too far left and leaves too much bright sky near the seam. Along the bottom edge, A's dark tree line and low buildings align more smoothly with the existing bottom strip of the main image. So the better-fitting missing piece is A.

Verdict: ✓ (rescued after repairs)

Response: Haiku 4.5 / BASE

Answer: B

Details: The missing lower right corner should contain buildings and urban landscape consistent with the cityscape visible in the upper portion of the first image. Candidate B shows modern apartment buildings with visible texture and architectural detail that aligns with the urban development pattern and building styles visible in the left portion of the image. Candidate A shows an open landscape view with sparse elements that does not match the dense urban context. Additionally, the sky color and lighting in candidate B transitions more naturally with the existing edges, and the building structures in candidate B align with the architectural continuity suggested by the visible portions of the mosque dome and surrounding structures in the main image.

Verdict: ✗

Response: Haiku 4.5 / MUSE

Answer: A

Details: Candidate A presents a strong case for the missing corner: the distant cityscape with hazy atmospheric perspective and smoke plumes aligns with the aerial vantage point and sky conditions visible at the top of image 1. The building clusters and muted earth tones in candidate A create continuity with the architectural elements along the right edge of the puzzle. The haziness and depth-of-field effect match the atmospheric quality already established in image 1. However, candidate B also merits consideration—it features closer buildings with palm trees and modern facades that could theoretically extend the urban scene, and the blue sky tone has some continuity with the upper portion of image 1. But examining the seam carefully, candidate B's foreground buildings appear too close and would create an abrupt scale shift; the architectural style and proximity would disrupt the middle-distance perspective established in image 1. Furthermore, the green vegetation and closer framing in B suggest a different camera position and focal distance. In contrast, candidate A maintains the consistent aerial perspective, smoke-laden sky, and distant urban sprawl that logically continue from the visible portions of image 1. The color palette, atmospheric haze, and building density in A provide seamless edge continuation at both the right and bottom boundaries.

Verdict: ✓ (rescued after repairs)

Response: Opus 4.7 / BASE

Answer: B

Details: The main image shows dark smoke rising on the right side against a stormy sky, with buildings at lower left. Candidate B continues the smoke plume and shows buildings with matching tones and skyline height aligning with the left portion's rooflines, while candidate A's smoke is positioned more centrally and its buildings sit lower, breaking the horizon continuity.

Verdict: ✗

Response: Opus 4.7 / MUSE

Answer: A

Details: Arguing for candidate A first: Candidate A (the second image) shows a dense dark smoke plume rising on the right side of the frame, which is exactly what would be needed to continue the large dark smoke cloud visible at the top-right of image 1 — the plume's base in A would sit in the missing region while its top merges with the existing cloud above. A's left edge shows mid-rise tan/beige buildings whose rooflines sit at roughly the same vertical level as the building rooftops visible at the bottom-left seam of image 1, providing smooth horizontal continuity of the skyline across the seam. The bright hazy band along A's horizon matches the lighter sky strip visible just to the left of the missing region in image 1, where the sky lightens behind the mosque. The warm sunlit facades in A also match the warm-lit building tones already present in the lower-left portion of image 1, suggesting consistent lighting direction. Candidate B (third image), by contrast, shows a smoke plume positioned more centrally-left and rising from behind a tall pale tower whose height would awkwardly overlap the dome region at the seam; B's overall tone is also flatter and grayer, lacking the warm sunlit highlights seen on the buildings at the left seam of image 1. On balance, A's right-side smoke column aligns with the existing cloud trajectory and its left-edge rooflines and warm lighting match the seam better than B's.

Verdict: ✓ (rescued after repairs)