Uni-MMMU: A Massive Multi-discipline Multimodal Unified Benchmark

Kai Zou[1,3]*, Ziqi Huang[2]*†, Yuhao Dong[2]*, Shulin Tian[2], Dian Zheng[4], Hongbo Liu[1], Jingwen He[1,4], Bin Liu[3]✉, Yu Qiao[1]✉, Ziwei Liu[2]✉

[1]Shanghai Artificial Intelligence Laboratory [2]S-Lab, Nanyang Technological University
[3]University of Science and Technology of China [4]The Chinese University of Hong Kong

https://vchitect.github.io/Uni-MMMU-Project

Abstract

Unified multimodal models aim to jointly enable visual understanding and generation, yet current benchmarks rarely examine their true integration. Existing evaluations either treat the two abilities in isolation or overlook tasks that inherently couple them. To address this gap, we present **Uni-MMMU**, a comprehensive and discipline-aware benchmark that systematically unfolds the bidirectional synergy between generation and understanding across eight reasoning-centric domains, including science, coding, mathematics, and puzzles. Each task is *bidirectionally coupled*, demanding models to (i) leverage conceptual understanding to guide precise visual synthesis, or (ii) utilize generation as a cognitive scaffold for analytical reasoning. Uni-MMMU incorporates verifiable intermediate reasoning steps, unique ground truths, and a reproducible scoring protocol for both textual and visual outputs. Through extensive evaluation of state-of-the-art unified, generation-only, and understanding-only models, we reveal substantial performance disparities and cross-modal dependencies, offering new insights into *when and how* these abilities reinforce one another, and establishing a reliable foundation for advancing unified models.

1 Introduction

Recent advances in large language models (LLMs) (Floridi and Chiriatti, 2020) and high-fidelity image synthesis (Rombach et al., 2022; Peebles and Xie, 2023) have catalyzed multimodal systems that accept visual inputs and produce grounded, instruction-following outputs (Liu et al., 2023; Achiam et al., 2023). Building on this progress, *unified* frameworks have emerged that aim to couple comprehension and generation within a single modeling or training recipe (Deng et al., 2025a; Wu et al., 2025; Lerer et al., 2024). The motivation is clear: many real tasks interleave perception with synthesis; a shared representation promises tighter control loops, reduced orchestration overhead, and better data efficiency.

While unification is intuitively appealing, it remains unclear *when* and *how* generation (Gen) and understanding (Und) actually reinforce one another. Human cognition thrives on this synergy. To solve a challenging geometry problem, a student might draw auxiliary lines—a generative act—to create tangible visual cues that scaffold abstract deduction. Conversely, an artist painting a realistic scene leverages an understanding of optical principles to guide the generative act of depiction. Complex problem-solving thus involves an iterative loop between generation and understanding, where the correctness of both the intermediate steps and the final results is crucial for success and thus requires rigorous evaluation.

However, existing benchmarks for unified models fail to adequately assess this critical interaction. They are often limited in two ways: some evaluate generative or understanding capabilities in isolation (Zhao et al., 2025), while others that test both focus on superficial aspects like content association (Chen et al., 2024) or cross-modal consistency (Mollah et al., 2025). Crucially, they lack tasks that enforce a **necessary logical dependency** between the two processes—a dependency that is the cornerstone of complex, multi-step problem-solving.

To bridge this gap, we propose the Uni-MMMU, a novel suite of tasks designed to explicitly evaluate this mutual reinforcement. We curate eight tasks from logically rigorous disciplines such as mathematics, physics, and coding. These fields are ideal as they demand the same interplay of concrete visualization and abstract reasoning seen in human problem-solving. The Uni-MMMU frame-

*These authors contributed equally.
†Project Leader
✉Corresponding authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 908–924
July 2-7, 2026

work probes two core synergistic pathways: *Understanding aids Generation* and *Generation aids Understanding*. Each task is built upon a foundational design of a deterministic reasoning path and a unique correct answer, enabling a dual-level assessment of both the final outcome and intermediate steps. Crucially, this structure underpins our fully automated and reproducible evaluation pipeline, which employs a combination of programmatic parsers, perceptual metrics, and LLM-as-a-Judge to deliver objective, consistent, and interpretable results across all models.

Our evaluation using Uni-MMMU on state-of-the-art models yields a critical insight: the synergy between generation and understanding is most potent in tasks with strong logical dependencies. In these scenarios, intermediate modal information is pivotal, even imperfect, model-generated intermediates significantly improve final accuracy over end-to-end approaches, while oracle intermediates lead to substantial performance gains. This analysis also reveals a clear imbalance in current unified models: they are heavily skewed towards understanding, with generation acting as the primary bottleneck. Common failure points include imprecise image editing, the synthesis of schematic diagrams, and fine-grained spatial reasoning.

In summary, our main contributions are as follows:

- We propose **Uni-MMMU**, a benchmark of eight *bidirectionally coupled* tasks that enforce Gen–Und logical dependency.
- We design a *deterministic, dual-level* evaluation protocol with oracle trajectories and programmatic metrics for reliable, interpretable, and reproducible assessment.
- We provide a comprehensive, multi-discipline evaluation of unified and specialized models, diagnosing where synergy holds and where it breaks.

2 Related Works

2.1 Unified Multimodal Models

Recently, there has been growing interest in developing unified models that integrate both understanding and generation capabilities. Emu3 (Wang et al., 2024) primarily focused on directly combining understanding and generation modules, whereas VILA-U (Wu et al., 2024; Chen et al., 2025b) has explored systematic integration strategies that enable these abilities to complement each other rather than merely merging them. Among these efforts, BAGEL (Deng et al., 2025b) established one of the first widely adopted baselines by employing mixed transformer experts to realize emerging unified multimodal intelligence. Show-o2 (Xie et al., 2025a) further supports video generation within a unified modeling framework. OneCAT (Li et al., 2025a) proposed a decoder-only architecture that achieves native multimodal unification, while UniPic (Wei et al., 2025) introduced reinforcement learning algorithms to further advance performance. Despite these advances, most existing models are still evaluated separately on understanding and generation benchmarks, revealing the lack of a comprehensive benchmark for assessing their joint capabilities from a unified perspective. To address this gap, we propose Uni-MMMU, a holistic benchmark designed to evaluate unified models in an integrated manner and to highlight the synergy between their understanding and generation abilities.

	MMU	Gen&Edit	Multi-Turn	Dual Eval
MMMU	✓	✗	✗	✗
WISE	✗	✗	✗	✗
RISEBench	✗	✓	✗	✗
OpenING	✓	✓	✓	✗
MME-Unify	✓	✓	✗	✗
UniEval	✓	✗	✗	✗
Uni-MMMU	✓	✓	✓	✓

Table 1: **Uni-MMMU VS. prior benchmarks**. MMMU (Yue et al., 2024), WISE (Niu et al., 2025), RISEBench (Zhao et al., 2025), MME-Unify (Xie et al., 2025b), UniEval (Li et al., 2025b), and OpenING (Zhou et al., 2025). We compare across four key dimensions: multimodal understanding (**MMU**), generation and editing (**Gen&Edit**), multi-turn evaluation (**Multi-Turn**), and dual evaluation of the process and result (**Dual Eval**).

2.2 Multimodal Understanding and Generation Benchmarks

Evaluation benchmarks in multimodal AI are evolving, moving beyond siloed assessments of individual capabilities. In understanding, the focus has shifted from basic perception, as in MMMU (Yue et al., 2024), toward integrated "thinking-with-images" paradigms that incorporate generation (Su et al., 2025). Similarly, generation evaluation has progressed from measuring basic semantic fidelity to assessing complex, understanding-driven tasks in frameworks such as ImgEdit (Ye et al., 2025) and Understanding-in-Generation (Lyu et al., 2025).

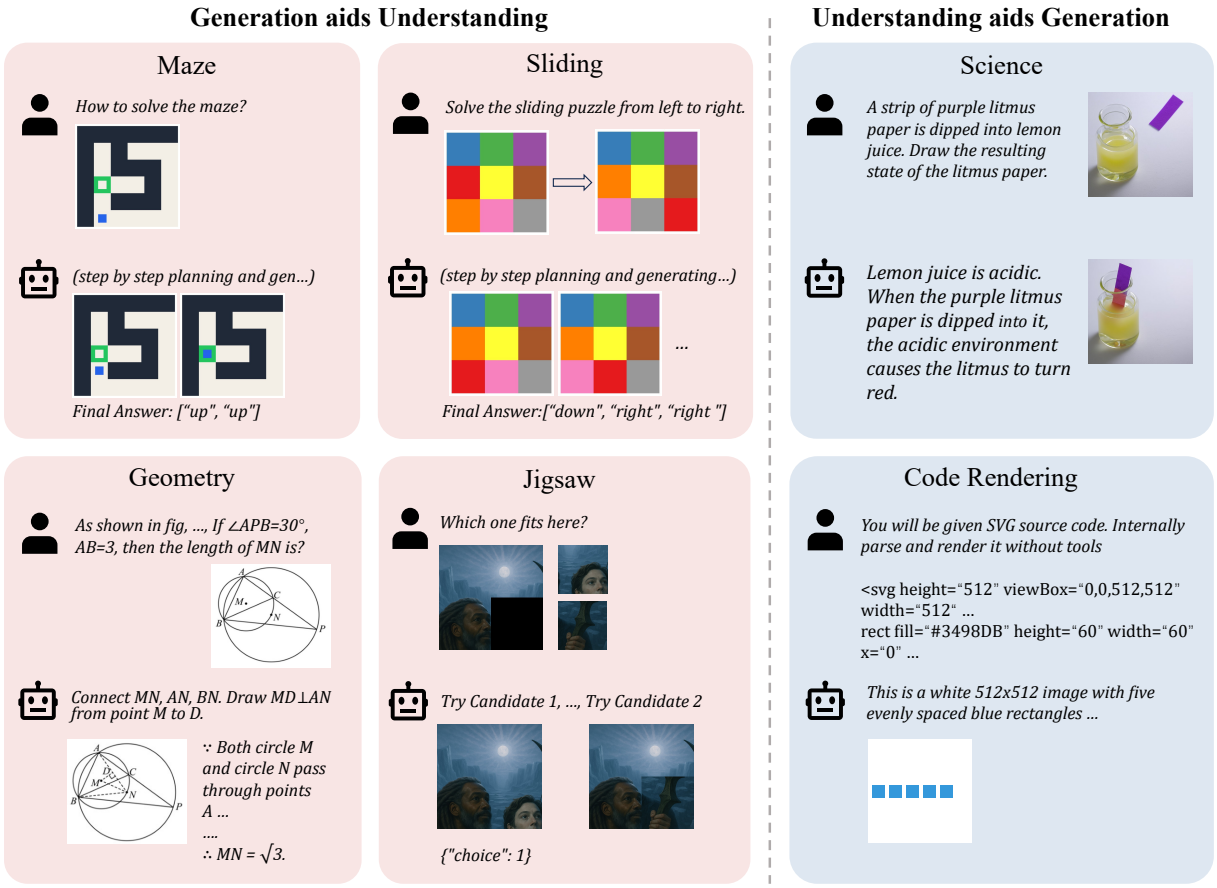


Figure 1: **Overview of Uni-MMMU.** Eight tasks are grouped into two paradigms: generation aids understanding (Maze, Sliding, Geometry, Jigsaw) and understanding guides generation (Science: Physics/Chemistry/Biology; Code Rendering). Each task reports dual-channel scores (text + image).

This evolution extends to new evaluation frameworks like Evaluation-Agent (Zhang et al., 2024) and in the video domain (Huang et al., 2024a,b; Zheng et al., 2025).

While this reflects a clear trend toward synergistic models, a critical gap persists in evaluation: existing unified benchmarks like MME-Unify (Xie et al., 2025b) still assess understanding and generation largely in isolation, failing to probe their interplay. To bridge this gap, we propose Uni-MMMU, a benchmark designed to directly assess this interaction. As summarized in Tab. 1, Uni-MMMU provides a more comprehensive assessment by uniquely combining multimodal understanding with generation, incorporating complex multi-turn tasks, and crucially, performing a dual evaluation of both the intermediate process and the final result to enable fine-grained error attribution.

3 The Uni-MMMU

As shown in Fig. 1, Uni-MMMU is a comprehensive benchmark designed to evaluate the capabilities of unified multimodal models across diverse tasks that require integrated understanding and generation abilities. Our benchmark systematically assesses models' capacity to leverage visual generation as an auxiliary tool for enhanced reasoning and problem-solving.

3.1 Uni-MMMU Benchmark Suite

To better evaluate the unified models' capabilities of mutual enhancement for generation and understanding tasks, we curate a diverse collection of tasks organized into distinct categories based on the type of vision-language interaction required as shown in Fig. 2.

3.1.1 Generation aids Understanding

This paradigm focuses on tasks where *visual generation serves as an external cognitive scaffold* to support intermediate reasoning steps. Rather than treating visual outputs as final products, these tasks require models to iteratively generate visual states and use them to solve complex problems, mirroring

how humans use sketches, diagrams, or step-wise visualizations for spatial reasoning.

Maze Navigation. This task challenges a model's visual state tracking and pathfinding. Models receive a 6×6 "perfect maze" image as input, with a blue block marking the start, a green frame marking the goal, black walls, and white paths. The model must plan and execute the unique shortest path to the goal, alternately generating (1) the next move direction and (2) the corresponding updated maze image, ultimately outputting the full move sequence as text. Mazes are procedurally generated using *DFS carving* to ensure a loop-free structure and *BFS verification* to guarantee a unique solution. Only mazes with shortest path lengths between 2 and 10 are retained.

Sliding Puzzle. This task evaluates optimal state-space search and visual execution. Models are given the initial and goal states of a 3×3 8-puzzle rendered in a fixed 9-color palette. The task is to produce the shortest solution sequence, alternating between textual move descriptions and visual puzzle states after each move, culminating in a JSON list of moves. Puzzles are generated by applying random moves to the solved state, then using BFS to verify solvability and uniqueness of the optimal path. Instances with multiple shortest paths are discarded.

Geometry with Auxiliary Lines. This task directly tests a model's ability to scaffold logical deduction with visual constructs. Models are required to solve geometry problems by first interpreting textual instructions to generate a new diagram with correctly drawn auxiliary lines. Using this self-generated figure as a visual aid, they must then produce a step-by-step textual solution. The problems are sourced from the Geo-Laux benchmark (Fu et al., 2025) and come complete with the original figures, line instructions, and ground-truth diagrams.

Jigsaw Puzzle. This task assesses visual coherence through conditional generation and comparative reasoning. Models are given a 2×2 image panel with one missing patch and two potential candidates. The required process involves two stages: first, the model must sequentially generate two completed images, one for each candidate patch. Second, it must reason over its own generated outputs to textually justify and decide which candidate

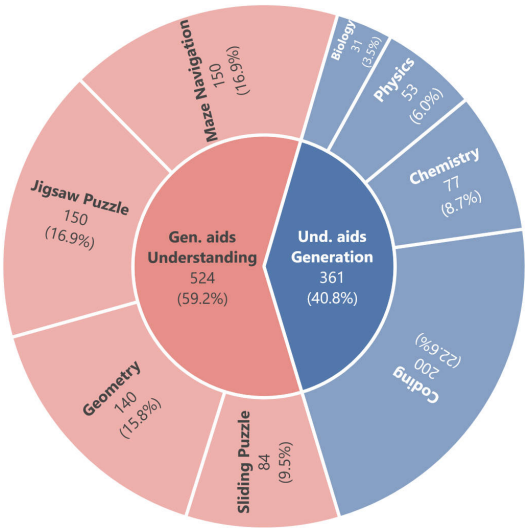


Figure 2: **Data distribution in Uni-MMMU.** The chart illustrates the breakdown of the 885 instances into two primary paradigms—Generation aids Understanding (59.2%) and Understanding aids Generation (40.8%)—and further details the distribution across eight distinct disciplines.

correctly completes the panel. Each instance is created by cropping a 3×3 grid from a sample from ShareGPT-4o-Image (Chen et al., 2025a) dataset, masking the bottom-right patch, and selecting a distractor from the remaining patches.

Across these tasks, visual generation functions as an external working memory: models must maintain and update structured spatial states over multiple steps, which would be error-prone if done purely textually. Each generated visual serves both as a *validation checkpoint* and as a *planning canvas*, enabling robust multi-step spatial reasoning.

3.1.2 Understanding aids Generation

This paradigm flips the evaluation direction: models must first understand and reason, then generate images based on their reasoning, assessing whether they can use conceptual understanding to guide structured generation.

Physics. The tasks probe a model's qualitative reasoning on core physical principles across mechanics, thermodynamics, and electromagnetism. Scenarios are designed to have unambiguous and deterministic visual outcomes, such as predicting changes due to thermal expansion or magnetic attraction. We construct these tasks using a hierarchical, LLM-driven pipeline to procedurally generate textual prompts for initial and final states. These prompts guide the synthesis of corresponding images, which are then manually curated to ensure scientific validity.

Chemistry. The tasks test a model's capacity to predict the visual manifestations of common chemical reactions. The content covers acid-base reactions, oxidation, and precipitation, with scenarios selected specifically for their salient visual cues, like distinct color changes or the formation of a solid. This design ensures that the reasoning is directly linked to the observable outcomes of fundamental chemical principles, such as predicting the result of mixing two reactive solutions.

Biology. The tasks target a model's reasoning about fundamental life science processes and their macroscopic consequences. The content focuses on plant physiology and cellular phenomena, featuring scenarios with clear, observable changes over time. Key concepts include phototropism, fruit ripening, and osmosis, which require the model to visualize the physical transformation resulting from an underlying biological mechanism, such as a plant bending towards light.

Code Rendering. The Code Rendering task probes a model's ability to natively interpret and visualize programmatic graphics. Models are given raw SVG snippets and must (i) produce a concise natural-language description of the depicted scene and (ii) render an image faithful to the SVG specification. The SVG corpus is procedurally generated at three difficulty tiers: *simple* (single primitive), *medium* (multiple non-overlapping primitives), and *complex* (overlapping geometry, control-flow constructs, and symbol reuse).

3.2 Multi-discipline Evaluation Method Suite

The evaluation of Uni-MMMU is underpinned by a fine-grained, programmatic pipeline designed to jointly assess both textual and visual outputs. To ensure objectivity and reproducibility over subjective human scoring, we employ a combination of *(1) deterministic parsers*, *(2) perceptual similarity metrics*, and *(3) model-as-a-judge evaluations*. While each task category features a tailored protocol, a shared philosophy guides our approach: we meticulously evaluate the intermediate visual steps in a process and separately assess the final textual outcome against ground-truth solutions.

Maze Navigation & Sliding Puzzle. For tasks demanding **multi-step spatial planning**, we score both the intermediate visual states and the final textual answers:

- **Image Evaluation.** Each generated state image is processed by a deterministic color-discretization parser. For mazes, the parser crops the 6×6 region and classifies each cell via pixel color distance against a fixed palette, using a 75% majority threshold. For sliding puzzles, it classifies each 3×3 cell by its dominant color with an 80% tolerance. We report two metrics:
  - `img_sample_acc`: A binary score, 1 if *all* generated images in a sequence perfectly match the ground-truth grids after parsing.
  - `img_step_acc`: The fraction of correctly parsed images, averaged across all steps in the solution.
- **Text Evaluation.** The final predicted action sequence (*e.g.*, movement directions or sliding operations) is parsed from the model's text output and compared against the ground-truth path:
  - `text_sample_acc`: A binary score, 1 if the entire predicted sequence exactly matches the ground truth.
  - `text_step_acc`: The proportion of correctly matched moves, evaluated position-wise.

This dual-pronged evaluation allows us to precisely diagnose failures, determining whether they stem from inaccurate state representation (visual errors) or flawed reasoning over those states (textual errors).

Jigsaw Puzzle. Evaluation for jigsaw tasks is twofold, covering both image reconstruction quality and final decision correctness:

- **Image Quality.** For each candidate patch, the model must generate two completed 2×2 panels. Their perceptual similarity to the corresponding ground-truth composites is measured using the DreamSim (Fu et al., 2023) metric, which jointly considers low-level structure and high-level semantics.
- **Decision Accuracy.** The model's final structured JSON output is parsed to extract its chosen candidate index (0 or 1), which is compared to the ground-truth label to compute `text_sample_acc`.

Geometry with Auxiliary Lines. This evaluation employs a **model-as-a-judge** framework using strong open-weights Vision-Language Models (see Tab. 3 for validation):

Model	Generation aids Understanding								Understanding aids Generation						Avg.
	Jig. I	Jig. T	Maze I	Maze T	Slid. I	Slid. T	Geo I	Geo T	Sci. R.	Sci. T	Sci. I	C. T	C. S	C. P	
Unified Models															
Bagel	56.0	48.0	0.0/0.0	0.0/0.0	0.0/0.0	5.6/1.2	8.5	32.8	63.1	57.3	28.0	53.0	2.2	1.8	22.0
OmniGen2	70.3	48.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	4.2	5.7	42.0	33.1	8.9	17.0	13.2	13.3	16.0
Ovis-U1	57.0	53.0	0.0/0.0	12.5/0	0.0/0.0	0.0/0.0	7.1	3.5	42.7	36.3	24.8	18.0	8.0	10.5	16.5
Qwen-Image-Edit	72.0	43.3	0.0/0.0	13.8/0.7	0.0/0.0	3.6/0.0	12.8	8.5	61.1	50.3	26.7	36.0	23.5	20.3	26.3
nano-banana	48.9	57.0	1.8/0.0	23.3/4.7	1.0/0.0	6.2/0.0	21.4	47.8	91.7	79.6	43.9	75.0	36.5	33.7	37.3
GPT4.1 + GPT-image	80.7	80.0	0.8/0.7	49.0/18.1	8.4/0.0	25.6/2.4	25.7	17.1	93.6	91.1	61.8	71.6	83.6	68.6	44.1
Generation-only															
FLUX.1-Kontext	-								–	–	9.5	–	0.1	0.1	–
Imagen 3-001	-								–	–	8.3	–	0.2	0.3	–
Imagen 4	-								–	–	–	–	54.8	51.6	–
Understanding-only															
Qwen2.5-VL-72B	–	72.6	–	42.1/8.7	–	33.0/2.4	–	18.6	-						–
GPT4.1	–	78.0	–	56.1/9.4	–	24.0/2.4	–	16.4	-						–
Gemini-2.5 Pro	–	71.3	–	51.3/8.7	–	44.7/39.3	–	52.1	-						–

Table 2: **Model Performance Comparison on the Uni-MMMU Benchmark.** All scores are normalized to a **[0, 100]** scale for consistency. **Column abbreviations** are as follows: **Jig.** (Jigsaw), **Maze** (Maze Navigation), **Slid.** (Sliding Puzzle), **Geo** (Geometry), **Sci.** (Science), and **C.** (Code Rendering). **I** (Image accuracy), **T** (Text accuracy), **R** (Reasoning accuracy), **S** (Shape&Color score), and **P** (Position score). For multi-step tasks (Maze, Sliding Puzzle), scores in the format *a/b* represent *step-level accuracy / sample-level accuracy*.

- **Image Accuracy** (`image_acc`). The generated diagram with auxiliary lines is evaluated by the open-weights Qwen2.5-VL-72B (Bai et al., 2025), which receives (1) the original figure, (2) the textual auxiliary line instructions, (3) the ground-truth auxiliary line figure, and (4) the model's generated figure. It returns a binary correctness score, tolerating minor stylistic differences but not geometric errors.

- **Text Accuracy** (`text_acc`). The model's step-by-step textual solution is judged by the open-weights Qwen3-32B (Yang et al., 2025), which checks both the logical rigor of the reasoning and the correctness of the final numerical or symbolic answer.

Scientific Reasoning (Physics, Chemistry, and Biology). Evaluation for these tasks is conducted by a VLM judge, which assesses the model's output across three dimensions. It first examines the textual output on two criteria: the correctness of the scientific reasoning (`text_reason_acc`), ensuring the explanation applies relevant principles, and the physical accuracy of the described outcome (`text_result_acc`). The judge then evaluates the generated image (`img_acc`), verifying that it visually and semantically corresponds to the ground-truth final state while maintaining consistency with the initial scene.

Code Rendering. A VLM judge evaluates the outputs using a fine-grained qualitative rubric that compares against the ground-truth rendering. It first assesses the model-generated textual summary (`text_acc`) for semantic consistency, checking for correct object types, counts, colors, and relative layout. The judge then scores the rendered image itself along two axes: geometric and color fidelity (`shape_color_acc`), which covers the correctness of shapes (including polygon side counts) and color usage on a 0–5 scale; and spatial precision (`position_acc`), which evaluates the accuracy of the object layout, alignment, spacing, rotation, and relative placement within the canvas.

A key design choice is that both text and image channels are evaluated separately and jointly, ensuring that a model cannot compensate poor reasoning with good rendering or vice versa. This dual-modality evaluation allows us to isolate whether failures stem from perception, generation, reasoning, or the integration thereof. We deliberately select open-weights models as judges for long-term accessibility and reproducibility; their reliability is validated in Tab. 3.

4 Experiments

4.1 Experiment Setup

We evaluated a suite of 6 advanced unified models and 6 specialized models. The unified models include four leading open-weights systems: Bagel (Deng et al., 2025a), OmniGen2 (Wu et al., 2025), Ovis-U1 (AIDC-AI), and Qwen-Image-Edit (Team and collaborators, 2025); nano-banana (gemini-2.5-flash-image), which is capable of automatic interleaved image and text generation; and

	κ_{img}	κ_{txt}
Ours vs Gemini-2.5-pro	0.7512	0.7538
Ours vs Human	0.717	0.7404

Table 3: **Validity of Uni-MMMU.** The table shows the inter-rater reliability, measured by Cohen's Kappa (κ), between our model-as-a-judge evaluator and both a stronger proprietary model (Gemini-2.5-pro) and professional human annotators.

Model	Jig.	Maze	Sliding	Math	Sci.	Code S	Code P
GPT-image	-	-	-	-	55.4	81.5	66.2
GPT-4.1	78.0	56.1/9.4	24.0/2.4	16.4	-	-	-
GPT	80.0	49.0/18.1	25.6/2.4	17.1	61.8	83.6	68.6
GPT w/ GT	98.0	68.1/24.8	34.7/3.6	27.8	84.0	83.7	72.6

Table 4: **Ablation study.** The table compares the performance of the understanding-only module (GPT-4.1), the unified model generating its own intermediate steps (GPT), and an oracle setup with ground-truth intermediates (GPT w/ GT).

GPT (GPT4.1 + GPT-image) (Lerer et al., 2024), a closed-source agent-based model where GPT4.1 invokes GPT-image for image synthesis. With the exception of nano-banana, all other unified models required manual iterative calls to achieve interleaved generation. For Ovis-U1, which accepts only a single image as a VAE-encoded reference, we used the output image from the previous step for the Maze and Sliding tasks, and a stitched concatenation of the input images for the Jigsaw task. Our evaluation also includes specialized generative models such as the Diffusion Transformer-based FLUX.1-Kontext-dev (Batifol et al., 2025), Imagen 3-001 (Saharia et al., 2022), and Imagen 4, and leading-edge Large Vision-Language Models (LVLMs) for understanding, including Qwen2.5-VL-72B (Bai et al., 2025), GPT4.1 (Lerer et al., 2024), and Gemini-2.5 Pro (DeepMind, 2025).

4.2 Uni-MMMU Evaluation Results

Generation aids Understanding. As shown on the left side of the Tab. 2, we present the performance of various models on the Maze, Sliding, Jigsaw, and Geometry tasks, with all metrics normalized to a 0-100 scale. On the Jigsaw task, GPT achieves the highest scores in both image generation and textual understanding, whereas Bagel performs the lowest in both categories. This suggests a positive correlation between image generation quality and final reasoning accuracy. This finding is further corroborated in the Maze and Sliding tasks, where GPT and nano-banana leverage their superior image generation capabilities to aid reasoning, thereby achieving better path-planning performance. (Nano-banana's lower score on Jigsaw is largely attributable to the additional difficulty imposed by its automatic generation capability, which can lead to penalties for producing an incorrect number of images.) Among the open-source models, Bagel demonstrates strong reasoning abilities, evidenced by its leading final score on the Geometry task. In comparison to specialized understanding models, we observe that unified models generally score lower on Jigsaw. This is primarily because the intermediate generative steps create longer and more complex contexts, which degrades their ability to adhere to the specific formatting instructions for the final answer. Gemini-2.5-pro exhibits the strongest reasoning capabilities, demonstrated by its top performance in Geometry problem-solving and its exceptionally advanced spatial planning skills on the less complex Sliding task.

Understanding aids Generation. As shown on the right side of the Tab. 2, we present the performance of different models on the Science and Code tasks, with all metrics also normalized to a 0-100 scale. On the Science task, a strong positive correlation is evident between understanding/reasoning ability and final generation quality. OmniGen2 exhibits the weakest reasoning and image generation abilities, whereas GPT achieves the highest generation accuracy, attributable to its superior world knowledge and understanding. This observation is reinforced by the performance of non-understanding models (e.g., FLUX-Kontext), which also show the lowest image accuracy. The Code task reveals a notable finding: despite Bagel's strong understanding capabilities, it achieves low image accuracy as it is not adept at producing simple graphical outputs. Furthermore, Imagen4's high score on this task unveils its strong underlying understanding and reasoning abilities. A general trend across models is that positional accuracy is consistently lower than shape accuracy, indicating a common deficiency in precise spatial awareness.

4.3 Validity of Uni-MMMU

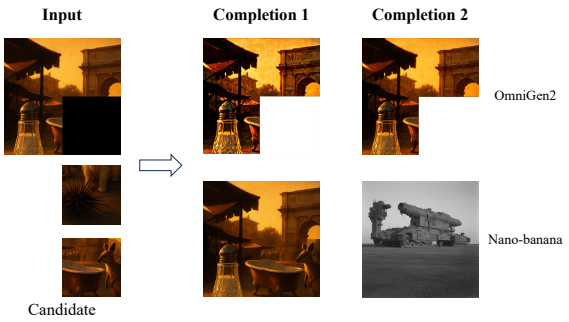
To validate the efficacy of the LLM-based evaluation components in our methodology, we sampled 150 model outputs from across the Math, Science, and Code tasks. These samples were independently scored by professional human annotators and by a more powerful commercial model, Gemini-2.5-pro, using the identical scoring rubrics as our LLM judge. To facilitate a binary comparison for the

Code task, we converted the multi-point image scores into a binary metric: an image was considered a positive sample if both its 'Shape&Color' and 'Position' scores were greater than or equal to 4, and a negative sample otherwise. As shown in Tab. 3, we report the Cohen's Kappa coefficient, a metric that measures inter-rater agreement while correcting for the possibility of agreement occurring by chance and is robust to imbalanced class distributions. The evaluator designed in our method demonstrates a Cohen's Kappa coefficient in the 0.6 to 0.8 range when compared with both human evaluators and Gemini-2.5-pro, indicating substantial agreement. Notably, the consistency for textual evaluations was even higher, demonstrating the strong discriminative performance in assessing understanding capabilities. To complement the binary Cohen's Kappa with a more fine-grained measure, we additionally computed the Pearson correlation on the full 6-level scale of the Code Rendering task (707 samples), yielding coefficients of 0.84 for Shape&Color and 0.76 for Position, further confirming strong agreement between the VLM judge and human annotators.

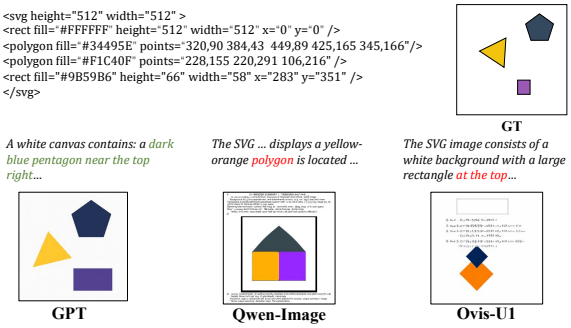
4.4 Insights and Analysis

When is Gen&Und synergy effective? Mutual reinforcement is most beneficial on tasks with a *strict logical dependency* between intermediate states and final outcomes. As evidenced by Tab. 4, coupling the understanding module with intermediate generative states outperforms running understanding and generation in isolation; notably, even *imperfect* intermediates confer gains over the decoupled baseline, while supplying ground-truth intermediates yields consistent and sizeable improvements across all tasks. The effect is especially pronounced for Maze (solution accuracy) and Science (image correctness). As shown in Fig 3b, we also observe clear causal handoffs in the *Code Rendering* task: when GPT-4.1 correctly analyzes SVG semantics, GPT-image reliably produces faithful renderings; conversely, misanalysis propagates directly into visual errors. This confirms that a well-defined intermediate pathway is not just beneficial but critical for solving complex, multi-step problems.

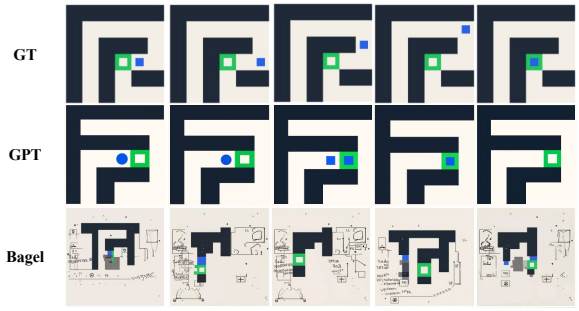
Current capability landscape. Overall, current unified models exhibit *stronger understanding than generation*, with generation often the bottleneck—consistent with in Tab. 2 where image-generation scores generally trail understanding scores. As shown in Fig 3, common failure



(a) Error examples for Jigsaw.



(b) Examples of Code Rendering.



(c) Examples of Maze puzzle.

Figure 3: **Qualitative case studies illustrating common failure modes of unified models on the Uni-MMMU benchmark.**

modes concentrate in the intermediate steps: (i) instruction-following lapses (adding or omitting requested elements; rendering text-only fields onto images), (ii) background and style drift across edits, (iii) fragile spatial perception and world-knowledge application, and (iv) topology/semantics violations that corrupt downstream reasoning. Concretely, for *Maze*, GPT maintains inter-image consistency yet sometimes distorts wall–path topology or object placement, misleading subsequent planning; Bagel tends to inject extraneous, nonsensical glyphs that render states unparsable. For *Jigsaw*, OmniGen2 frequently copies the 2×2 reference rather than producing a coherent completion, whereas nano-banana can generate the completion but occasion-

ally introduces irrelevant content that confuses later decisions. Similar background-consistency and instruction-following issues appear in *Sliding Puzzle* and *Geometry*. In *Code*, Ovis-U1 and OmniGen2 often misread SVG semantics (colors, side counts, sizes, or relative positions), while Qwen-Image-Edit erroneously rasterizes the `Render Summary`—specified as text-only—onto the image. Addressing these deficits will likely require tighter controllability (e.g., program- or constraint-guided generation), stronger spatial/state invariants across edits, and interfaces that make executable intermediate representations first-class citizens in the reasoning–generation loop.

5 Conclusion

We introduce Uni-MMMU, a comprehensive benchmark designed to address the gap in evaluating the synergistic capabilities of unified models. Through eight diverse, reasoning-centric tasks, Uni-MMMU assesses models on bidirectionally coupled challenges where either generation aids understanding or understanding aids generation. Our novel evaluation pipeline scores both intermediate processes and final outcomes to enable fine-grained analysis. Our extensive evaluation of state-of-the-art models reveals significant room for improvement, highlighting common failures in precise instruction adherence, and spatial reasoning. To ensure full transparency and reproducibility, we release all code, datasets, evaluation tools, and judge configurations at https://github.com/uni-mmmu/Uni-MMMU.

6 Acknowledgements

This study is supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 2 (MOET2EP20221-0012, MOE-T2EP20223-0002). This research is also supported by cash and in-kind funding from NTU S-Lab and industry partner(s). This work is also supported by Shanghai Artificial Intelligence Laboratory.

7 Limitations

While Uni-MMMU takes a significant step toward evaluating the synergistic capabilities of unified multimodal models, we acknowledge several limitations that also highlight promising directions for future research.

First, the scope of tasks in Uni-MMMU is primarily focused on reasoning-centric disciplines with deterministic and verifiable solutions, such as science, coding, and puzzles. This design choice enables objective and reproducible evaluation but does not cover a broader range of real-world scenarios that may require open-ended creativity, subjective judgment, or nuanced commonsense reasoning. Furthermore, the current benchmark is based entirely on static images, leaving the evaluation of models on tasks involving video or longer-term temporal interactions as an area for future work.

Second, our data curation methodologies have inherent trade-offs. For tasks like Maze Navigation, Sliding Puzzle, and Code Rendering, we employed procedural generation to ensure unique solutions and facilitate objective parsing. However, this approach may result in data that lacks the complexity, noise, and visual diversity of real-world imagery. For the Science tasks, we used an LLM-driven pipeline followed by manual curation to ensure scientific validity. While rigorous, this process could potentially introduce subtle biases from the generation models or human curators.

Finally, our evaluation pipeline has its own constraints. For several tasks, we rely on a "model-as-a-judge" framework using powerful VLMs. Although we validated the substantial agreement of our VLM judge with human annotators, these judge models are not infallible and may have their own biases or knowledge gaps that could affect evaluation accuracy.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AIDC-AI. Ovis-u1: Unified understanding, generation, and editing. https://github.com/AIDC-AI/Ovis-U1. Accessed 2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, and 1 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, and 1 others. 2025. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*.

Dongping Chen, Ruoxi Chen, Shu Pu, Zhaoyi Liu, Yanru Wu, Caixi Chen, Benlin Liu, Yue Huang, Yao

Wan, Pan Zhou, and 1 others. 2024. Interleaved scene graphs for interleaved text-and-image generation assessment. *arXiv preprint arXiv:2411.17188*.

Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. 2025a. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation. *arXiv preprint arXiv:2506.18095*.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. 2025b. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*.

Gemini Team Google DeepMind. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. Technical report, Google DeepMind. Technical report describing Gemini 2.5 Pro.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. 2025a. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. 2025b. Emerging properties in unified multimodal pretraining. *Preprint*, arXiv:2505.14683.

Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds and machines*, 30(4):681–694.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. 2023. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*.

Yumeng Fu, Jiayin Zhu, Lingling Zhang, Bo Zhao, Shaoxuan Ma, Yushun Zhang, Yanrui Wu, and Wenjun Wu. 2025. Geolaux: A benchmark for evaluating mllms' geometry performance on long-step problems requiring auxiliary lines. *arXiv preprint arXiv:2508.06226*.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. 2024a. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. 2024b. VBench++: Comprehensive and versatile benchmark suite for video generative models. *arXiv preprint arXiv:2411.13503*.

Adam Lerer, Adam P. Goucher, Aditya Perelman, Aidan Ramesh, Aidan Clark, A. J. Ostrow, Akila Welihinda, Alec Radford, Aleksander Mądry, and 1 others. 2024. Gpt-4o system card. arXiv preprint arXiv:2410.21276. System Card, OpenAI.

Han Li, Xinyu Peng, Yaoming Wang, Zelin Peng, Xin Chen, Rongxiang Weng, Jingang Wang, Xunliang Cai, Wenrui Dai, and Hongkai Xiong. 2025a. Onecat: Decoder-only auto-regressive model for unified understanding and generation. *Preprint*, arXiv:2509.03498.

Yi Li, Haonan Wang, Qixiang Zhang, Boyu Xiao, Chenchang Hu, Hualiang Wang, and Xiaomeng Li. 2025b. Unieval: Unified holistic evaluation for unified multimodal understanding and generation. *arXiv preprint arXiv:2505.10483*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Yuanhuiyi Lyu, Chi Kit Wong, Chenfei Liao, Lutao Jiang, Xu Zheng, Zexin Lu, Linfeng Zhang, and Xuming Hu. 2025. Understanding-in-generation: Reinforcing generative capability of unified model via infusing understanding into generation. *Preprint*, arXiv:2509.18639.

Sabbir Mollah, Rohit Gupta, Sirnam Swetha, Qingyang Liu, Ahnaf Munir, and Mubarak Shah. 2025. The telephone game: Evaluating semantic drift in unified models. *arXiv preprint arXiv:2509.04438*.

Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Chaoran Feng, Kunpeng Ning, Bin Zhu, and 1 others. 2025. Wise: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*.

William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Sam Ghasemipour, Alexander M. Rush, and 1 others. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, Linjie Li, Yu Cheng, Heng Ji, Junxian He, and Yi R. Fung. 2025. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *Preprint*, arXiv:2506.23918.

Qwen Team and collaborators. 2025. Qwen-image: Crafting with native text rendering. *arXiv preprint arXiv:2508.02324*. Technical report / preprint.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, and 1 others. 2024. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*.

Hongyang Wei, Baixin Xu, Hongbo Liu, Cyrus Wu, Jie Liu, Yi Peng, Peiyu Wang, Zexiang Liu, Jingwen He, Yidan Xietian, Chuanxin Tang, Zidong Wang, Yichen Wei, Liang Hu, Boyi Jiang, William Li, Ying He, Yang Liu, Xuchen Song, and 2 others. 2025. Skywork unipic 2.0: Building kontext model with online rl for unified multimodal model. *Preprint*, arXiv:2509.04548.

C. Wu and 1 others. 2025. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*.

Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, and 1 others. 2024. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. 2025a. Show-o2: Improved native unified multimodal models. *Preprint*, arXiv:2506.15564.

Wulin Xie, Yi-Fan Zhang, Chaoyou Fu, Yang Shi, Bingyan Nie, Hongkai Chen, Zhang Zhang, Liang Wang, and Tieniu Tan. 2025b. Mme-unify: A comprehensive benchmark for unified multimodal understanding and generation models. *arXiv preprint arXiv:2504.03641*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. 2025. Imgedit: A unified image editing dataset and benchmark. *Preprint*, arXiv:2505.20275.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.

Fan Zhang, Shulin Tian, Ziqi Huang, Yu Qiao, and Ziwei Liu. 2024. Evaluation agent: Efficient and promptable evaluation framework for visual generative models. In *Annual Meeting of the Association for Computational Linguistics (ACL), 2025*.

Xiangyu Zhao, Peiyuan Zhang, Kexian Tang, Xiaorong Zhu, Hao Li, Wenhao Chai, Zicheng Zhang, Renqiu Xia, Guangtao Zhai, Junchi Yan, and 1 others. 2025. Envisioning beyond the pixels: Benchmarking reasoning-informed visual editing. *arXiv preprint arXiv:2504.02826*.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. 2025. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*.

Pengfei Zhou, Xiaopeng Peng, Jiajun Song, Chuanhao Li, Zhaopan Xu, Yue Yang, Ziyao Guo, Hao Zhang, Yuqi Lin, Yefei He, and 1 others. 2025. Opening: A comprehensive benchmark for judging open-ended interleaved image-text generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 56–66.

A Benchmark Details

A.1 Generation aids Understanding

A.1.1 Maze Navigation

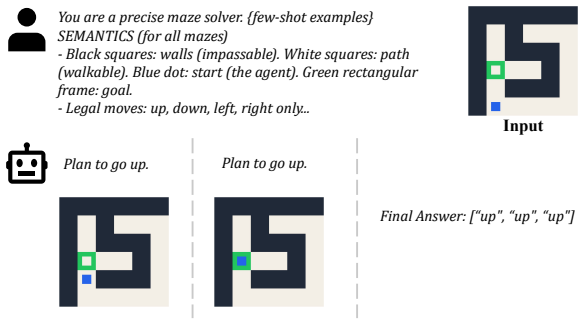


Figure 4: **Details of Maze Puzzle.**

Task Definition. As shown in Fig 4, given the initial state of a 6×6 'perfect maze' as an image (blue block for the agent's start position, green frame for the goal, black for walls, and white for paths), the model is tasked with planning and executing the unique shortest path to the goal. The output must be a sequence of state images, each depicting the board after a single move, followed by a final textual representation of the entire path. This task evaluates the model's capabilities in visual parsing, spatial layout comprehension, multi-step spatial reasoning, and adherence to complex output format instructions.

Construction. Mazes are procedurally generated to ensure a unique shortest path for unambiguous evaluation. The core algorithm first employs a Depth-First Search (DFS) to carve passages, creating a 'perfect maze' with no loops. This guarantees that a single unique path exists between any two cells. The DFS process starts from a random odd-numbered coordinate and uses a step size of two, effectively creating corridors and preventing trivial patterns. Subsequently, a Breadth-First Search (BFS) is used to find the shortest path and verify its uniqueness. Generated mazes are filtered to ensure their shortest path lengths fall within a controlled range of 2 to 10 steps. All visual elements are rendered with a fixed, minimalistic style and color palette to facilitate robust programmatic parsing.

Sampling. The prompting strategy is adapted to the model's architecture. For models capable of autonomously generating interleaved image and text outputs, a single prompt is issued to elicit the complete solution. For models that require separate calls for text and image generation, we employ an iterative process: we first prompt the model to generate the text for the next move, then use this plan to prompt the image generator for the corresponding state image. This loop continues until the model outputs the final sequence of path actions.

Evaluation. An automated programmatic parser is used to evaluate each generated image. As shown in Fig 5, the parser first isolates the main maze area from the image, discretizes it into a 6×6 grid, and converts it into a character-based matrix. This is achieved by analyzing the color distribution of pixels within each cell and calculating their distance to a predefined reference palette. A color tolerance threshold is applied; for instance, a cell is classified as a wall only if over 75% of its pixels are closer to the wall color than any other color in the palette. Cells that cannot be reliably classified are marked with '?'.

The evaluation metrics are divided into two categories:

- **Intermediate Steps (Image):**
  - `img_sample_acc`: A binary score for the entire sample. It is 1 if and only if every generated state image, after parsing, perfectly matches its corresponding ground-truth grid representation.
  - `img_step_acc`: The fraction of generated images that correctly match their corresponding ground-truth state, calculated over the total number of steps in the ground-truth path.
- **Final Answer (Text):**
  - `text_sample_acc`: A binary score. It is 1 if the generated list of moves exactly matches the ground-truth sequence in both content and order.
  - `text_step_acc`: The proportion of moves in the predicted sequence that correctly match the ground-truth moves, calculated position-wise.

A.1.2 Sliding Puzzle

Task Definition. As shown in Fig 6, given an initial and a final (solved) state of a 3×3 sliding puzzle (8-puzzle) as images, the model must devise the shortest sequence of moves to solve it. The required output is a series of intermediate state images, each representing the board after one move,

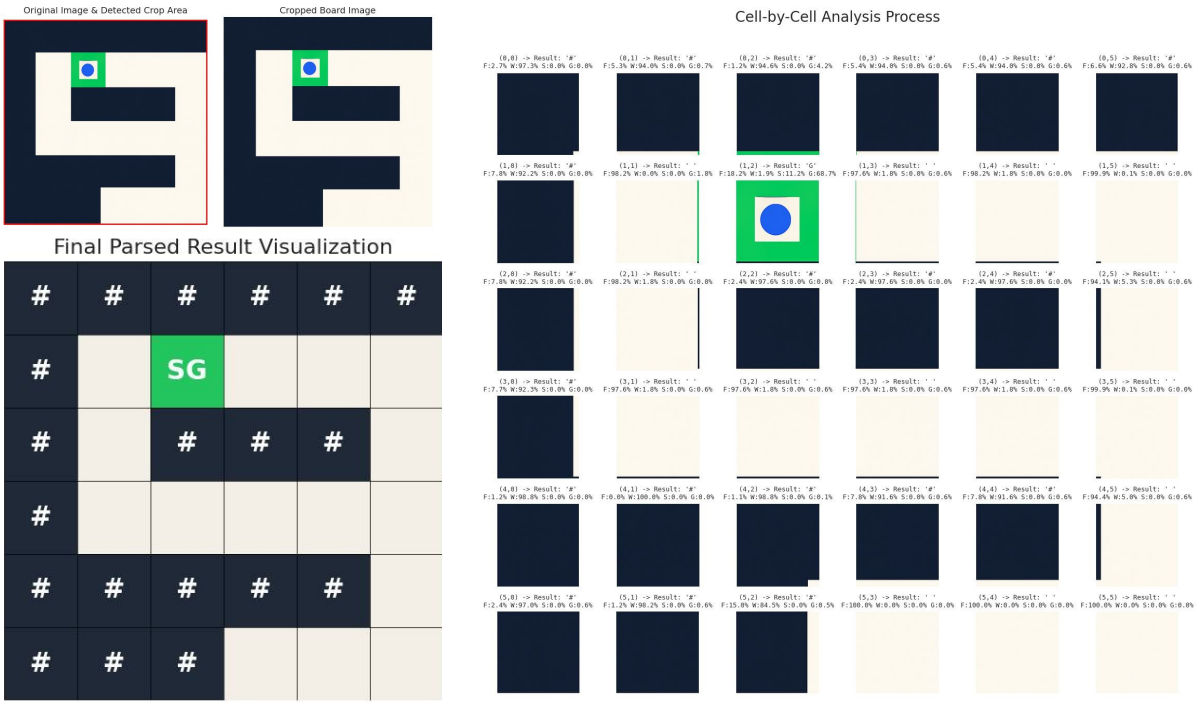


Figure 5: **Overall Pipeline of Programmatic Parser.**

followed by a final JSON list of the moves performed. This task assesses a model's ability to parse complex visual states, perform state-space search for optimal planning, and execute the plan through a sequence of generative actions.

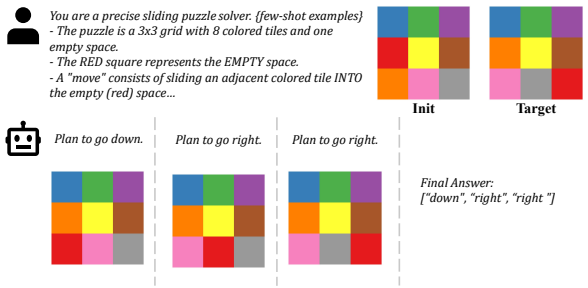


Figure 6: **Details of Sliding Puzzle.**

Construction. Each puzzle instance is procedurally generated to have a unique, optimal solution. The process begins with the solved state, and a number of random moves are applied to generate a scrambled, yet solvable, initial configuration. A Breadth-First Search (BFS) solver is then employed to find the length of the shortest solution path and, critically, to count the number of unique shortest paths. An instance is retained only if this count is exactly one, thereby eliminating ambiguity for evaluation. For robust parsing, the nine tiles (8 numbered, 1 empty) are rendered as solid color blocks using a fixed, high-contrast 9-color palette, with no occluding numbers or borders.

Sampling. The model is prompted using a multimodal, few-shot context. This context includes a complete demonstration with example initial/goal states, the sequence of intermediate solution images, and the final JSON answer. As with the Maze task, the inference procedure is adapted to the model's architecture, employing either a single call for models that support autonomous interleaved generation or an iterative, manual prompting process for models requiring separate calls.

Evaluation. A programmatic evaluation pipeline assesses the model's output. A dedicated parser first processes each generated image. It locates the 3×3 board and discretizes it into a grid of tile identifiers by classifying each cell based on the dominant color's proximity to the reference palette. A strict classification threshold is used: a color must constitute at least 80% of a cell's pixels ('tolerance=0.80') for the tile to be identified. The final score is based on four metrics, as shown in the Maze task.

A.1.3 Geometry

Task Definition. As shown in Fig 7, given a geometry problem consisting of an image and text, the model is required to solve it through a two-stage process that couples generation with reasoning. First, it must interpret textual instructions to

generate a new image by accurately drawing specified auxiliary lines on the original figure. Second, using its own generated diagram as a visual aid, it must produce a step-by-step textual solution, which can be either a calculation or a formal proof. This task directly evaluates the "generation aids understanding" paradigm by assessing the model's ability to create meaningful visual constructs to scaffold complex logical deduction.

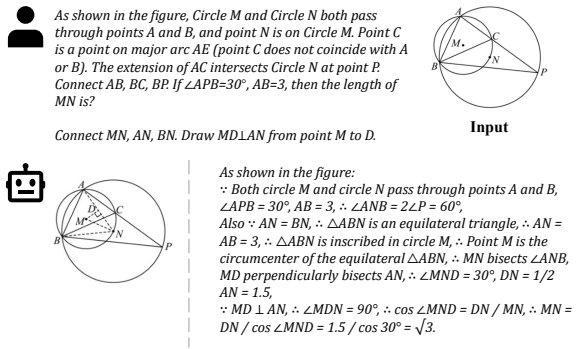


Figure 7: **Geometry.**

Construction. The dataset comprises 140 challenging problems sampled from the established Geo-Laux benchmark (Fu et al., 2025). The collection includes a mix of calculation and proof-based questions, each meticulously annotated with a comprehensive set of ground-truth data: (1) the original problem figure, (2) precise textual instructions in English for constructing necessary auxiliary lines, (3) a ground-truth image showing the correctly drawn auxiliary lines, and (4) a complete, step-by-step textual solution and final answer.

Sampling. A mandatory two-stage inference process is enforced to ensure the dependency of reasoning on generation. **Stage 1 (Image Generation):** The model is first prompted with the problem statement, the original figure, and the explicit instructions for the auxiliary lines. It is tasked with performing a precise image editing operation: overlaying these lines and outputting *only the resulting image*. **Stage 2 (Textual Reasoning):** The image generated by the model in the first stage is then appended to the conversation context. Subsequently, the model is prompted to provide the complete textual solution, compelling it to base its reasoning on the visual cues it has just created.

Evaluation. A model-as-a-judge pipeline is employed for a nuanced, dual-component evaluation of the model's output.

- **image_acc:** The correctness of the generated auxiliary lines is assessed by a powerful Vision-Language Models (VLMs) judge (`Qwen2.5-VL-72B`). The judge receives the original image, the ground-truth auxiliary line image, the model's generated image, and the textual instructions. It provides a binary score indicating whether all required lines were drawn correctly, with tolerance for minor stylistic variations but not for geometric errors.
- **text_acc:** The textual solution is evaluated by a separate Large Language Model (LLM) judge (`Qwen3-32B`). The judge compares the model's reasoning and final answer against the ground-truth solution. It provides a final binary score which is 1 if and only if both the logical steps are rigorous and the final conclusion (or numerical result) is correct.

A.1.4 Jigsaw

Task Definition. As shown in Fig 8, given a 2×2 image panel with one missing quadrant and two candidate patches, the model must identify the correct patch that completes the image. The task is structured to evaluate both generation and understanding capabilities in sequence. First, the model must generate two composite images, each showing the result of placing one of the candidates into the missing slot. Second, based on its own generated outputs, it must provide a textual analysis and a final decision indicating the correct choice. This task assesses visual reasoning about local and global coherence, including continuity of textures, colors, and geometric structures, as well as the ability to perform conditional image generation and subsequent comparative judgment.

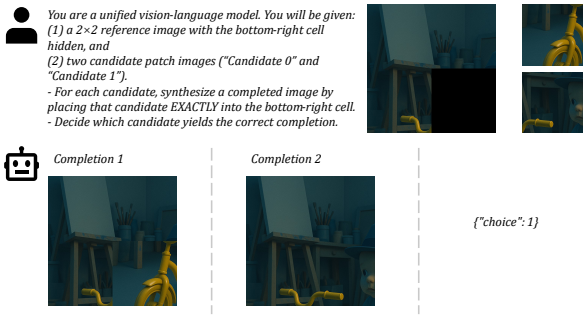


Figure 8: **Details of Jigsaw.**

Construction. Puzzle instances are generated from a high-quality image dataset (Chen et al., 2025a). Each source image is first standardized

by center-cropping and resizing, then partitioned into a 3×3 grid of patches. The problem is formulated using the top-left 2×2 portion of this grid. The patch corresponding to the bottom-right of this 2×2 panel is designated as the target and is masked out in the reference image provided to the model. The two candidates consist of: (1) the ground-truth target patch, and (2) a distractor patch randomly selected from the remaining patches of the same source image. The positions of the correct and distractor candidates are randomized for each instance.

Sampling. The task employs a multi-stage, sequential prompting strategy that compels the model to reason over its own generative outputs. The model is provided with the reference panel and the two candidate patches. The inference proceeds in three steps: **Generate Completion 1:** The model is prompted to generate the completed 2×2 image by inserting the first candidate patch into the missing quadrant. **Generate Completion 2:** The model is then prompted to generate the second completed image using the other candidate patch. **Analyze and Decide:** With both of its generated composite images included in the context, the model is finally prompted to produce a textual rationale and a structured JSON object containing its final choice (0 or 1).

Evaluation The evaluation is twofold, assessing both the quality of the generated images and the accuracy of the final decision.

- **image_score:** The perceptual similarity of the two generated composite images is measured against their corresponding ground-truth versions using the **Dream-Sim** (Fu et al., 2023) metric. A lower Dream-Sim distance indicates a higher quality generation that more accurately reconstructs the scene. A final image score is calculated as `1.0 - mean_distance`, with a penalty applied if the model fails to generate exactly two valid images.

- **text_sample_acc:** The model's final text output is parsed to extract the chosen candidate index from the structured JSON object. This choice is compared against the ground-truth label to compute a standard classification accuracy.

A.2 Understanding aids Generation

A.2.1 Science

Task Definition. As shown in Fig 9, this task assesses a model's ability to apply fundamental principles from natural sciences (physics, chemistry, and biology) to predict and visualize the outcome of physical processes. Given an image depicting an initial state and a textual description of a change or condition, the model must first provide a textual explanation of the resulting final state based on scientific reasoning. Subsequently, it must generate an image that visually represents this final state. This "understanding aids generation" task evaluates the model's world knowledge and its capacity to use causal reasoning to guide a generative process.

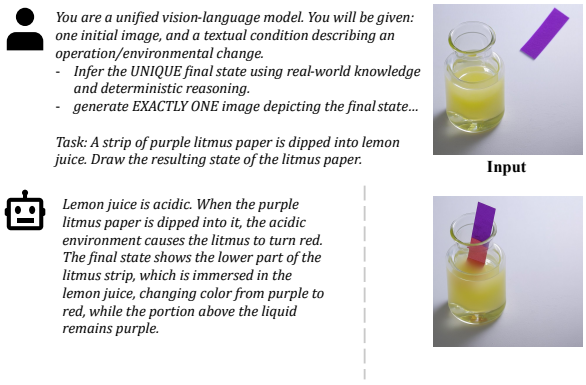


Figure 9: **Details of Natural sciences(physics, chemistry, and biology) Tasks**

Construction. The dataset is constructed through a hierarchical, LLM-driven pipeline. A large language model (GPT) is prompted to generate a wide range of scientific scenarios, starting from broad categories (e.g., Thermal, Fluid Mechanics, Chemistry, Biology) and refining them into specific principles (e.g., Thermal Expansion, Oxidation, Phototropism). Within each scenario, it produces a structured set of text-to-image prompts, including a detailed initial image prompt to create the starting scene and an output image editing prompt describing the transformation to the final state. These prompts are then fed into a unified generation and editing model (nano-banana) to synthesize the initial and ground-truth final images. Every resulting image pair undergoes a rigorous manual curation process, where human experts verify its scientific accuracy, visual plausibility, and ensure the outcome is unambiguous and deterministic. Samples that do not meet these standards are either regenerated or discarded entirely to maintain the integrity of the benchmark.

Inference Procedure. A two-stage "reason, then generate" inference process is enforced. **Stage 1 (Scientific Reasoning):** The model is provided with the initial state image and a text prompt describing the applied condition (e.g., "The temperature is raised to 100 degrees Celsius," "A zinc strip is placed into the solution"). It is first required to output a textual explanation (`<OUTPUT_PROMPT>`) detailing the scientific principles at play and describing the resulting final state. **Stage 2 (Image Generation):** The model's own textual reasoning from the first stage is then added to the conversation context. Subsequently, the model is prompted to generate a single image that visually depicts the final state it just described.

Evaluation. The evaluation is conducted using a dual-component, model-as-a-judge pipeline, assessing both the textual reasoning and the generated image.

- **text_reason_acc:** A Vision-Language Models (VLMs) judge (`Qwen2.5-VL`) evaluates the model's textual output a binary score indicating whether the explanation correctly applies relevant scientific principles.
- **text_result_acc:** A binary score indicating whether the described final state is physically plausible and accurate.
- **img_acc:** The VLMs judge provides a binary score (`image_correct`) assessing whether the image semantically matches the expected outcome while maintaining the consistency of the initial scene.

A.2.2 Code

Task Definition. As shown in Fig 10, this task evaluates a model's ability to interpret and render Scalable Vector Graphics (SVG) source code without an external interpreter. Given raw SVG code as input, the model must first generate a textual description summarizing the visual output it intends to create. It then must generate the final rendered image. This "understanding aids generation" task probes the model's intrinsic knowledge of the SVG specification, its capacity to parse declarative code, and its ability to translate programmatic logic—including control flow—into a precise visual representation.

Construction. The dataset is procedurally generated with three tiers of escalating difficulty to test a range of capabilities.

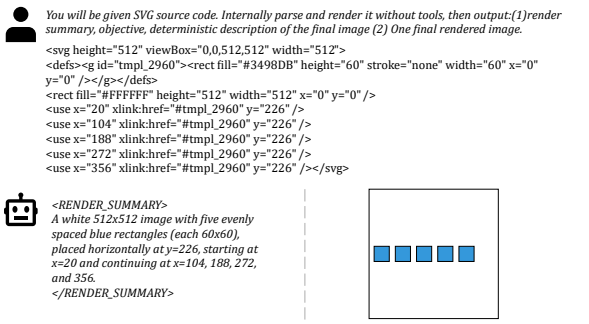


Figure 10: **Details of Code Rendering.**

- **Simple:** A single geometric primitive (e.g., circle, rectangle, or polygon with 3-6 sides).
- **Medium:** Multiple non-overlapping shapes or the introduction of simple control flow.
- **Complex:** Multiple potentially overlapping shapes, including lines and curves, and more advanced control flow.

A crucial feature is the inclusion of programmatic control flow, which is synthesized in the SVG code through constructs like 'for' loops (emulated as regularly spaced, repeated elements) and '<defs>'+'<use>' blocks (defining a template and instantiating it multiple times). All visual elements are drawn from a fixed 8-color palette.

Sampling. A two-stage "understand, then generate" inference process is mandated. **Stage 1 (Code Understanding):** The model receives the raw SVG code and is first prompted to output a concise textual summary (`<RENDER_SUMMARY>`) of the visual scene the code describes. **Stage 2 (Image Generation):** The model's own generated summary from Stage 1 is then added to the context, and it is subsequently prompted to render the final image. This sequence encourages the model to leverage its textual interpretation to guide the visual synthesis.

Evaluation. A Vision-Language Models (VLMs) judge (`Qwen2.5-VL`) performs a qualitative, multi-faceted evaluation of the model's outputs against the ground-truth rendered image.

- **text_acc:** The VLMs judge compares the model-generated summary against the ground-truth image to assess semantic consistency, checking for the correct object types, counts, colors, and relative layout. This yields a binary accuracy score.

- **shape_color_acc:** The VLMs judge evaluates the model-generated image using a detailed rubric, providing scores on a 0-5 scale for two distinct axes: Assesses the correctness of object types (e.g., circle vs. polygon), side counts for polygons, and adherence to the specified color palette.
- **position_acc:** use VLMs as Judge; Assesses the correctness of the overall layout, including relative positions, alignment, spacing, layering (z-order), and rotation.

B Elaboration

Potential Risks. The work focuses on creating a benchmark for evaluating AI capabilities on reasoning tasks like puzzles and scientific problems. The potential risks are minimal and indirect, related to the general misuse of advanced AI, which is outside the scope of this specific benchmark's contribution.

Budget and Setups. All experiments were conducted on NVIDIA A800 GPUs, consuming approximately 48 GPU-days in total. Sampling parameters for all models were set to their official default values.

C Quantitative Failure-Mode Analysis

To complement the qualitative case studies in Fig. 3, we provide per-task quantitative statistics for the most frequent failure modes across unified models:

- **Jigsaw.** nano-banana generates an incorrect number of images in 13.3% of cases and produces irrelevant images (DreamSim distance < 0.2) in 41.6%; OmniGen2 copies the 2×2 reference panel instead of producing a coherent completion in 100% of cases.
- **Code Rendering.** Qwen-Image-Edit erroneously rasterizes the text-only Render Summary onto the generated image in 55% of cases; Ovis-U1 does so in 100% of cases. Both models also frequently misread SVG semantics (wrong colors, polygon side counts, or relative positions).
- **Maze.** Invalid images—those containing any grid cell that cannot be parsed into a valid category (floor, wall, start, or goal)—occur for 15% of nano-banana outputs, 84% of GPT outputs, and 100% of Bagel outputs.
- **Sliding Puzzle & Geometry.** The dominant failures are background/style drift across edits and instruction-following lapses (omitting requested auxiliary lines or adding extraneous elements), which together account for the majority of score degradation.

These statistics confirm that the primary bottleneck lies in *visual generation fidelity*—especially precise spatial editing and instruction adherence—rather than in textual reasoning.